NORTHWESTERN UNIVERSITY

On Adaptive Time-Constrained Macro X-Ray Fluorescence Scanning and Analysis of Works of Art

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Electrical Engineering

By

Henry Hayashi Chopp

EVANSTON, ILLINOIS

September 2023

ABSTRACT

On Adaptive Time-Constrained Macro X-Ray Fluorescence Scanning and Analysis of Works of Art

Henry Hayashi Chopp

In the late 2000's, scientific studies in cultural heritage saw a great advancement in macro X-ray fluorescence (XRF) imaging of paintings. These images are used to generate elemental distribution maps, which aid in identifying chemical elements and paint pigments as well as their locations throughout the layers of the paintings. However, since this technique uses a scanning probe that operates pixel by pixel, it often requires many hours, or even days, to collect high quality image data.

We introduce novel image processing techniques to reduce the acquisition time of the image data regardless of the XRF hardware. We investigate two image denoising techniques: XRF volume denoising, which merges dictionary learning with a Poisson noise model, and elemental map denoising, which incorporates a novel Poissonian regularizer. These denoising methods allow for fast, noisy scans without losing image quality.

Additionally, we detail a pair of sampling algorithms to collect the most informative data. In one method, an initial fast raster scan is conducted, which is then followed by a

pixel-wise dwell time-varying scan designed to minimize the expected error. Our second approach builds upon the first, whereby we predict the initial scanning pattern using only a handful of samples. Knowing that artists paint with a finite number of paints (and therefore XRF responses), these initial samples are strategically chosen via a color image of the painting.

To find these sampling patterns, we detail novel optimization schema that allow users to include strict time constraints. One method, called the Constrained Average, Variance, and Extrema (CAVE) function, is a differentiable function meant to impose strict global mean, range, and/or variance constraints on the output. CAVE is designed for gradient descent-based optimization algorithms, including applications in neural networks. Our other solver is non-differentiable, but is quick to converge upon the exact solution and even allows for additional time constraints to be imposed on the pixel level.

We demonstrate that by combining the denoising and adaptive sampling techniques, we have a powerful framework that can reduce XRF acquisition times to hours or even minutes.

Acknowledgements

I am forever grateful to have so many incredible people in my life that have gotten me to where I am today. First and foremost, I would like to thank my advisor, Professor Aggelos Katsaggelos, for providing me with the opportunity to pursue my doctorate. Aggelos placed his full faith and support in me on both academic and personal levels and can never be thanked enough. I would also like to thank committee members Matthias Alfeld, Oliver Cossairt, and Marc Walton for their invaluable contributions and encouragement. Their deep expertise made my projects possible, and I could not have studied under more passionate and caring mentors.

I would like to thank my colleagues in the Image and Video Processing Lab who provided support throughout my time in graduate school. Thank you Emanuel Azcona for being the best study partner, Semih Barutçu for being nice to only me, Alice Lucas for kicking off my healthier lifestyle, Arjun Punjabi for always bringing life to the lab, Juan Gabriel Serra Pérez for his mentorship and kindness, Philipp Srivastava for supporting my research, and everyone in the Image and Video Processing Lab for their encouragement. Special thanks to Srutarshi Banerjee, who always pushed me to be a better researcher.

I also want to thank all those who worked with me on NSF PIRE grant 1743748, the funder of my work. Qiqin "Tim" Dai, Lionel Fiske, Alicia McGeachy, Emeline Pouyet, and Marc Vermeulen provided academic support and insights that I greatly appreciate. I

would like to thank in particular Katya Bitkin for her encouragement and Matthew Hsu for providing unique opportunities for me to share our work.

My family, immediate and extended, deserves my deepest thanks as well. Mom, Dad, and Ken, your support of me cannot be overstated—this thesis would not be possible without your love. Grandma and Aunt Tobi too, you have always been there for me too. I don't say this often enough, but I love you all so much. To my partner, Matt Bertagna, I love you and the love you have shown me. I cannot wait to see what the future has in store for us. And to the Sokol/Siegel and Heifitz families, thank you for letting me into your families and for the love and support you give to me.

To my friends, thank you for being there as an escape to keep me sane and making life more enjoyable. Thank you to Bryan Kelly, Justin Loew, and Alex Russell for being some of my longtime friends, always making me laugh and always being someone to talk to about anything. My friends still in Chicago, Will and Rachel Trabaris, Laura Tatgenthorst, Kylie Cunningham-Kasper, Nic Scrutton, Ari Aisen, and Ben Gordon, thank you for the countless fun conversations and games; I'm beyond happy we can still get together in person. Last but certainly not least, I want to thank Kolton Boothman, Raudel Cabral, and Faye Oyang for the friendship they provide and genuinely feeling that their success is my success.

This thesis is not mine—it's *ours*.

Dedication

For my late grandfather, Melvyn Chopp, a.k.a. Grandpa Pa—the only person I would have believed if he told me he read this thesis in its entirety. I love you and miss you.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Image Processing Perspectives of X-Ray Fluorescence Data in Cultural Heritage Sciences

Abstract

XRF analysis of art objects has rapidly gained popularity since the late 2000s due to its increased accessibility to scientists. This introduced an imaging component whereby the XRF image volume provides clues as to which chemical elements are present and where they are located spatially in the object. However, as is the nature of collecting measurements, there are limitations preventing perfect acquisition; e.g. spatial resolution, signal-to-noise ratio, etc. The field of image processing, in part, aims to overcome these limitations. Image processing applications in XRF imaging are only just starting to arise due to the increased interest and availability in XRF analysis. In this chapter, we aim to reach readers in XRF imaging or image processing in an effort to call for further research in the field. We review the basics of XRF imaging and analysis that is tailored for those unfamiliar with this imaging modality. We then delve into various publications of image processing methods as applied to XRF data. Throughout this chapter, we examine (and opine on) the XRF field through a lens of the image processing field.

1.1. Introduction

In 1887, Vincent van Gogh painted a patch of grass, aptly titled, *Patch of Grass*. This painting is shown on the left of Fig. 1.1. Its surface appearance only provides a part of the work and the artist's history; it is known that van Gogh by this time was transitioning from a darker paint palette to a lighter one. Underneath the visible surface, however, the painting in its entirety embodies this transition. Little would van Gogh have known that over a century later, Dik *et al.* [1] would use XRF imaging to analyze it (especially considering van Gogh lived until 1890, and x-rays were yet to be discovered by Wilhelm Röntgen in 1895). Their colorized result, shown on the right of Fig. 1.1, uncovered in detail a woman's face that van Gogh painted over. Known to reuse canvases, van Gogh composed the woman and the background with dark pigments prior to overpainting a more vibrant grassy scene. XRF imaging is powerful in that it is capable of revealing the iterations of a painting before the final coat is applied.

The use of XRF in cultural heritage science is not new. Other art objects (aside from paintings) underwent XRF studies, particularly in archaeometrical analysis [2, 3] since acquiring XRF data is non-invasive and non-destructive—certainly an ideal for preservation. What differs is the use of imaging as opposed to spot analysis: prior XRF applications examine select locations of interest to identify pigments and materials. A select few locations would be chemically understood, but these areas are not necessarily representative of other areas similar in visual appearance.

With the advent of XRF imaging in the late 2000s, a flurry of research in cultural heritage science incorporated XRF imaging of paintings. Researchers published applications that expose other hidden paintings [4, 5], authenticate paintings [6], or aid in conservation

Figure 1.1. (Left) Vincent van Gogh, *Patch of Grass*, Paris, April–June 1887, oil on canvas, $30 \times 40\,\mathrm{cm}^2$, Kröller-Müller Museum, Otterlo, The Netherlands (KM 105.264; F583/JH1263). The red frame indicates the field of view of the right image (rotated 90° counter-clockwise). (Right) Approximate color reconstruction using chemical elements Sb and Hg from XRF data. Adapted from [1].

efforts [7] for example. Additionally, researchers developed various low-cost, mobile XRF imaging instruments [8, 9, 10, 11, 12]. These devices provide other scientists increased access to XRF analysis due to their affordability and mobility compared to early experiments using synchotron sources. Mobility is desirable since the artwork remains where it is housed. This minimizes the risk of damaging the painting during transit, easing the concerns of the parties involved.

As with any imaging instrument, however, there are limitations to its capabilities, *e.g.* signal-to-noise ratio, spacial resolution, acquisition time, etc. Image enhancement techniques prove to alleviate these concerns in a wide variety of signal domains. A plethora of denoising, super-resolution, and subsampling methods exist in the literature. Oftentimes these algorithms are domain-specific in order to incorporate prior knowledge

of the signal. Due to the relative newness of XRF imaging, there are understandably fewer dedicated publications of XRF image enhancement techniques. Perhaps because spectroscopists instead of imaging scientists first developed the field, XRF images are conventionally viewed as a stack of *spectra* rather than a stack of *images*. This is not to say that writing this chapter on XRF image processing techniques is unwarranted; rather, the limited number of existing algorithms surveyed in this chapter only adds purpose.

We want in this chapter to provide an image processing perspective to XRF analysis. For example, many of the papers that address interpretability of XRF data use denoising methods, yet denoising is often nowhere mentioned in these papers. This is by no means a critique of the works, but hopefully via this chapter we bring an additional viewpoint to the published work. We have two main goals for this chapter: we aim to reach readers from (1) the XRF community to provide a resource for enhancing XRF imaging and why it should be further researched, and (2) the image processing community to introduce XRF imaging, establish the current state of XRF image processing research, and emphasize the need for further developments in the area. In the discussion to follow, we hope that researchers of either discipline can identify aisles of opportunity for further development in XRF image processing and perhaps foster new interdisciplinary collaborations.

This chapter is structured as follows: we first introduce the science of XRF imaging and analysis. Second, we review and provide new insights in different areas of XRF image processing, namely denoising, super-resolution and inpainting, and subsampling. Lastly, we opine on the state of XRF image processing research as well as directions of further research.

1.2. X-Ray Fluorescence Imaging Overview

XRF imaging was introduced as an alternative to other techniques that image art objects beneath their surfaces. X-ray- and infrared radiation-based imaging are common ways of viewing internal structures of paintings [13], although practically the entire electromagnetic spectrum has been used for these investigations [9]. These methods are employed to avoid extracting samples of the painting. What distinguishes XRF from other modalities is its ability to elucidate atomic elemental composition; this only further reduces the need to remove paint samples for chemical analysis. We will discuss how the underlying science of XRF imaging is used to identify pigments and materials throughout the layers of paints. For a more in-depth yet gentle introduction to XRF spectrometry than provided here, we refer the reader to Brouwer's work [14], which is tailored towards those new to the field.

1.2.1. Physics of XRF Spectroscopy

To collect XRF data, a source illuminates an object with a continuous spectrum of x-rays. These x-rays are collimated on a small spot. As the sample is exposed to x-rays, some of the x-rays are absorbed by the electrons in the sample. Impacted electrons may be dislodged from the atom if the energy of the incoming x-rays are larger than that of the binding energies. Losing electrons creates energetically unfavorable vacancies in the atom's electron configuration. To stabilize, electrons in outer orbitals move inward to fill the vacancy.

During this transition, energy is conserved. Outer orbitals have higher energies than inner orbitals, so the transitioning electron loses energy. This loss is realized as a photon emitted by the atom. The photon's energy equals the energy loss.

Not all photon energies are possible, however. Within any type of atom, there are different electron orbitals/shells at unique and distinct energy levels. The lowest energy orbital is the K-shell, which holds two electrons. The L-shell, subdivided into three sub-shells, has the next three lowest energy levels for eight additional electrons. The M-shell has five sub-shells, all with greater energy than those of the L-shell; it can hold eighteen electrons. These shells constitute the main transition lines whereby electrons fill the vacancies: electrons typically move from (1) the L- to K-shell, (2) the M- to K-shell, and (3) the M- to L-shell. Not all transitions are possible, and some transitions that are more likely to occur than others.

The fluorescence photons are emitted in an isotropic manner, and those not absorbed on their path to the detector are recorded by it. In the semiconductor detector, the photons create a charge that enhances the conductivity. This is transferred into the energy dispersive spectrum used in XRF analysis.

The sources of the incoming x-rays are not only from the electrons in atoms of the top surface layer. X-rays have high energy that can penetrate below the surface layer and interact with hidden atoms. Photons from these atoms must pass back through intermediate layers and into the detector to be recorded. These photons are less frequent than those of the same element that lie on the top level, but are still present in large quantities. There are established limits on the penetration depth that depend on a multitude of factors, but paintings are often thin enough to record photons throughout all the layers.

1.2.2. XRF Measurement Challenges

There are some challenges that arise in XRF spectrometry that introduce noise or artifacts into the recorded spectrum. Here, we highlight some problems that are characteristic of XRF spectroscopy.

1.2.2.1. X-Ray Source. X-ray tubes emit incoherent, polychromatic x-rays by accelerating electrons from a filament towards an anode. Upon contact, the electrons decelerate, and x-rays are generated. A sizeable portion of these x-rays are inadvertently reflected back into the detector, which records a broad spectrum of signals. This is called the continuum, which can be estimated and subsequently subtracted from the spectrum. Some collisions even result in electron vacancies in the anode itself. Photons characteristic of the anode's material are then emitted from the source, which can create a false (or amplified) peak in the XRF spectrum due to this *backscattering*.

1.2.2.2. Interactions between X-Rays and the Object. Aside from the source, x-ray interactions with electrons of the object of interest do not always involve electron ejection or x-ray reflection. *Rayleigh scattering* may occur when electrons hit by x-rays instead vibrate at the same frequency as the incident photons. The vibrations cause photons of the same frequency to be released, which contributes towards the continuum.

Compton scattering occurs when the incoming x-ray is backscattered, but loses some of its energy. This scattering phenomenon is more apparent in low Z elements (*i.e.* elements of low proton count), but can disappear in high Z elements.

1.2.2.3. Detector. One last major origin of error occurs at the detector. Escape peaks occur when incoming photons excite the detector itself. The XRF photon is not reabsorbed but rather escapes the detector. The photon then loses some of its energy

before being converted to a voltage, but the energy loss is well-documented based on the material of the detector.

Pileup peaks can also be produced where two photons are incident on the detector in a small time window. This creates a seemingly large energy equal to the sum of the individual photon contributions that the post-processor cannot resolve as two distinct photons. Pileup is enhanced by high count rates.

These sources of error cannot be controlled, but they can be mitigated in XRF analysis. What can be (roughly) controlled is the number of photons recorded by adjusting the scan time. Photons arrive according to a Poisson process with some unknown underlying rate. Since XRF spectrometry is a photon counting measurement, photon peaks can only be detected if there are enough arrivals to distinguish them from both noise and the continuum. The longer the scan time, the more apparent the peaks will be. XRF imaging presents a challenge in that the dwell times per spot cannot be too small such that peaks are lost in the noise, and cannot be too long such that it takes an excessive amount of time to collect the volume.

Collection times are long since XRF systems are single pixel scanners. In order to generate an XRF image volume, the x-ray source and detector are mounted on a gantry that moves in a (typically) raster scanning motion to cover the desired area. The acquisition process can be lengthy depending on the dwell time and spatial resolution. For example, the XRF volume in Fig. 1.1 reportedly took two days to collect a $17.5 \times 17.5\,\mathrm{cm}^2$ area [1], although being an early paper, care was taken to get good statistics. While XRF systems have since improved, the scan times are still generally on the order of hours or days. The

same area that was scanned in *Patch of Grass* can nowadays be scanned in approximately an hour.

1.2.3. Notation

Before understanding how XRF analysis is done, we need to establish notation. Throughout the thesis, we use the following rules: (1) lowercase lettering denotes scalars, (2) uppercase lettering denotes matrices and vectors, and (3) boldface uppercase letters denote 3D tensors.

Let $\mathbf{X} \in \mathbb{N}^{C\times H\times W}$ be the collected XRF data where $\mathbb{N}$ is the set of nonnegative integers. The volume has height H, width W, and channels C. Each channel corresponds to an energy level where the incoming photons are binned. Each entry $\mathbf{X}_{c,h,w}$ contains the number of recorded photons at pixel (h, w) with energy c.

As will become clear, many XRF analysis techniques revolve around dictionary learning or other matrix factorization methods. Thus, we introduce here some additional terms: $D \in \mathbb{R}_+^{C\times M}$ is the dictionary composed of M different spectra, and $\mathbf{A} \in \mathbb{R}_+^{M\times H\times W}$ denotes the abundances of each of the M spectra.

In dictionary learning, D and $\mathbf{A}$ are found such that

$$\mathbf{X} \approx D\,\mathbf{A} \tag{1.1}$$

where the matrix-tensor multiplication is carried out via

$$(D\mathbf{A})_{c,h,w} = \sum_{m=1}^{M} D_{c,m} \cdot \mathbf{A}_{m,h,w}. \tag{1.2}$$

This is the basis for many of the techniques surveyed here.

1.2.4. XRF Analysis of Individual Response Lines

The core of XRF analysis is unmixing the resultant spectra: which atomic elements are present, and of the present elements, how much is present? All collected spectra are essentially linear combinations of the spectra of individual elements (along with the continuum, noise, backscattering, etc.). Even within the elemental spectra, they too can further be decomposed into the different emission lines.

The first step in XRF analysis is identifying the peaks that indicate the presence of a certain element. This is done manually or automatically by examining the sum spectrum, *i.e.* the spatial sum of all the spectra,

$$S_c = \sum_{(h,w)} \mathbf{X}_{c,h,w}. \tag{1.3}$$

This provides the least noisy presentation of which elements lie in the object. Since it is very likely the same elements/compounds exist throughout the painting spatially, the sum of many measurements reduce the noise. Any elements identified are included in a dictionary composed of the elemental responses. Fig. 1.2 shows a sample sum spectrum as well as a single pixel's spectrum. The peaks are easy to notice in the sum spectrum, but are more difficult to identify when analyzing the single pixel—some maxima may be due to noise in the individual pixel. Once the peaks are identified, a table can be used to attribute the peak's energy to an atomic element.

Of the identified elements, the next step is to decompose the XRF signal at each pixel according to the dictionary. Each peak is often modeled as a Gaussian, and each element consists of one or more peaks. These XRF response curves for elements present

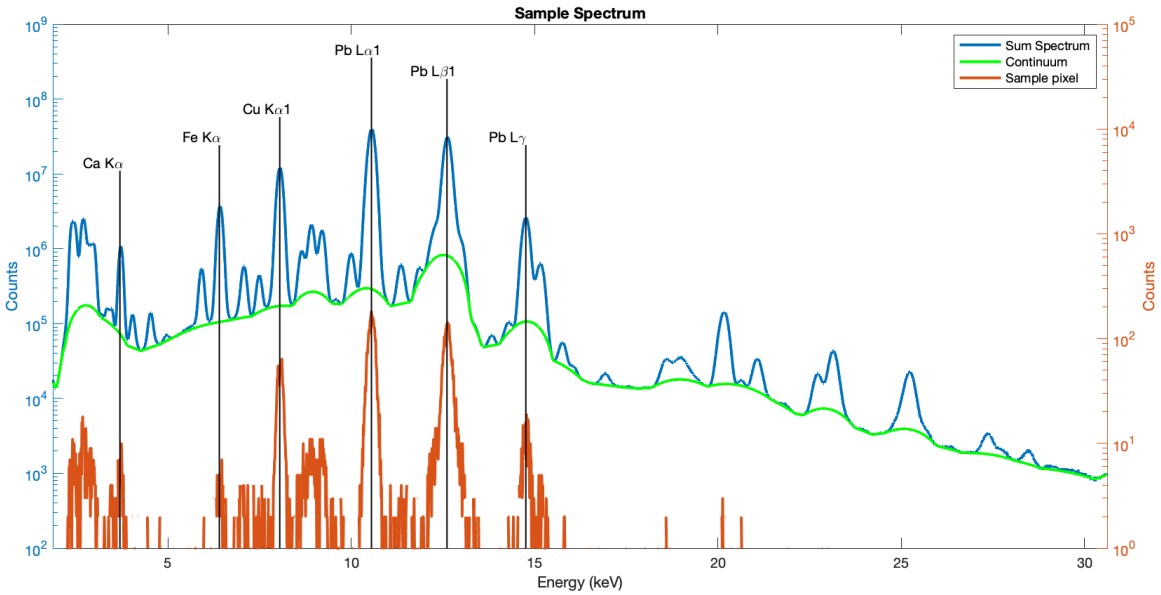


Figure 1.2. Sample XRF spectra with select peaks labeled. Note the different y-scales for the spectra. (Blue) The sum spectrum. (Green) The continuum. (Red) A sample pixel.

in the sample are the columns of the dictionary D. The continuum response is oftentimes included in the dictionary. A nonnegative least-squares approach is then used to fit the XRF signal at each pixel individually:

$$\mathbf{A}^* = \underset{\mathbf{A}\geq 0}{\arg\min}\ \mathcal{L}\left(\mathbf{X}, D, \mathbf{A}\right) \tag{1.4}$$

for some objective (loss) function $\mathcal{L}$, typically the ℓ_2 norm. Optimal abundance $\mathbf{A}^*$ conveys the relative amount of each element present. One can then visualize the individual channels of the abundance matrix to see how much of each element are present across the painting—these are the *elemental maps*.

PyMca [15] is a commonly used platform that can carry out the tasks above in a streamlined approach. It also takes into account other modeling factors beyond the scope of this chapter.

1.2.5. XRF Analysis using Different Bases

Since XRF data analysis is fundamentally an unmixing problem, most techniques use some form of dictionary learning and matrix factorization to analyze the data. Whereas the elemental decomposition formulation of Eq. (1.4) only solves for the abundance matrix $\mathbf{A}$, another formulation solves for the dictionary as well:

$$D^*, \mathbf{A}^* = \underset{D, \mathbf{A} \geq 0}{\arg\min} \; \mathcal{L}\left(\mathbf{X}, D, \mathbf{A}\right). \tag{1.5}$$

This allows for a more complex representation of the data that can aid in interpreting the data.

In particular, interpretability has posed an issue in terms of identifying trace elements. Trace elements have short peaks that may be lost in the total sum spectrum. A priori knowledge is sometimes needed to identify the trace elements in the sum spectrum, and it can be just as difficult to predict the abundance of the trace elements in the individual spectra [16].

In addition to finding trace elements, XRF analysis seeks out correlations in the data. Correlations can be difficult to identify using the individual elemental peaks alone. For example, say an element is present in multiple distinct compounds. It is a difficult task to separate out how much of each compound (as well as which compounds) may be present.

Many classical and some newer methods in data processing are used to analyze the chemical composition of paintings. These analysis techniques can also be used to denoise data in the spectral domain; we will explore this in the next section.

1.3. XRF Interpretation Methods as Spectral Denoising Mechanisms

The first attempts to better interpret XRF image data revolve around the fact that pure elements typically do not exist on their own in paintings, but rather as a compound. Vermillion, for example, is a red paint that was previously made from a Hg and S compound. In XRF analysis, characteristic peaks of Hg and S would appear should vermillion exist. This perhaps allows for dimensionality reduction that identifies pigments and mixtures of different paints that the artist used.

In XRF imaging, changing the composition of the dictionary D from individual elemental responses to a new basis is how many published analyses are carried out. As will become clear, these methods can often be considered denoising algorithms that are able to smooth the original data.

Data denoising is often overlooked in XRF analysis. Notice the bottom plot of Fig. 1.2 again. The signal is quite noisy compared to the sum spectrum, as the signal is inherently discretized and most peaks occur under 20 photon counts. There are two primary ways to mitigate the effects of noise on the individual spectra of a pixel: (1) increase the dwell time for each pixel, and/or (2) use image processing techniques to denoise the data. The former option is typically not available since experimentalists already set the dwell time to the longest reasonable length. Even the slightest addition of dwell time can have immediate impacts on the total scan time. For example, if the scan area is $500 \times 600\,\text{px}$,

each additional millisecond of dwell time per pixel adds 5 minutes of overall scan time. Instead of increasing the dwell time as a way to denoise the data, denoising techniques in image processing can be applied.

In this section, we first address early methods that use principal component analysis (PCA) or clustering approaches. Then, we go into more popular methods used today to perform XRF analysis. Throughout this section, we note how these techniques can be repurposed as potential denoising methods alongside its use as an analysis method in XRF literature.

1.3.1. Principal Component Analysis and Clustering Methods

Much of the early work in XRF interpretation uses PCA to generate a new XRF basis [17]. PCA is known to be an easy but effective way to extract correlations in data as the components (dictionary members) are the eigenvectors of the covariance matrix. The eigenvectors corresponding to the largest eigenvalues are chosen as they best capture the direction of highest variance in the data. Oftentimes only a few components are needed to capture a significant amount of the variation. Each pixel is then approximated by a linear combination of the PCA vectors, which compose D. The abundances $\mathbf{A}$ are found via Eq. (1.4). XRF interpretation here identifies peaks that correlate with one another.

What is perhaps missed in the XRF analysis is that PCA is also effective at signal denoising [18]. Many denoising algorithms have been developed around PCA to remove noise in images along the spectral and spatial components that could be of use in XRF analysis. The application of PCA here would be no exception, although perhaps rudimentary by today's image processing standards in terms of complexity.

Later, Vekemans *et al.* [19] used a combination of PCA and K-means clustering to automatically extract correlations. The goal here was to identify distinct regions of similar XRF response. By doing so, the authors are able to find sum spectra of similar regions. This aided them in the identification of trace elements that may be lost in the overall sum spectrum.

To find these regions, the authors first perform principal component analysis (PCA) over the channels of the XRF data. This finds the eigenvectors (or equivalently eigenimages) of the covariance matrix. Then, they perform K-means clustering of the pixels of the first n principal eigenimages to find clusters that represent a combination of elemental XRF responses. This method helped them approach the goal of automatic segmentation: by segmenting the images, sum spectra of each cluster can be used to better detect trace elements. This was one of the earliest work that used a combination of spectral and spatial methods for XRF analysis.

Clustering-based methods have also been shown as a way to denoise the data as well [20]. By establishing representative cluster centroids, the data is mapped into a lower dimension where the noise is mitigated. In this case, spectral denoising is done through K-means clustering while spatial denoising is done via PCA.

Image processing is rich in PCA and clustering methods for denoising, although more common now is the use of neural networks, which are often not an option in XRF analysis due to the lack of available data. That being said, there are a plethora of image and spectral denoising algorithms in signal processing literature more advanced than PCA and K-means clustering. Some of these algorithms may be suitable for XRF analysis perhaps with some changes that incorporate prior XRF domain knowledge.

1.3.2. Nonnegative Matrix Factorization

While PCA-based techniques were popular, the results are not always physically interpretable since a PCA decomposition can contain negative values. What exactly would a negative amount of a spectrum indicate? All raw XRF data consists of nonnegative photon counts, so it is not clear at all how to interpret a PCA decomposition in a physical sense. This realization led to the additional constraint that both the dictionary and abundances must be nonnegative to provide feasible results.

Instead of PCA-based interpretation methods, Alfeld *et al.* [21] proposed using nonnegative matrix factorization (NMF) as a way to analyze the XRF correlations along the spectral dimension. NMF is a problem that directly addresses the constraints of Eq. (1.5) unlike PCA. Once the basis (dictionary) and abundance matrix are found, the results can be more readily interpretable as the values are all nonnegative. Despite the more realistic interpretation, the cost function typically evaluates higher than, say, that of a solution found through PCA. This is due to the additional constraint. The other downside with the general NMF algorithm is that the solution is non-deterministic and must go through an iterative optimization process. The dictionary D and abundance matrix $\mathbf{A}$ must be initialized to some nonnegative values before the NMF algorithm can proceed. After many iterations, the algorithm converges to a local minimum. Many trials with different initializations should be conducted to find the best local minimum.

NMF is often solved using an algorithm called Fast Non-Negative Least Squares (FNNLS) [22]. The method iteratively solves the following two equations until convergence:

$$\mathbf{A}^{k+1} \leftarrow \underset{\mathbf{A}^k \geq 0}{\arg\min} \|\mathbf{X} - D^k \mathbf{A}^k\|_F^2 \tag{1.6}$$

$$D^{k+1} \leftarrow \underset{D^k \geq 0}{\arg\min} \|\mathbf{X} - D^k \mathbf{A}^{k+1}\|_F^2 \tag{1.7}$$

where the superscript denotes the iteration number for $k \geq 0$ and $\|\cdot\|_F$ is the Frobenius norm. The entire, simple algorithm for solving Eqs. (1.6) and (1.7) individually is provided in the reference.

It was found in [21] that NMF does indeed provide more interpretable results with nonnegativity enforced. There was however some difficulty in distinguishing between different pigment groups that contain some of the same elements. This was mitigated by incorporating the additional information that some of the compounds were known a priori. NMF here was also done on some of the elemental maps instead of the XRF volume as a whole. Santos *et al.* [23] also used NMF, but on the entire XRF volume.

Again, just as PCA can smooth the data, so too can NMF. Perhaps one of the most noticeable differences is the nonnegativity constraint imposed on the NMF problem that introduces sparsity to some extent. There is a possibility that in the final spectral decomposition, D and $\mathbf{A}$ contain entries of 0. Sparse representations of $\mathbf{A}$ are known to be able to effectively denoise when redundancies exist in the data [24]. Intuitively, these zero-valued elements indicate the nonexistence of some XRF response. The absence of certain responses can be just as beneficial in XRF analysis as a response that is present [25].

1.3.3. Other Factorization Methods

NMF provides a nonnegative factorization of the XRF data, but in its native form does not require any other constraints be imposed. Take, for example, the issue of many solutions for the decomposition. This so-called rotational ambiguity is easily illustrated by choosing some nonsingular matrix $Q \in \mathbb{R}^{M\times M}$ [26]. The XRF decomposition of Eq. (1.1) can be rewritten as

$$\mathbf{X} \approx (DQ)\left(Q^{-1}\mathbf{A}\right). \tag{1.8}$$

Any invertible matrix Q can be chosen so long as the entries of DQ and $Q^{-1}\mathbf{A}$ are all nonnegative. No matter Q, there is no error added (or subtracted) to the original factorization. Additional constraints can be added in order to further constrain the solution.

1.3.3.1. Multivariate Curve Resolution–Alternating Least Squares. One constrained NMF algorithm that is employed is called multivariate curve resolution–alternating least squares (MCR-ALS) [26]. While this may not be familiar to those in the image processing community, this algorithm is in essence a nonnegative matrix factorization framework where certain additional constraints can be added. A popular constraint to include is that the final dictionary is a combination of the individual elemental spectra. As a least squares minimization, we have that

$$\left(A^{\text{mix}}\right)^*, \mathbf{A}^* = \underset{A^{\text{mix}},\mathbf{A}\geq 0}{\arg\min} \|\mathbf{X} - \left(D\,A^{\text{mix}}\right)\mathbf{A}\|_F^2 \tag{1.9}$$

where $A^{\text{mix}} \in \mathbb{R}_+^{M\times N}$ is the mixing matrix describing the linear combination of the individual elemental spectra D. Note that A^{mix} need not be a square matrix; in fact, it is the

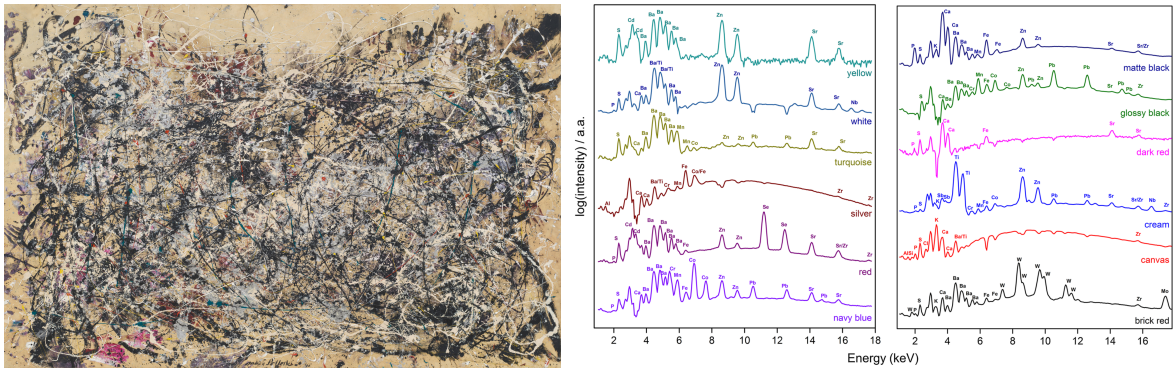


Figure 1.3. (Left) Pollock *Number 1A, 1948* (1948). Oil and household enamel paint on canvas (172.7 × 264.2 cm) The Museum of Modern Art. (Right) Plots of the dictionary after applying MCR-ALS. From [27].

case that $N < M$ to reduce the dimensionality of the dictionary and establish XRF correlations. The new dictionary is $DA^{\text{mix}} \in \mathbb{R}^{C \times N}$. The dimensions of $\mathbf{A}$ are also modified to $\mathbf{A} \in \mathbb{R}_{+}^{N \times H \times W}$ in order to have a valid matrix multiplication operation.

With A^{mix} known, dictionary DA^{mix} is quite easily interpreted as it is simply a linear combination of the original dictionary. This can be seen in the work by Martins *et al.* [27] where they analyzed a painting by Jackson Pollock seen in Fig. 1.3. Their analysis shows that the dominant colors can be identified based on the peaks of each spectrum in the dictionary. It however does not encompass a complete separation of the compounds present in the painting, which would be ideal considering the many overlapping paints that are characteristic of Pollock's work. Still, with only twelve dictionary endmembers, MCR-ALS is able to capture correlations that are present in the XRF spectra. The authors report that their criterion for selecting N is in part based on whether at least 95% of the variance can be explained.

We note that along with the other aforementioned techniques, MCR-ALS can be used to denoise the original data in addition to the XRF interpretation. Representing all

pixels as a linear combination of twelve spectra does not allow for much of the noise to be reconstructed without incurring more error in the overall cost of the objective function of Eq. (1.9).

1.3.3.2. Simplex Volume Maximization. Aside from MCR-ALS, Simplex Volume Maximization (SiVM) has also been proposed [28] to overcome computational speed issues of NMF and MCR-ALS. SiVM is also unique in that the dictionary is composed of data points in the XRF volume $\mathbf{X}$. These points are chosen from the set of points that lie on the convex hull of $\mathbf{X}$. Oftentimes the number of vertices of the hull is more than the desired number of endmembers in the dictionary. SiVM chooses the M extreme points such that the volume of the resulting simplex is maximized. These extreme points compose the dictionary, making XRF interpretation perhaps even easier than that of MCR-ALS since the basis can be readily found in the original data.

The authors reported some overlap in the elements present in the dictionary, but were still able to glean new information from the process that can be seen in Fig. 1.4. In particular, they noticed a glow around the hair in one of the bases that was not present in the individual elemental maps that highlights the intersection of paint that compose the hair (Ca) and the sky (Cu and Pb).

This method illustrates another way to select the dictionary, which has been shown to be a possible denoising algorithm [29]. The authors of this paper similarly note that hyperspectral unmixing is a form of denoising the data.

1.3.3.3. Dictionary Denoising of Poisson Data. A final method involves the combination of Poisson noise modeling and dictionary learning for the explicit purpose of denoising the XRF volume. We propose to use the Poisson negative log likelihood loss

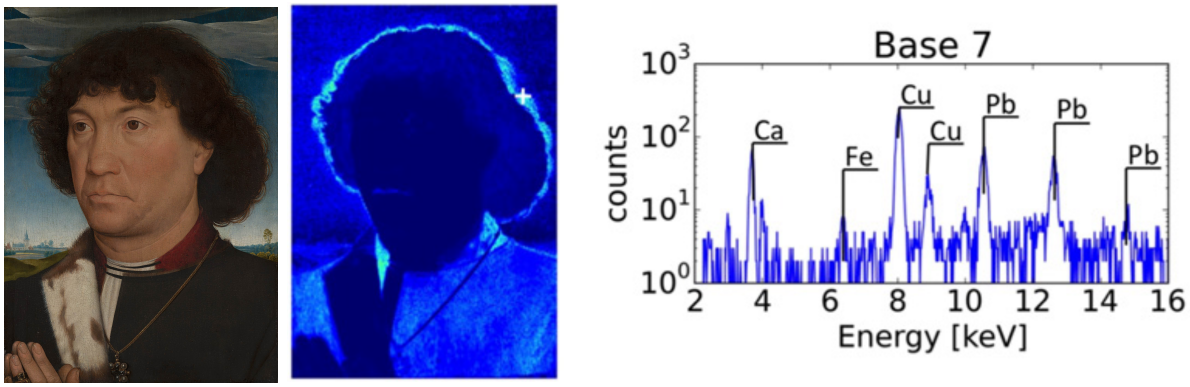


Figure 1.4. (Left) *Portrait of a man from the Lespinette family*, Hans Memling, 1485–1490, Mauritshuis, The Hague, Oil on panel, $30.1 \times 22.3\text{cm}^2$. (Middle) Abundance map of base 7. The base pixel is denoted by a yellow plus in the map. (Right) The spectrum of base 7. Adapted from [28].

(PNLL) when solving for the dictionary, particularly when the XRF pixels have relatively low photon counts [30]. This provides a better model for the noise instead of the ℓ_2 norm which assumes the data is drawn from a Gaussian model.

The PNLL loss is the average of the individual PNLL errors; the loss is defined by

$$\mathcal{L}_{\text{NLL}}\left(D\mathbf{A}, \mathbf{X}\right) = \frac{1}{N} \sum_{i \in (c,h,w)} (D\mathbf{A})_i - \mathbf{X}_i \cdot \ln\left(D\mathbf{A}\right)_i \tag{1.10}$$

where the summand is the PNLL error. Minimizing this loss is equivalent to maximizing the Poisson log likelihood. The objective function uses the PNLL instead of the ℓ_2 norm and is regularized using an adaptive total variation (TV) regularizer as well as an L0 penalty term on the abundance tensor to introduce sparsity, which is known to be an effective modeling strategy in smoothing data with redundancies [24].

They provide experiments comparing different denoising algorithms of fast XRF raster scans where the ground truth is known. The algorithm outperformed MCR-ALS as a

denoising method, both in terms of the error in the denoised XRF volume and the denoised elemental maps.

1.3.4. Perspectives

The dictionary-based XRF analysis algorithms reviewed here can be thought of as a method to denoise the data, even when this is not explicitly mentioned or studied in the papers present. Clearly, there is room for more advanced algorithms to arise for both XRF analysis as well as denoising. Perhaps by applying other denoising algorithms in image processing literature (or developing a method on one's own), better XRF analysis can be accomplished. The converse could also hold true whereby developing a new XRF analysis technique can decrease the noise present in the original data volume.

We encourage those that have applied dictionary decomposition methods in XRF imaging to revisit their techniques from a denoising perspective. Using the smoothed $D\mathbf{A}$ volume in place of $\mathbf{X}$, the original elemental maps can be found. Noisier maps with low count rates may appear smoother than before, and it would be an interesting study to see how well each of these methods performs as a denoiser.

We would also like to note that many of the algorithms for XRF interpretation only use spectral denoising techniques. It is well known in image processing literature that pixels in a local area generally have low variance in their values. This knowledge could be applied in the *spatial* domain in the form of a TV regularizer:

$$\mathcal{L}_{\text{reg}}(\mathbf{A}) = \mathcal{L}^{H}_{\text{reg}}(\mathbf{A}) + \mathcal{L}^{W}_{\text{reg}}(\mathbf{A}) \tag{1.11}$$

$$\mathcal{L}^{H}_{\text{reg}}(\mathbf{A}) = \sum_{m=1}^{M}\sum_{h=1}^{H-1}\sum_{w=1}^{W}\left(\mathbf{A}_{m,h+1,w} - \mathbf{A}_{m,h,w}\right)^2 \tag{1.12}$$

$$\mathcal{L}_{\text{reg}}^{W}(\mathbf{A}) = \sum_{m=1}^{M}\sum_{h=1}^{H}\sum_{w=1}^{W-1}\left(\mathbf{A}_{m,h,w+1} - \mathbf{A}_{m,h,w}\right)^2. \tag{1.13}$$

This term penalizes large changes in neighboring pixels. If we incorporate this regularizer into the objective function, we have

$$D^*, \mathbf{A}^* = \underset{D,\mathbf{A}\geq 0}{\arg\min}\ \mathcal{L}(\mathbf{X}, D, \mathbf{A}) + \alpha_{\text{reg}} \cdot \mathcal{L}_{\text{reg}}(\mathbf{A}) \tag{1.14}$$

for some scalar $\alpha_{\text{reg}} > 0$. Solving minimization problems of this form could improve how XRF signals are analyzed with the added prior knowledge, particularly in overcoming peak detection limits with weak XRF signals as seen in [30]. Using neighboring information could provide a better decomposition of the pixels individually.

Take the Memling painting in Fig. 1.4 which has many spatially smooth regions. It is reasonable to predict that most neighboring signals do not have wildly varying spectra. Adding TV regularization could provide a different optimized dictionary that takes these spatial relations into account. Of course, the TV term is not set in stone, and could be adapted.

An adaptation is likely needed in the Pollock painting of Fig. 1.3 for example, since there is a large amount of high spatial frequency. There are many different thin paint lines that intersect each other, so it is reasonable to assume that neighboring spectra vary quite a bit. The TV regularizer in Eq. (1.11) may inadvertently be detrimental to the results if the spatial resolution of the XRF image is low. Having an adaptive TV regularizer that identifies similar regions could be a way to modify the standard TV equation.

1.4. Spatial Deblurring, Super-resolution, and Inpainting Methods for XRF Volumes

Although XRF interpretation through known spectral denoising techniques is quite common in XRF literature, we have not yet focused on the spatial component of XRF image processing. There have been some published works on deblurring, super-resolution, and inpainting techniques applied to XRF data. The deblurring problem focuses on restoring high frequency components of the images lost in the data acquisition process. This is mainly an issue in Micro X-Ray Fluorescence (µ-XRF) where the spot size is more of a factor than in Macro X-Ray Fluorescence (MA-XRF) with paintings and other art objects.

In super-resolution and inpainting problems, some pixels are excluded or missing from measurements for some given reason. Following hard acquisition time constraints may be a factor that affects the spatial resolution. Instead of capturing the XRF data at the resolution that one desires, time can be saved by decreasing the resolution or sampling select areas.

In this section, we'll review the different image processing techniques that have been applied/developed for improving XRF imaging in the spatial domain.

1.4.1. Fourier-based Deblurring Approaches

General algorithms for image deblurring already exist and can be readily applied to XRF data. Yang *et al.* [31] conducted an experiment on some elemental maps that compare four different Fourier transform-based super-resolution methods: (1) Wiener deconvolution, (2) Richardson-Lucy [32], (3) Fast iterative shrinkage-thresholding algorithm (FISTA) [33],

and (4) blind deblurring with L0-regularized intensity and gradient prior (L0RIGP) [34]. The first three algorithms are non-blind methods, *i.e.* the point spread function (PSF) is known. L0RIGP is a blind method where the PSF is not known. Since the authors reported that the Richardson-Lucy algorithm performed the best for their µ-XRF datasets, we only review this method and refer the reader to Yang *et al.* 's work [31].

Let $Z, L \in \mathbb{R}_{+}^{H \times W}$ be the underlying super-resolved elemental map and the acquired elemental map respectively. With a PSF P, Z can be approximated by

$$Z \approx L * P \tag{1.15}$$

where $*$ denotes convolution. Richardson-Lucy is an algorithm that can notably be used to deblur images with Poisson noise. The algorithm is iterative and converges at the maximum likelihood solution given the PSF:

$$Z^{k+1} = Z^k \cdot \left(\hat{P} * \frac{L}{P * Z^k} \right) \tag{1.16}$$

where $\hat{P}$ is the "flipped" version of P whereby the rows and columns are reversed. Element-wise multiplication is denoted by $\cdot$, and division is carried out element-wise as well.

The authors tested this method on the Fe Kα, Ca Kα, and Si Kα elemental maps of an iron skarn, which is known to have sharp boundaries that are difficult to capture with µ-XRF imaging. These results are shown in Fig. 1.5. While the Fe Kα and Ca Kα maps have sharper boundaries as they expected, the Si Kα maps shows more degradations than there should be. They attribute this to the high noise in the Si Kα image due to the chemical nature of the quartz (which is composed of Si) as it interacts with x-rays. The

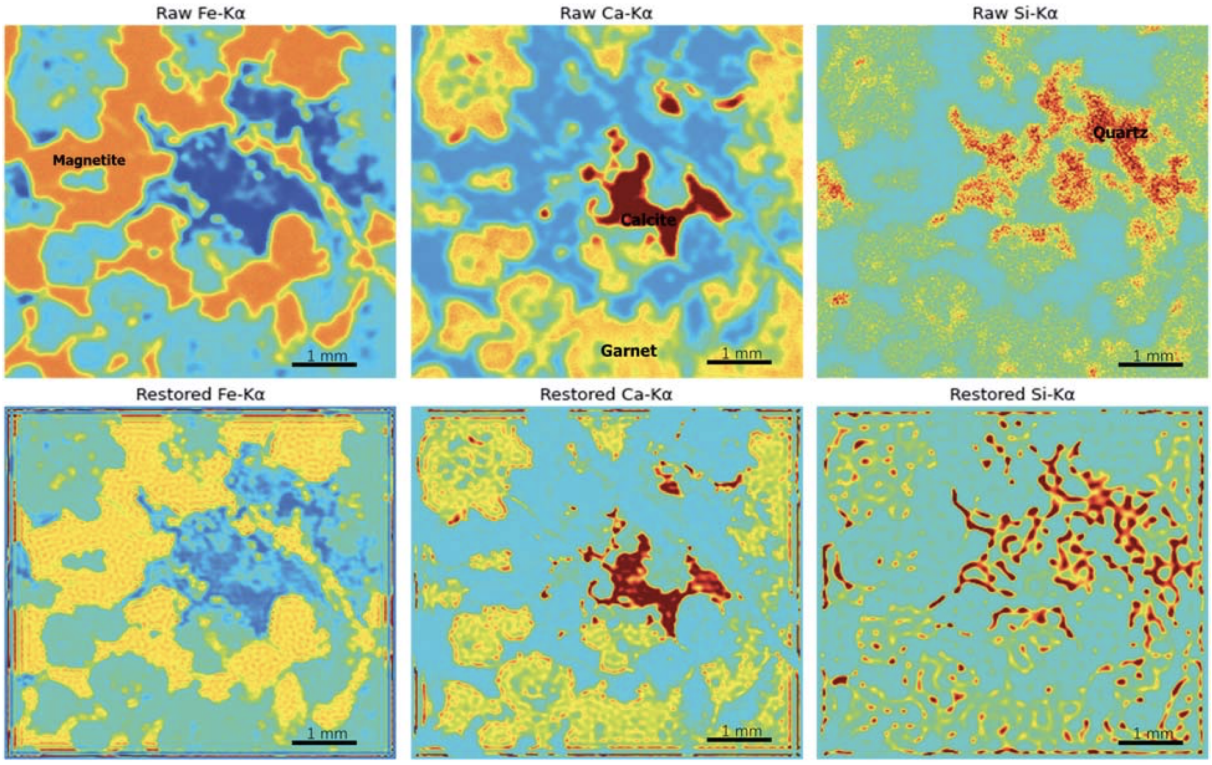


Figure 1.5. Elemental maps of an iron skarn. (Top row) Raw maps of Fe Kα, Ca Kα, and Si Kα respectively. (Bottom row) Denoised maps of Fe Kα, Ca Kα, and Si Kα respectively. From [31].

authors conclude that the Richardson-Lucy algorithm can be applied to elemental maps that have high XRF responses, but is not as effective with elements that are harder to detect.

1.4.2. Super-Resolution and Inpainting Approaches

The goal of super-resolution is to predict the value of certain pixels that were not measured. Here, the object is sampled along a uniform rectangular grid, which composes the low resolution image. This is in contrast to generalized inpainting methods which are not classically constrained to be sampled in a uniform fashion. Whether due to time

constraints or x-ray spot size, the resolution needs to be increased for improved spatial analysis.

Dai *et al.* [35] proposed a sparse dictionary based method for super-resolution of the XRF volume, $\mathbf{X} \in \mathbb{R}_{+}^{C \times H_l, W_l}$, using information from a high resolution color image of a painting, $\mathbf{I} \in [0,1]^{3 \times H_s \times W_s}$. It is understood that $H_l < H_s$ and $W_l < W_h$ to conform with the super-resolution problem. They propose to first separate the low resolution XRF data $\mathbf{X}$ and the super-resolved XRF image $\mathbf{Y} \in \mathbb{R}^{C \times H_s \times W_s}$ into two components: a visible $(\cdot)_v$ and non-visible component $(\cdot)_{nv}$ such that

$$\mathbf{X} = \mathbf{X}_v + \mathbf{X}_{nv} \tag{1.17}$$

$$\mathbf{Y} = \mathbf{Y}_v + \mathbf{Y}_{nv}. \tag{1.18}$$

The visible component is defined as the portion of the XRF signal attributed to the surface response of the painting. The non-visible component contains the XRF signal that originated from underneath the painting's top layer of paint.

The purpose of separating the XRF signal is to establish a relationship with the RGB signal, which only images the top layer of paint. They propose using a dictionary decomposition of $\mathbf{Y}_v$, $\mathbf{Y}_{nv}$, and $\mathbf{I}$:

$$\mathbf{Y}_v = D_v^{xrf}\, \mathbf{A}_v \tag{1.19}$$

$$\mathbf{Y}_{nv} = D_{nv}^{xrf}\, \mathbf{A}_{nv} \tag{1.20}$$

$$\mathbf{I} = D^{rgb}\, \mathbf{A}_v. \tag{1.21}$$

Notice the abundance $\mathbf{A}_v$ is shared across the XRF and RGB domains. The amount of contributions from the XRF and RGB domains at each pixel is proposed to be equal.

Further, the low and high resolution XRF volumes are related by a binary sampling matrix $\mathbf{T} \in \{0,1\}^{1\times H_s \times W_s}$ by

$$\mathbf{X}_v = \mathbf{Y}_v \cdot \mathbf{T} = \left(D_v^{xrf}\, \mathbf{A}_v\right) \cdot \mathbf{T} \tag{1.22}$$

$$\mathbf{X}_{nv} = \mathbf{Y}_{nv} \cdot \mathbf{T} = \left(D_{nv}^{xrf}\, \mathbf{A}_{nv}\right) \cdot \mathbf{T} \tag{1.23}$$

where $\cdot$ is an element-wise multiplication and $\mathbf{T}$ multiplies each channel of the multiplicand.

The optimization problem is then

$$\begin{aligned} \underset{\substack{\Theta\geq 0 \\ \|\mathbf{A}\|_0 \leq s}}{\arg\min}\ & \left\|\mathbf{X} - \left(D_v^{xrf}\mathbf{A}_v\right)\cdot\mathbf{T} - \left(D_{nv}^{xrf}\mathbf{A}_{nv}\right)\cdot\mathbf{T}\right\|_F^2 \\ & + \left\|\mathbf{I} - D^{rgb}\,\mathbf{A}_v\right\|_F^2 + \alpha_{\text{reg}}\cdot\mathcal{L}_{\text{reg}}\left(D_{nv}^{xrf}\,\mathbf{A}_{nv}\right) \end{aligned} \tag{1.24}$$

where set $\Theta = \{D_v^{xrf},\, D_{nv}^{xrf},\, D^{rgb},\, \mathbf{A}_v,\, \mathbf{A}_{nv}\}$, abundance $\mathbf{A} = \mathbf{A}_v + \mathbf{A}_{nv}$, and $\|\cdot\|_0$ is the ℓ_0 pseudonorm. This equation is constrained to have a sparse representation, which has been shown to have smoothing effects when there are redundancies in the data [24]. The first two terms of Eq. (1.24) are fidelity terms, and the third is the TV regularizer from Eq. (1.11) that captures spatial correlations. We refer the reader to the original paper to learn how to minimize this complicated objective function.

Their solution provided better super-resolution results than the other methods designed for hyperspectral images. Fig. 1.6 shows a super-resolved XRF image of *The*

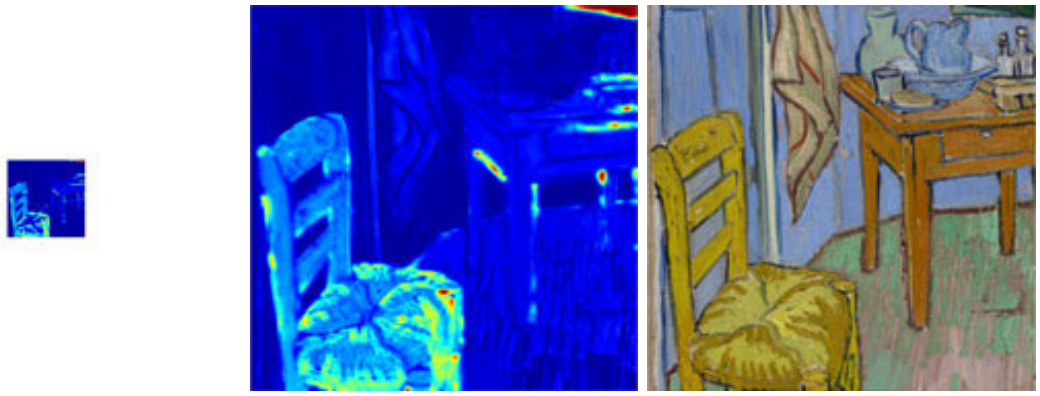


Figure 1.6. (Left) The low resolution XRF image. (Middle) The super-resolved XRF image. (Right) The high resolution RGB image: *The Bedroom*, Vincent van Gogh, 1853–1890, Arles, October 1888, oil on canvas, 72.4×91.3 cm^2. Adapted from [35].

Bedroom by Vincent van Gogh. They were able to capture the XRF response of the curtain in a more accurate manner than the other methods that either filter the curtain out or add artifacts to the map.

As was mentioned earlier, inpainting is closely related to the super-resolution problem since both problems require estimation of unknown pixels values. Dai *et al.* [36] further pushed their optimization algorithm to be applied to any sampling matrix. The sampling matrix will be discussed in the next section, but the updated optimization algorithm introduces a new penalty term. They postulate that the gradient of the visible component of the XRF volume should be similar to the gradient of the RGB image. This is included in the optimization algorithm of Eq. (1.24) as a weighted TV regularizer $\mathcal{L}_{\text{reg}}\left(D_v^{xrf}\mathbf{A}_v;\mathbf{I}\right)$. The new penalty term weighs the TV loss based on the RGB image gradient. Within the summand of the TV Eqs. (1.12) and (1.13), a multiplier controls the penalty based on the spatial location. In low varying areas of the RGB image, they expect the visible XRF component is low varying as well. Thus, a relatively higher penalty in the smoothing is

applied on pixels with like neighbors. On the other hand, along the edges of the RGB image, there is high contrast; relatively low penalties should be applied here because high variation is expected that should not be smoothed. We refer the reader to the work [36] for the full optimization and a definition of the adaptive TV regularizer.

These methods provide a good RGB-XRF fusion method for super-resolution and inpainting, although the algorithm was recently improved upon by Su *et al.* [37]. They make a slight change in the framework from Dai *et al.* [35]: instead of separating the XRF volume into visible and non-visible components, they propose to separate the volume by common and unique components. This slight change in thinking introduces a new decomposition scheme:

$$\mathbf{X} = \mathbf{X}_c + \mathbf{X}_u = \left(D_c^{xrf}\mathbf{A}_c\right)\cdot\mathbf{T} + \left(D_u^{xrf}\,\mathbf{A}_u^{xrf}\right)\cdot\mathbf{T} \tag{1.25}$$

$$\mathbf{I} = \mathbf{I}_c + \mathbf{I}_u = D_c^{rgb}\mathbf{A}_c + D_u^{rgb}\,\mathbf{A}_u^{rgb} \tag{1.26}$$

where $(\cdot)_c$ denotes the common components and $(\cdot)_u$ denotes the unique components. There are also new and updated terms in their objective function:

$$\begin{aligned}\underset{\Theta\geq 0}{\arg\min}\ & \left\|\mathbf{X} - D_c^{xrf}\mathbf{A}_c\cdot\mathbf{T} - D_u^{xrf}\mathbf{A}_u^{xrf}\cdot\mathbf{T}\right\|_F^2 \\ & + \left\|\mathbf{I} - D_c^{rgb}\mathbf{A}_c - D_u^{rgb}\mathbf{A}_u^{rgb}\right\|_F^2 \\ & + \alpha_{\text{reg}}\cdot\mathcal{L}_{\text{reg}}\left(D_u^{xrf}\mathbf{A}_u^{xrf}\right) \\ & + \alpha_{\text{MI}}\cdot\text{MI}\left(\mathbf{I}\cdot\mathbf{T},\ D_u^{xrf}\mathbf{A}_u^{xrf}\cdot\mathbf{T}\right)\end{aligned} \tag{1.27}$$

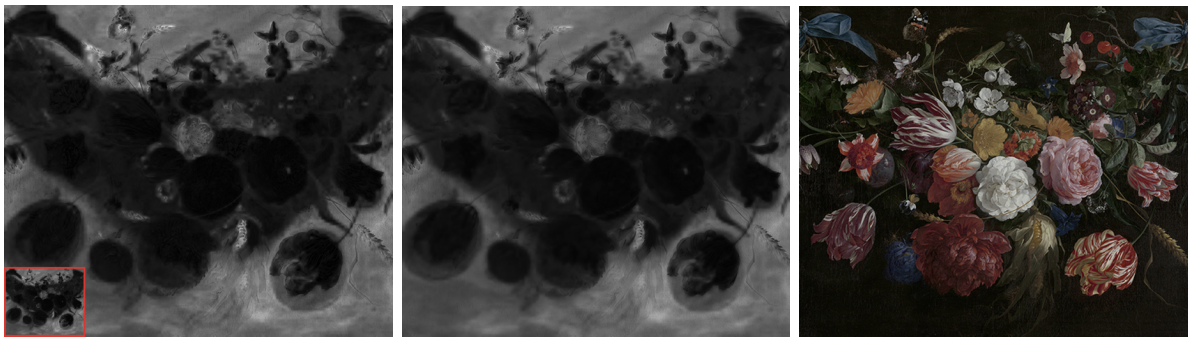


Figure 1.7. (Left) The low resolution Fe map and high resolution ground truth Fe map. (Middle) The super-resolved Fe map. (Right) The high resolution RGB image: *Bloemen en Insecten*, Jan Davidsz. de Heem, 49 × 67cm, Royal Museum of Fine Arts Antwerp, inv. no. 54, oil on canvas. Adapted from [37].

where $\Theta = \{D_c^{xrf}, D_u^{xrf}, D_c^{rgb}, D_u^{rgb}, \mathbf{A}_c, \mathbf{A}_u^{xrf}, \mathbf{A}_u^{rgb}\}$. The first two terms of Eq. (1.27) are fidelity terms similar to that of Eq. (1.24). The third term is again a TV regularizer term that penalizes high spatial variation in neighboring pixels. The last term MI$(\cdot)$ is novel in that it penalizes the estimated amount of mutual information shared between the unique components between the low resolution XRF and RGB data. Notice that the sparsity constraint was dropped as well, which perhaps is replaced by the mutual information loss to ensure separation of the representations. We refer the reader to Su *et al.* 's work [37] and Kraskov *et al.* 's work on estimating mutual information [38] from which the penalty term was derived.

They performed experiments showing that their method outperforms Dai *et al.* 's work on the painting *Bloemen en Insecten* by Jan Davidsz. de Heem. Their results are shown in Fig. 1.7. They also report a 2.42 dB increase in the Peak Signal-to-Noise Ratio (PSNR) to reach 47.71 dB.

Super-resolution and inpainting of XRF image volumes using joint dictionary optimization methods prove to be quite effective in estimating an XRF cube in higher dimensions. Whether the dictionaries are found from a sparsity lens or a mutual information standpoint, these early methods are already powerful. Perhaps a joining of these two ideas can reduce the reconstruction error even further.

1.5. XRF Subsampling Design

The last category we will address is the design of subsampling algorithms. Subsampling is a staple in signal processing. In the XRF image setting, the goal of this problem is to find a subsampling pattern that reduces the total acquisition time of the XRF data. Since the scan time is quite long in many cases, quickly acuiring this data without sacrificing quality is starting to emerge in the literature.

While not a method to find an optimal sampling pattern, we briefly note the work by us that provided some analysis on subsampling as it relates to the dwell time [30]. Using simulated fast XRF scans of *Bloemen en Insecten*, they tested the limits of how short the dwell time could be without incurring too much error. Scan times that are 20 times as fast as the original scan were reported as a possibility, which would greatly speed up the XRF acquisition process.

In an effort to break from the raster scan, three existing methods are used to find the optimal sampling pattern: (1) manual mask design, (2) convolutional neural networks (CNNs), and (3) reinforcement learning (RL). We will cover each in this section.

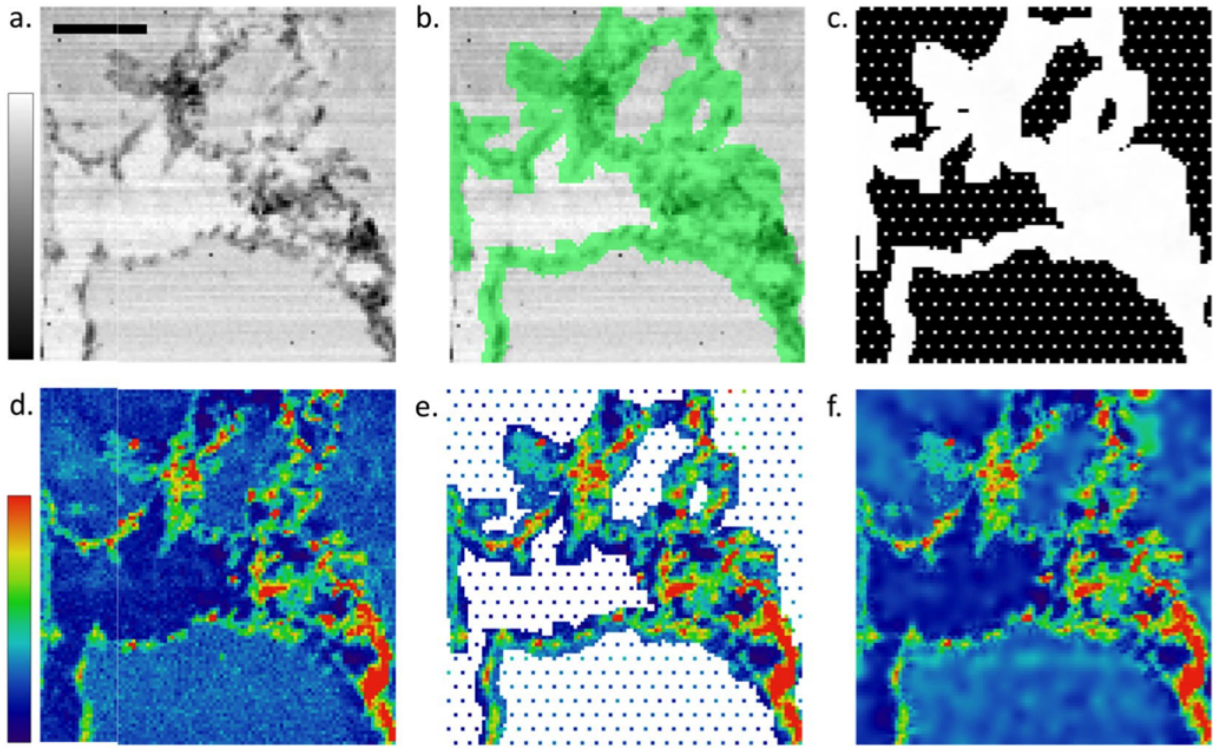


Figure 1.8. The hand-selected mask algorithm. (a) A quickly acquired XRF map of Si. (b) The region of interest. (c) The sampling mask with added sparse uniform sampling. (d) The Si map of a full raster scan. (e) The Si map of the masked scan, found approximately 3 times faster than the full raster scan. (f) The inpainted Si map of (e) using biharmonic inpainting.

1.5.1. Manual Mask Design

One simple approach to take is designing the sampling mask on one's own. Kourousias *et al.* [39] provided a proof of concept whereby they imaged a sampled to improve the quality of the Si map.

First, the XRF data is rapidly acquired, which results in blurry elemental maps. Next, using the blurry images as a guide (in this case Si only), the user manually selects the regions of interest for further scanning. A sparse, uniform sampling pattern is overlaid on the mask so that the background is not completely ignored. These locations are scanned

again. An inpainting algorithm then estimates the pixels that were not scanned. The authors provide an example in Fig. 1.8.

This approach is able to better capture the foreground, but it certainly has its limitations. First, the algorithm ignores the initial scan in the final reconstruction. The information may be blurry or noisy, but it can still be used as prior information to aid in the inpainting algorithm.

Additionally, there is a possibility of human error when some of the foreground is mistakenly excluded. A different display method (e.g. linear vs. log scale) may highlight details in the first maps that could be undetected by human vision.

This is where calculated methods come into play. Aside from automating the mask design (and therefore further reducing the total acquisition time), an algorithmic approach may not only select different areas, but also provide insight into how to allocate dwell times that vary per pixel.

1.5.2. Convolutional Neural Networks

Convolutional neural networks have grown in popularity over the past decade due to the increase in computational power on computers and their proven effectiveness at solving tasks. CNNs are excellent for extracting correlations along the dimensions that they are applied along. When designing a sampling mask, the spatial correlations are exploited to find the best pixels to sample. These selected pixels are typically the most difficult ones to estimate should they have been excluded from the set of samples.

Dai *et al.* [36] sought to bring neural networks to find the best sampling mask. To do so, they introduced a convolutional network network (CNN) NetM whose purpose is

to find the optimal sampling mask $\mathbf{T}$ for some given image $\mathbf{I}$ and sampling rate $r \in (0, 1)$ which determines the fraction of pixels to sample. This is tricky to accomplish since there are no ground truth sampling masks available. Without these training pairs, the neural network cannot be trained. To circumvent this issue, another CNN they call NetE was appended to the output of the mask network. NetE requires a subsampled RGB image as input that is inpainted at the output.

The inpainting network was trained first separate from the mask generating network using the ℓ_2 norm as a loss function:

$$\Theta_E^* = \arg\min_{\Theta_E} \ \|\mathbf{I} - \text{NetE}\left(\mathbf{I} \cdot \mathbf{T}; \Theta_E\right)\|_F^2 \tag{1.28}$$

where Θ_E are the neural network weights.

Once this network is trained, NetM can be trained with a fixed NetE in a feedforward manner. Instead of providing $\mathbf{T}$, it is instead estimated via NetM. The loss function for training NetM is

$$\Theta_M^* = \arg\min_{\Theta_M} \ \|\mathbf{I} - \text{NetE}\left(\mathbf{I} \cdot \text{NetM}\left(\mathbf{I}; r, \Theta_M\right)\right)\|_F^2 \tag{1.29}$$

where Θ_M are the neural network weights for NetM. At inference time, the inpainting network NetE is dropped. The output of NetM is not binary; the final mask is instead drawn from a Bernoulli distribution where the pixel values of the mask is the probability of drawing a 1. This binarizes the output with the desired rate. Fig. 1.9 shows a sample random mask and a mask from NetM when *Bloemen en Insecten* (see Fig. 1.7, Right) is the input.

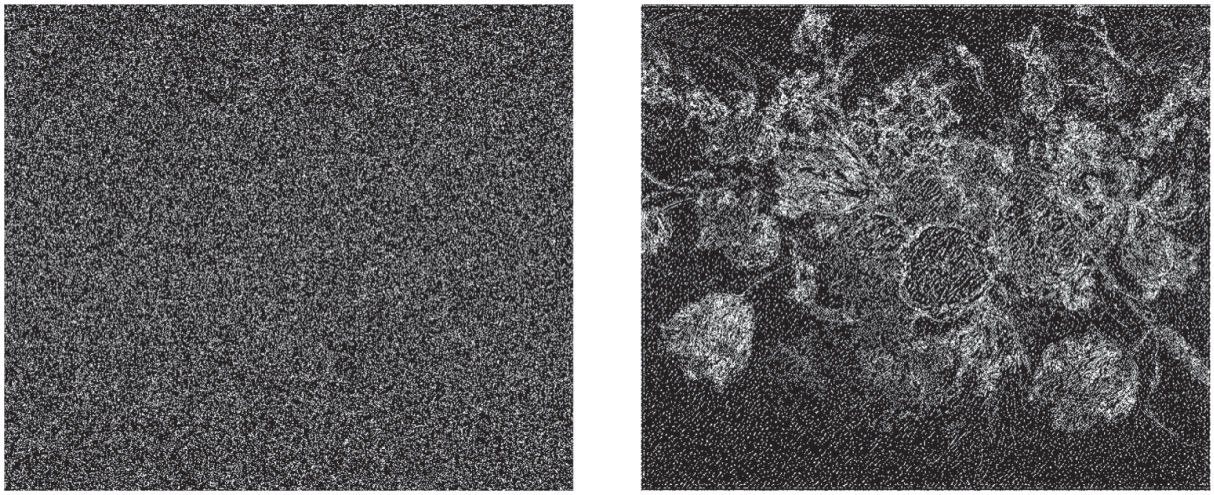


Figure 1.9. (Left) A random uniform mask with 20% of the pixels selected, denoted with white. (Right) A NetM mask of *Bloemen en Insecten* also with 20% of the pixels selected. From [36].

Notice that the background is sparsely sampled due to its uniformity. The remaining samples are concentrated in the foreground where the flowers are located, which is intuitively where one would place the samples. This mask was shown to have improvements for inpainting over other algorithms that solve for an optimal sampling mask. The results were consistent over different sampling rates as well.

1.5.3. Reinforcement Learning

Reinforcement learning, much like CNNs, has gained in popularity for solving tasks related to action taking. The general framework of reinforcement learning is based on a reward structure. With each action that is taken, a reward (or penalty) is calculated. The larger the reward at the end of an episode (*i.e.* a complete set of actions), the more the network will learn to take similar steps that previously rewarded it with a high score.

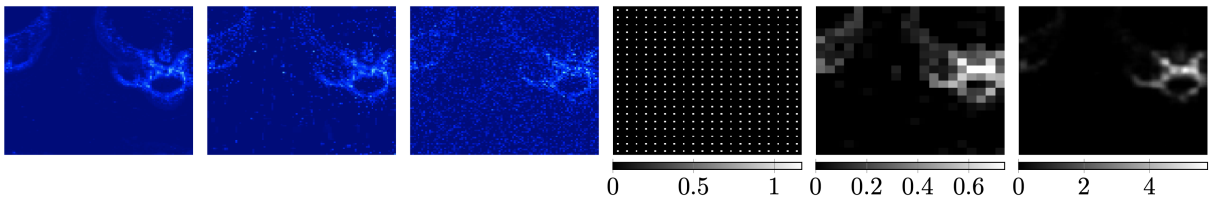


Figure 1.10. From left to right: (1) The ground truth XRF sample; (2) The XRF estimation using the method of [40]; (3) The XRF estimation using a raster scan of equal time; (4) The initial scan; (5) The next adaptive scan; (6) The final adaptive scan. The scale in (4)–(6) denotes the scan time. Adapted from [40].

Reinforcement learning is typically framed as a maximization problem, but this is not always the case.

Betterton *et al.* developed a RL algorithm for XRF sampling. They formulated a unique sampling method that uses different apertures for the x-ray beam. This is to first capture the general scene in a quick manner, then allot more time on the areas of interest with more focused apertures. The scan times for local areas of the object are optimized with each aperture.

There are two objectives to jointly minimize in their formulation: (1) the main objective, $\mathcal{L}_Q$, which is to provide the best quality image at the end of the scans, and (2) the amount of time spent scanning, $\mathcal{L}_C$, is penalized if it is excessive. A simplified representation of their algorithm for training tries to find a path τ_k for each time step $k \in \{1, \ldots, K\}$ that minimizes the expected sum of the quality penalty and time penalties:

$$\tau^* = \arg\min_{\tau} \mathbb{E}\left[\mathcal{L}_Q(\tau_K) + \alpha_C \sum_{k=1}^{K} \mathcal{L}_C(\tau_k)\right]. \tag{1.30}$$

The first term only penalizes the deviation of the XRF estimation from the ground truth after the final time step. The second term penalizes the time taken different for each of

the K trajectories. Hyperparameter α_C controls how much the time taken is penalized relative to the XRF fidelity term.

Fig. 1.10 shows results on a cropped region of an XRF scan. The RL approach has both qualitative and quantitative improvements over the raster scanning method. More noise is present in the raster scan, and the foreground is better estimated as well. This is attributed to how the algorithm allocates more time to the foreground and avoid the background.

These results are undoubtedly impressive, but may be hard to adapt to on a large scale. The resolution of the ground truth is 50×50 px, which is small for XRF volumes now that typically are on the order of hundreds of pixels in height and width. Reinforcement learning problems are hard to optimize normally, not to mention the lack of XRF data needed to train large scale networks.

1.6. Conclusion

The field of XRF imaging is quite young. Yet despite its youth, many paintings and other art objects have been studied by many groups using many techniques. It is only recently that efforts are being made to join image processing research with XRF research.

We first introduced XRF imaging and analysis particularly for the signal processing community to become acquainted with this new imaging modality—establishing goals and problems that are faced by researchers in the XRF field. We then provided a take of XRF analysis through the lens of image and signal processing particularly as it relates to denoising techniques. Many XRF analysis methods can also be classified as denoising algorithms (mostly denoising in the spectral domain). This aspect, we believe, should see

more attention as the metaphorical field is ripe for research. No matter if the goal is to improve how XRF signals are analyzed or denoise the data, we encourage cross collaboration between the two disciplines to provide perspectives on one another's research.

In addition, we also reviewed some existing techniques that are classical problems in the image processing community, namely deblurring, super-resolution and inpainting, and signal subsampling. We hope that those in the XRF community can see the value of applying these algorithms to their own work, whether out of analytical necessity or just to collect more data in a faster manner.

The lack of published algorithms is apparent, but we hope that the methods highlighted here serve as stepping stones for improving the aspects of XRF acquisition and analysis. Again, this will require cross collaboration in order for the image processing community to understand the needs and concerns of the XRF imaging community and vice-versa.

With time, these paintings fade and degrade. The sooner we as a collective group can develop better tools for XRF analysis and acquisition, the more of these timeless pieces we can understand and potentially delay their degradation.

CHAPTER 2

Denoising the Volumes of Fast X-Ray Fluorescence Raster Scans of Paintings

Abstract

XRF imaging of cultural heritage objects, while a popular non-invasive technique for providing elemental distribution maps, is a slow acquisition process in acquiring high signal-to-noise ratio XRF volumes. Typically on the order of tenths of a second per pixel, a raster scanning probe counts the number of photons at different energies emitted by the object under x-ray illumination. In an effort to reduce the scan times without sacrificing elemental map and XRF volume quality, we propose using dictionary learning with a Poisson noise model as well as a color image-based prior to restore noisy, rapidly acquired XRF data.

2.1. Introduction

In the growing field of applying scientific methods to cultural heritage research, XRF imaging is frequently used as a non-invasive tool to analyze works of art. This approach leverages the insights gained from XRF point analysis in providing elemental information on a per-pixel basis. These elemental distribution maps provide information as to what chemical elements compose the layers of paint. With these maps for example, an art

Figure 2.1. Jan Davidsz. de Heem's *Bloemen en Insecten*, 49×67 cm, Royal Museum of Fine Arts Antwerp, inv. no. 54, oil on canvas.

conservator can better preserve paintings [7], or an art historian can deduce an artist's painting techniques—sometimes revealing hidden paintings [4].

In XRF imaging, a source excites a small target area of the painting by irradiating it with x-rays. An inner orbital electron can be ejected if the impinging x-ray has greater energy than the electron's binding energy. An electron at an outer orbital then drops to fill the inner orbital vacancy by emitting a photon of energy equal to the energy difference of the orbitals. Each element has characteristic orbital energy levels (and therefore a characteristic XRF spectrum). A detector and digital post processor records and bins each photon according to its energy.

While macro XRF is a powerful, increasingly popular technique, acquiring elemental maps for entire paintings with good signal-to-noise ratios often translates to long acquisition times. Depending on the painting size, spot size, and dwell time, it can take many hours or even days to acquire the XRF volume. Take as an example a painting of modest

size $600 \times 720\,\mathrm{mm}^2$. If we specify a scan with spot size $1\,\mathrm{mm}^2$ and dwell time $200\,\mathrm{ms/px}$, it would take exactly 1 day to scan. There are two problems in these long scan times since (1) access to paintings often occur in short time windows when they are off-view, en route to other sites, etc., and (2) the x-ray exposure time should be minimized to best preserve the painting.

Analysis of XRF volumes uses photon count *rates* instead of photon counts, as the dwell time can vary by scan. These volumes are then separated into elemental maps using a least squares fit where the feature matrix is composed of known elemental XRF responses. Before collecting XRF data, a trade-off between image quality, such as the root mean-square error (RMSE), and time must be taken into account: the longer the dwell time, the more accurate the measured photon count rates from the count-limited photon data. Our goal here is to develop an XRF denoising algorithm where we test it on simulated scans at different dwell times based on real XRF data. We focus on Jan Davidsz. de Heem's *Bloemen en Insecten* as shown in Fig. (2.1), the data of which has been generously shared by de Keyser *et al.* [41].

2.2. Related Work

Dictionary learning approaches frequently appear in XRF literature since each element emits a characteristic set of discrete fluorescent lines. Limiting the number of spectral representations to the number of elements makes intuitive sense, as each pixel is then a linear combination of different elemental spectra. Martins *et al.* proposed denoising XRF volumes using multivariate curve resolution-alternating least squares (MCR-ALS),

a simple dictionary learning approach in the spectral domain to separate elemental compositions [27, 42]. Kogou *et al.* used an unsupervised learning method called self-organizing maps (SOMs) that also extracts a set of spectral dictionary atoms to decompose the XRF volumes into a representative basis [43]. This method effectively uses k-means clustering to generate the set of dictionary endmembers. More elaborate dictionary methods have been explored by Dai *et al.* whereby joint RGB and XRF dictionaries inpaint a spatially selective subsampled XRF volume [44].

Even though photons arrive according to a Poisson process [45], each of these methods (implicitly) uses a Gaussian noise model since the dwell times are assumed to be long. This noise model was shown to be a good approximation in XRF denoising due to the central limit theorem and Poissonian data resembling Gaussian data at high count rates. This assumption, however, can break down with short dwell times when Gaussian noise is no longer an accurate approximation as our experiments show.

PURE-LET from Luisier *et al.* is an algorithm specifically for Poisson image denoising that minimizes the Poisson unbiased risk estimate in the Haar-wavelet domain [46]. This method was originally published using tests on conventional images, MRI brain data, and fluorescence-microscopy of biological samples. To the best of our knowledge, it has not been applied to XRF data, but is another tool that can be used as it partially addresses the concerns of current dictionary learning approaches for XRF denoising.

Our method merges the best characteristics of the two solution approaches: a spectral dictionary learning approach with a Poisson model (instead of a Gaussian model) for denoising XRF volumes. An RGB image prior and sparsity coding are also used to denoise the data.

2.3. Algorithm

Assume for now that we have the XRF count volume from a fast raster scan, $\mathbf{X} \in \mathbb{N}^{C\times H\times W}$ of channels (*i.e.* energy bins) C, height H, and width W. Each pixel has an identical dwell time $t \in \mathbb{R}_+$ where $\mathbb{R}_+$ is the set of nonnegative real numbers. The photons arrive with unknown underlying photon arrival rate $\mathbf{\Psi} \in \mathbb{R}_+^{C\times H\times W}$. Additionally, assume we have an RGB image of the painting $\mathbf{I} \in [0,1]^{3\times H\times W}$ registered with the XRF data. We want to estimate $\mathbf{\Psi}$ using $\mathbf{X}$, $\mathbf{I}$, and t in our optimization formula detailed here. Keep in mind that t is short relative to a standard raster scan. Standard scans often spend on the order of low hundreds of milliseconds per pixel, but we will test values of t at least 5x faster.

2.3.1. Formulation

Before beginning, we need to define the relationships between our data. During the sampling process, Poisson noise is introduced into our scan via

$$\mathbf{X} \sim \text{Poiss}\left(\mathbf{Y}\right) \tag{2.1}$$

where

$$\mathbf{Y} = t \cdot \mathbf{\Psi} \tag{2.2}$$

is the average number of photons we expect to record given the dwell time t. Note that the longer the scan is conducted, the more photons will be recorded. At shorter and shorter dwell times, it can even become unlikely to record a photon that would otherwise

likely be present with a standard dwell time. This is the targeted time regime we propose for our denoising algorithm.

2.3.1.1. Data Fidelity Derivation. Recall that the XRF signal is a combination of elemental spectra. Each element has its own unique XRF response that we can exploit for sparse coding, which has been shown to be effective in signal denoising [24]. We can use dictionary learning as a means to deconstruct our XRF signal into a finite set of signals and corresponding abundances for each pixel. Let $D \in \mathbb{R}_+^{C\times M}$ be the nonnegative dictionary with M endmembers representing spectral responses, and let $\mathbf{A} \in \mathbb{R}_+^{M\times H\times W}$ be the nonnegative sparse abundance volume. We need to ensure that the entries are nonnegative since negative spectral responses and negative abundances have no physical meaning. The XRF signal is decomposed via

$$\mathbf{X} \approx D\mathbf{A} \tag{2.3}$$

where the matrix-tensor multiplication is carried out via

$$(D\mathbf{A})_{c,h,w} = \sum_m D_{c,m} \cdot \mathbf{A}_{m,h,w}. \tag{2.4}$$

Learning the dictionary D and abundance matrix $\mathbf{A}$ provides both a spectrally smooth XRF volume and a more accurate representation of the chemical processes governing XRF data acquisition.

When scanning each pixel, photons of different energies arrive according to a Poisson sum model, which can be split into multiple independent Poisson processes [47]. Each pixel we also assume to be spatially independent from one another for now. It follows that each entry of $\mathbf{X}$ is independent from one another as well.

In denoising problems, a data fidelity term is often needed so that the output does not deviate too far from the original data. Since our data is Poissonian, we use the Poisson negative log likelihood (PNLL) loss as the data fidelity term:

$$\mathcal{P}_{\mathrm{NLL}}\left(\mathbf{X}_i, \mathbf{\Psi}_i\right) = \mathbf{X}_i - \mathbf{Y}_i \cdot \ln\left(\mathbf{X}_i\right) \tag{2.5}$$

where i represents any index. The PNLL differs from the mean-squared error (MSE) in that it is a relative error that accounts for the variance of the Poisson distribution, which is identical to the mean. For example, the MSE of $\mathbf{X}_i + a$ and $\mathbf{Y}_i + a$ is the same regardless of the value of a. The PNLL on the other hand decreases with increasing a since we expect large absolute errors at large magnitudes.

Since $\mathbf{Y}$ is unknown, we instead try to best match the dictionary decomposition results of Eq. (2.3) with the data we measured:

$$\mathcal{P}_{\mathrm{NLL}}\left(\left(D\mathbf{A}\right)_i, \mathbf{X}_i\right) = \left(D\mathbf{A}\right)_i - \mathbf{X}_i \cdot \ln\left(\left(D\mathbf{A}\right)_i\right) \tag{2.6}$$

which combines Eq. (2.3) and Eq. (2.5) together. As a loss function, we can compute the mean PNLL given our data, which uses our assumption that all $\mathbf{X}$ entries are independent of each other:

$$\mathcal{L}_{\mathrm{NLL}}\left(D\mathbf{A}, \mathbf{X}\right) = \frac{1}{N} \sum_{i \in (c,h,w)} \mathcal{P}_{\mathrm{NLL}}\left(\left(D\mathbf{A}\right)_i, \mathbf{X}_i\right). \tag{2.7}$$

where $N = CHW$ is the number of entries in $\mathbf{X}$. Note that by using the PNLL instead of the MSE, small magnitude counts can contribute just as much error as the large magnitude

counts. The MSE would prioritize minimizing the error of the high counts, but the PNLL prioritizes the error equally regardless of the count magnitude.

2.3.1.2. Spatial Denoising. We know, however, that neighboring pixels are not uncorrelated. In fact, they are often highly correlated with each other. To promote spatial smoothness, we include an often-used regularizer in the ℓ_2 norm of neighboring pixel differences. This is equivalent to the squared error, which penalizes large spatial differences. We choose to use the 4-neighborhood of a pixel; the total, vertical, and horizontal regularizers are

$$\mathcal{L}_{\text{reg}}(\mathbf{A}) = \mathcal{L}^{H}_{\text{reg}}(\mathbf{A}) + \mathcal{L}^{W}_{\text{reg}}(\mathbf{A}) \tag{2.8}$$

$$\mathcal{L}^{H}_{\text{reg}}(\mathbf{A}) = \sum_{m=1}^{M}\sum_{h=1}^{H-1}\sum_{w=1}^{W}\left(\mathbf{A}_{m,h+1,w} - \mathbf{A}_{m,h,w}\right)^2 \tag{2.9}$$

$$\mathcal{L}^{W}_{\text{reg}}(\mathbf{A}) = \sum_{m=1}^{M}\sum_{h=1}^{H}\sum_{w=1}^{W-1}\left(\mathbf{A}_{m,h,w+1} - \mathbf{A}_{m,h,w}\right)^2 \tag{2.10}$$

respectively. The abundance matrix is used instead of $D\mathbf{A}$ since we want similar mixture amounts of the dictionary endmembers.

Our spatial regularizer in Eq. (2.8) smooths identically without regard to edges in the XRF volume. This can pose a problem: for example, local areas in the painting similar in color likely have similar spectra. These areas should be penalized for spatial differences. On the other hand, local areas of different colors likely have different spectra, and should not be penalized as heavily as the smooth areas. We seek to incorporate the RGB image, $\mathbf{I}$ since it provides valuable and rudimentary insight into the spatial structure of the XRF volume (at least on a surface level).

Weighting mechanisms can be introduced into Eq. (2.8) to account for the color edges:

$$\mathcal{L}_{\text{reg}}\left(\mathbf{A};\mathbf{I}\right) = \mathcal{L}_{\text{reg}}^{H}\left(\mathbf{A};\Omega^{H}\right) + \mathcal{L}_{\text{reg}}^{W}\left(\mathbf{A};\Omega^{W}\right) \tag{2.11}$$

$$\mathcal{L}_{\text{reg}}^{H}\left(\mathbf{A}\right) = \sum_{m=1}^{M}\sum_{h=1}^{H-1}\sum_{w=1}^{W}\Omega_{h,w}^{H}\left(\mathbf{A}_{m,h+1,w} - \mathbf{A}_{m,h,w}\right)^{2} \tag{2.12}$$

$$\mathcal{L}_{\text{reg}}^{W}\left(\mathbf{A}\right) = \sum_{m=1}^{M}\sum_{h=1}^{H}\sum_{w=1}^{W-1}\Omega_{h,w}^{W}\left(\mathbf{A}_{m,h,w+1} - \mathbf{A}_{m,h,w}\right)^{2} \tag{2.13}$$

where $\Omega^{H} \in \mathbb{R}^{(H-1)\times W}$ and $\Omega^{W} \in \mathbb{R}^{H\times(W-1)}$ are the weights defined as

$$\Omega_{h,w}^{H} = \exp\left(-\beta\sum_{c=1}^{3}\left(\mathbf{I}_{c,h+1,w} - \mathbf{I}_{c,h,w}\right)^{2}\right), \tag{2.14}$$

$$\Omega_{h,w}^{W} = \exp\left(-\beta\sum_{c=1}^{3}\left(\mathbf{I}_{h,w+1,c} - \mathbf{I}_{h,w,c}\right)^{2}\right), \tag{2.15}$$

which are adapted from Dai *et al.* [44]. Hyperparameter $\beta \geq 0$ controls the reliance on the color edges; at $\beta = 0$, the adaptive regularizer of Eq. (2.11) simplifies to the standard regularizer of Eq. (2.8). Note that these adaptive weights Ω^{H} and Ω^{W} are large when the RGB gradient is small and vice-versa. We postulate that similar weighting schemes can be developed for other modalities outside of color imaging, but the RGB image can be quickly captured.

2.3.1.3. Sparsity. Lastly, we assume that not every spectral response is expected to be present in every pixel. For instance, assume we have a painting with one or a couple paint layers. In blue regions, which may be due to a copper-based pigment, we should not expect mercury, which is often present in red pigments in old paintings. Mathematically,

we add the ℓ_0 pseudonorm of the abundance volume in order to promote spectral sparsity:

$$\mathcal{L}_{\ell_0}(\mathbf{A}) = \|\mathbf{A}\|_0, \tag{2.16}$$

which counts the number of nonzero entries in $\mathbf{A}$. The ℓ_0 pseudonorm is difficult to optimize since it is non-differentiable, but there are means to approximate it that will be discussed.

2.3.1.4. Optimization. We now define the full optimization problem as a weighted sum of the losses defined in Eqs. (2.7), (2.11), and (2.16):

$$D^*, \mathbf{A}^* = \underset{D,\mathbf{A}\geq 0}{\arg\min}\ \mathcal{L}_{\text{NLL}}(D\mathbf{A}, \mathbf{X}) + \alpha_{\text{reg}} \cdot \mathcal{L}_{\text{reg}}(\mathbf{A}; \mathbf{I}) + \alpha_{\ell_0} \cdot \mathcal{L}_{\ell_0}(\mathbf{A}). \tag{2.17}$$

Hyperparameters α_{reg} and α_{ℓ_0} control the weighting for their respective losses.

Once we have the optimized dictionary and abundance, we can then find the optimized XRF count volume by

$$\mathbf{X}^* = D^*\mathbf{A}^* \tag{2.18}$$

and subsequently the count rate by

$$\mathbf{\Lambda}^* = \frac{\mathbf{X}^*}{t}. \tag{2.19}$$

Further XRF processing and analysis can be done using these volumes, such as extracting the elemental distribution maps.

2.3.2. Solution

The $\mathcal{L}_{\text{NLL}}$ term in Eq. (2.17) is complex, and affects the optimization methods we can use. Ideally, we would use methods like ADDM [48], but the separable losses in D and $\mathbf{A}$ have no analytical solution. Thus, we propose to use gradient descent-based optimization methods to minimize the loss function.

As was mentioned, the ℓ_0 norm is not differentiable, so we require a differentiable substitute. Instead, we use the Elastic Net loss [49] to replace the l_0 norm. Elastic Net penalizes elements of $\mathbf{A}$ by a convex combination of the ℓ_1 and ℓ_2 norms:

$$\mathcal{L}_{\text{elas}}(\mathbf{A}) = \rho\|\mathbf{A}\|_2^2 + (1-\rho)\|\mathbf{A}\|_1 \tag{2.20}$$

for some $\rho \in [0,1]$. The optimization equation used for implementation is updated from Eq. (2.17) as

$$D^*, \mathbf{A}^* = \underset{D,\mathbf{A}\geq 0}{\arg\min}\ \mathcal{L}_{\text{NLL}}(D\mathbf{A}, \mathbf{X}) + \alpha_{\text{reg}} \cdot \mathcal{L}_{\text{reg}}(\mathbf{A}; \mathbf{I}) + \alpha_{\text{elas}} \cdot \mathcal{L}_{\text{elas}}(\mathbf{A}) \tag{2.21}$$

by replacing the ℓ_0 loss with the Elastic Net loss.

This loss penalizes nonzero entries, but in almost all cases, it can never set entries to exactly zero. Zero-valued entries are possible using the Least Absolute Shrinkage and Selection Operator (LASSO) [50] method. LASSO is an optimization technique that sets values in $\mathbf{A}$ below a certain threshold to zero, then removes those entries from future optimization updates to maintain the zero value. This, in conjunction with the Elastic Net loss, produce the intended effect of the ℓ_0 norm in an optimization-friendly manner.

Good initializations of D and $\mathbf{A}$ are also important in the convergence of the optimization. We use K-means clustering of the pixels to initialize the dictionary. This method ensures that all cluster centers are nonnegative given that the dataset is nonnegative. The abundance volume is the found via a nonnegative least squares fit of the K-means dictionary and the XRF data.

Adam [51] optimizes D and $\mathbf{A}$ until convergence, although we employ an alternating optimization framework. The abundance matrix $\mathbf{A}$ is updated by itself, then the dictionary D in another iteration. This is to accommodate for LASSO since we do not want the dictionary to update based on abundances that may be reduced to zero.

2.4. Experiments

We test our algorithm on Jan Davidsz. de Heem's *Bloemen en Insecten* as shown in Fig. 2.1. It was scanned by de Keyser *et al.* [41] and consists of 2048 photon energy channels and has a resolution of 578×673 after registering the RGB image to the target XRF volume. In our experiments, we treat this volume $\mathbf{Y} \in \mathbb{R}^{578\times673\times2048}$ as the ground truth photon count. The ground truth count rate $\mathbf{\Psi}$ is found by

$$\mathbf{\Psi} = \frac{\mathbf{Y}}{t_{\mathbf{Y}}} \tag{2.22}$$

where the reported dwell time per pixel is $t_{\mathbf{Y}} = 285$ ms/px. Scanning an area of 578×673 pixels with this dwell time would require over 30 hours of scanning.

2.4.1. Procedure

Using $\mathbf{\Psi}$, we simulate raster scans at various dwell times from a 5-fold speedup (57 ms/px, about a 6 hr scan) to a 100-fold speedup (2.85 ms/px, about a 19 min scan). To test our algorithm, we identified 37 elements likely to compose a painting, leading to our choice of $M = 37$ dictionary atoms. Additionally, we set $\lambda_{\text{TV}} = 10^{-2}$ and $\lambda_{EN} = 10^{-4}$. Hyperparameters $\beta = 16$ of Eqs. (2.14) and (2.15), and $\rho = 0.2$ of Eq. (2.20). We compare against three other methods: (1) PURE-LET2 with cycle-spinning (5 cyclic shifts) and 5 Haar wavelet scales [46, 52], (2) our implementation of MCR-ALS [27] also with 37 dictionary endmembers, and (3) the original simulated data without optimization.

2.4.2. Evaluation Methods

There are two metrics we use for comparison: the RMSE and mean Poisson Kullback-Leibler divergence (PKLD). Whereas the RMSE is a measure of the absolute error, the PKLD measures the statistical distance between two Poisson variables. It is defined as

$$\mathcal{L}_{\text{KLD}}\left(\mathbf{\Lambda}, \mathbf{\Psi}\right) = \frac{1}{N} \sum_{i \in (c,h,w)} \mathbf{\Lambda}_i - \mathbf{\Psi}_i \left(\ln \mathbf{\Lambda}_i - \ln \mathbf{\Psi}_i + 1\right). \tag{2.23}$$

This is identical to the PNLL loss shifted such that the minimum value is zero:

$$\mathcal{L}_{\text{KLD}}\left(\mathbf{\Lambda}, \mathbf{\Psi}\right) = \mathcal{L}_{\text{NLL}}\left(\mathbf{\Lambda}, \mathbf{\Psi}\right) - \mathcal{L}_{\text{NLL}}\left(\mathbf{\Psi}, \mathbf{\Psi}\right). \tag{2.24}$$

The same properties that the PNLL enjoys, the PKLD also enjoys. Including the PKLD as a metric provides a measure of the relative error, which the RMSE cannot capture. We

compare the denoised volume rates against the ground truth rates since the respective dwell times are different.

In addition to evaluating the performance on the XRF volume, we also compare the performance of the elemental distribution maps. These maps are generated from the XRF volumes by factoring $\mathbf{X}^*$ with a dictionary of known XRF responses for each chemical element. PyMca [15], a platform for XRF analysis, is used to generate the elemental maps for all the XRF volumes. The RMSE and PKLD are the metrics for comparing the elemental distribution maps as well.

2.4.3. Results

Fig. 2.2 shows some elemental maps of varying count rate distributions and their corresponding RMSEs and PKLDs. The dwell time for those elemental maps is 20x faster than $t_{\mathbf{Y}}$, which is approximately 14.25 ms/px (≈ 92 min scan time).

Numerically, PURE-LET is the main competitor of our algorithm. PURE-LET performed best for As K, Ca K, and the PKLD of Cl K. MCR-ALS only performed best in PKLD for Si K and erratically in the other maps. Some maps, such as Ca K and Cu K, were the second best performing for MCR-ALS, but other maps were wildly off from the other algorithms (see As K and Pb L3). Our algorithm performed best for Co K, Cu K, Pb L3, and the RMSEs of Cl K Si K, which is a slight majority of the comparisons.

Interestingly, there is one element in each denoising method where the results were worse than the raw data. Our algorithm has higher error for Ca K, while PURE-LET and MCR-ALS were outperformed by the raw data in Cu K. This suggests that there may not be one set of XRF volume denoising parameters that can denoise all the maps equitably.

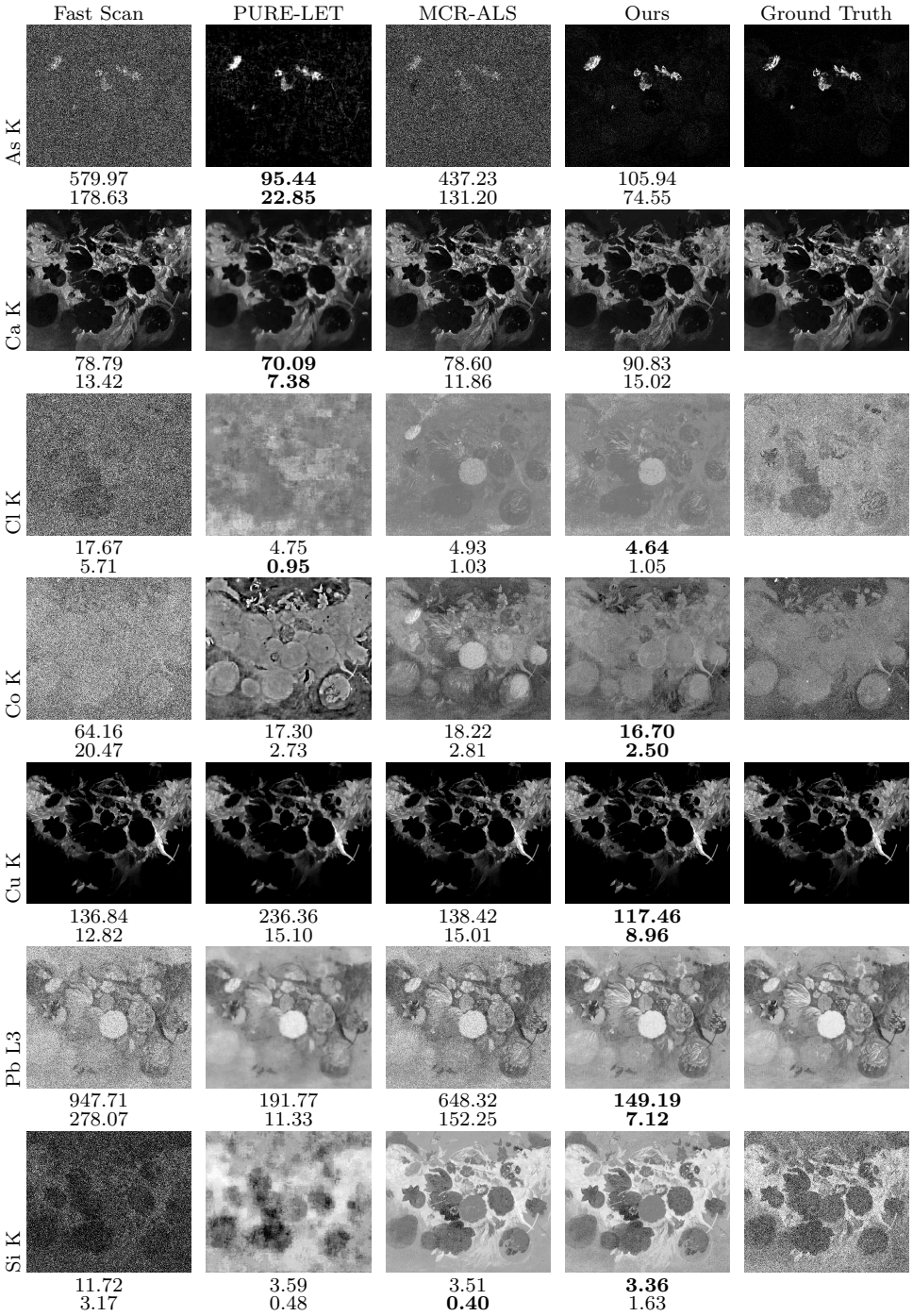


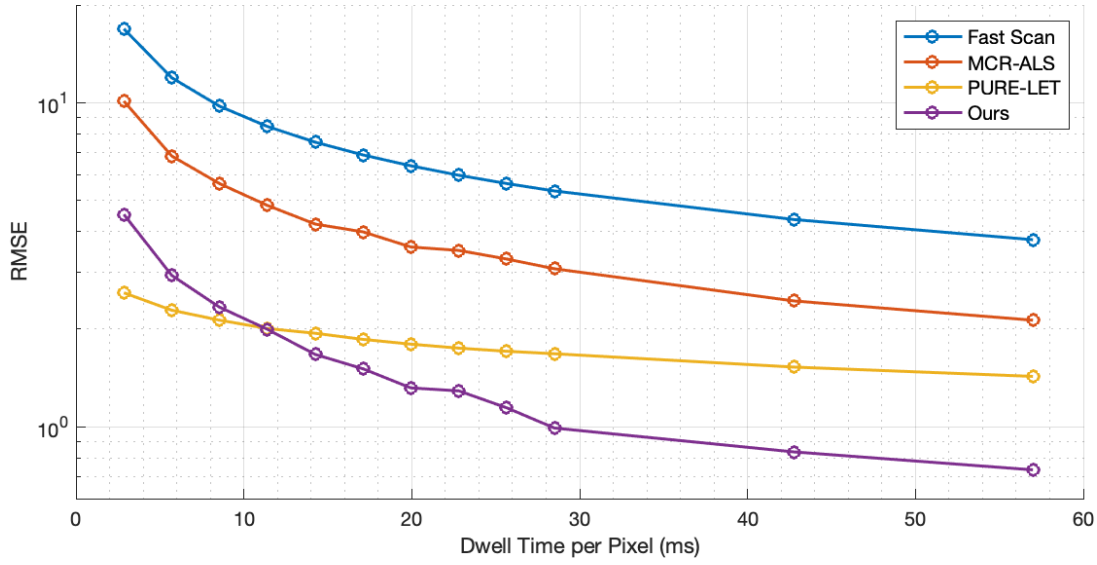
Figure 2.2. Visual comparison of seven elemental maps. Numbers below are the RMSE (top) and PKLD (bottom).

Qualitatively, PURE-LET tends to show different structures than the ground truth despite all the maps being shown in the same display range per element. Elements Cl K and Si K show similar noise patterns that may be beneficial quantitatively, but is difficult to deduce visually. Just as MCR-ALS was numerically erratic, so too is the visual quality. For example, As K and Pb L3 appear minimally denoised, but MCR-ALS extracted structural information from the noisiest of elements Cl K, Co K, and Si K. However, false high counts are introduced in Cl K and Co K in some areas such as the bright flower in the top left that is not apparent in the ground truth. Introducing false high or low counts can be highly detrimental to XRF analysis, especially if it is used to inform painting treatments for example. Our algorithm reliably denoises each map without introducing egregious artifacts.

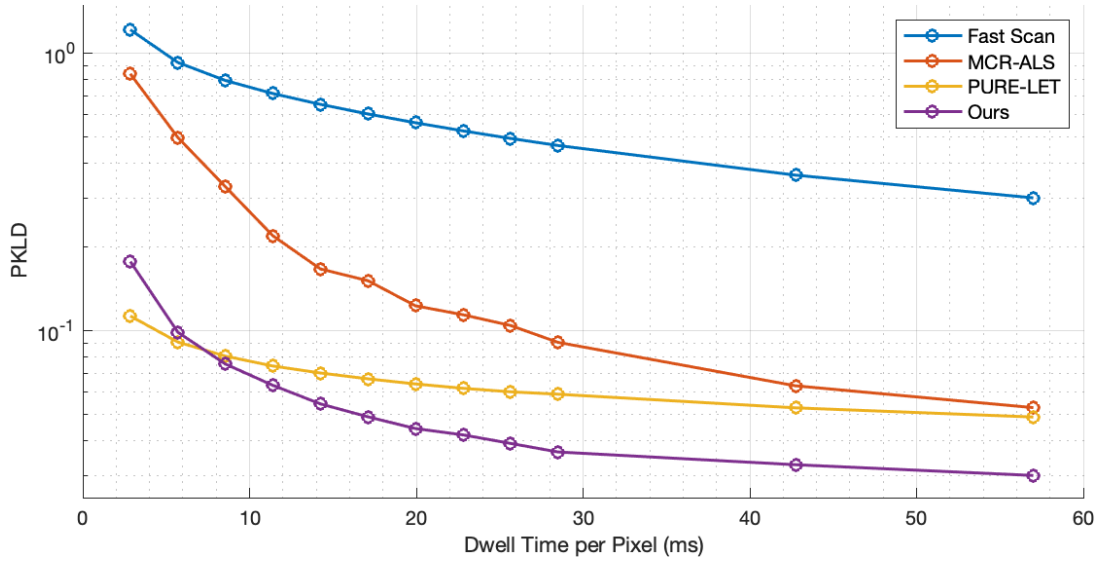
Both quantitatively and qualitatively, we believe our algorithm performs best for denoising the elemental maps.

A volumetric comparison of the algorithms are shown in the plots of Fig. 2.3. These plots show the error in the denoised volumes for different sampling times. As expected, the error decreases with increasing scan time regardless of the metric or algorithm. The differences lie in the magnitude of the error and the rate of error decrease.

Our algorithm starts off with a higher RMSE and PKLD than PURE-LET, but overtakes it at about 11.5 ms/px for the RMSE and 7.5 ms/px for the PKLD. Depending on the instrument, these low dwell times, while fine for simulations, may approach or even exceed the recommended scanning speed. The ability of the gantry to localize where it is in real space becomes increasingly difficult at higher scan speeds. We recommend this be tested on one's system before conducting fast scan experiments. Considering the volume



(a) RMSE errors of the optimized XRF volumes.



(b) PKLD errors of the optimized XRF volumes.

Figure 2.3. Error plots of the XRF volumes at different dwell times.

error plots and elemental distribution map results, we believe our algorithm should be the algorithm of choice.

2.5. Conclusion

We introduced a new method for denoising XRF volumes that combines a Poisson noise model with sparse dictionary learning. A regularizer that uses gradient information from a color image spatially smooths the data. The ℓ_0 norm approximation offers further spectral denoising beyond the dictionary model. Our algorithm outperforms methods designed for XRF denoising and Poisson denoising in general both in quantitative and qualitative terms.

Despite these speedups, our algorithm can still recover high-quality XRF volumes and elemental maps. Scans that require 20x less time can not only ease time-related issues for scanning works of art, but could also open the opportunity for researchers to scan more paintings in a session. We hope that this opens the door for more XRF scans of historical paintings.

CHAPTER 3

Denoising the Elemental Maps of Fast X-Ray Fluorescence Raster Scans of Paintings

Abstract

Macro x-ray fluorescence (XRF) imaging of cultural heritage objects, while a popular non-invasive technique for providing elemental distribution maps, is a slow to acquire high signal-to-noise ratio XRF volumes. In an effort to reduce the scan times without sacrificing elemental map quality, we propose a Poisson-based optimization to remove noise incurred from these faster scans. A Poisson noise model and regularizer restores the data quickly; an optional color image-based prior improves upon the denoising algorithm. Simulated experiments demonstrate that scan times can be reduced by an order of magnitude with minimal error.

3.1. Introduction

In the scientific studies of paintings, macro x-ray fluorescence (XRF) imaging is frequently used as a non-invasive tool to analyze works of art. This approach leverages the insights gained from XRF point analysis by scanning over an area. The data collected from these XRF measurements can be used to determine the presence and location of chemical elements. These elemental distribution maps provide information not just on the surface layer of the artwork, but throughout all the paint layers. With these maps,

Figure 3.1. Bernardo Biti, *Raising of the Cross*, $31 \times 23\,\mathrm{cm}^2$, The Thoma Foundation, inv. no. 2017.72, oil on copper.

for example, an art conservator can better preserve paintings [7], or an art historian can deduce an artist's painting techniques—sometimes revealing hidden paintings [4].

While collecting XRF data has evolved from sampling single points to sampling a two dimensional array, spectroscopists still use a point scanner mounted on a gantry to perform XRF imaging since it is cheap and effective. As the gantry moves the scanner in a whisk broom motion, the detector captures the spectral response emitted from the atoms bombarded by x-rays at each point. These photons are binned based on their energy, which then compose a single pixel. Without an array of sensors like a conventional camera, the

time to acquire the XRF data increases drastically having to go point by point instead of in a single capture akin to a conventional camera.

Depending on the scanning area, spot size, and sampling time per pixel (typically on the order of low tenths of a second), it often takes many hours to acquire the XRF volume. Take as example a modest painting of size $600 \times 720\,\mathrm{mm}^2$. If we specify a scan with spot size $1\,\mathrm{mm}^2$ and dwell time $0.2\,\mathrm{s/px}$, it would take exactly 1 day to scan. Normally, this time scale would not be a problem; however, the cultural significance of these samples cannot be discounted. Access to historical paintings is under the discretion of their owners, who may only allow data collection when, for example, the painting is not on display, under supervision, etc. Of note, Alfeld *et al.* had to leave the bottom left corner of a painting unscanned due to these time constraints [**53**]. Spectroscopists may also want to scan multiple paintings in a collection, which further increases the time needed to complete all the scans. Additionally, time is not the only issue for the spectroscopists themselves. Cultural heritage objects cannot be replaced, so the x-ray exposure time should be minimized to best preserve the painting. These considerations introduce an engineering trade-off between the time spent scanning the artwork and the quality of the signal: the longer the scan, the higher the signal-to-noise ratio.

Analysis of XRF volumes uses the recorded photon counts. These photon count volumes are then separated into elemental maps using a nonnegative least squares fit with a matrix composed of known elemental XRF responses [**14**]. The quality of the volume and elemental maps depends heavily on the amount of time each pixel was sampled. Before collecting XRF data, the spectroscopist must determine the trade-off between image quality (e.g. mean squared error) and acquisition time: the longer the sampling time, the

less noisy the sample. Without postprocessing algorithms, increasing the sampling time is the only way to improve the quality (aside from hardware upgrades). Our goal here is to provide such an algorithm so that spectroscopists are able to quickly and confidently acquire high quality data in a fraction of the time. We test our technique on simulated elemental maps at different dwell times based on XRF data of Bernardo Biti's *Raising of the Cross* as shown in Fig. 3.1.

3.2. Related Work

Dictionary learning approaches frequently appear in XRF literature since each element emits a characteristic set of discrete fluorescent lines. Limiting the number of spectral representations makes intuitive sense, as each pixel is then composed of a linear combination of different spectra. We detailed how many XRF analysis techniques can themselves be considered spectral denoising techniques [54]. Of note, Alfeld *et al.* proposed using nonnegative matrix factorization to better analyze XRF correlations than principal component analysis [21]. Martins *et al.* proposed denoising XRF volumes using multivariate curve resolution-alternating least squares (MCR-ALS), a dictionary learning approach in the spectral domain to separate different elemental compositions [27, 42]. Kogou *et al.* used an unsupervised learning method called self-organizing maps that also extracts a set of spectral dictionary atoms to decompose the XRF volumes into a representative basis [43]. This method effectively uses k-means clustering to generate the set of dictionary endmembers. Vermeulen *et al.* [55] combined sparse coding and dictionary learning to identify pigments or pigment mixtures.

There are some concerns with these algorithms. Primarily, even though photons arrive according to a Poisson process [45], each of these methods (implicitly) uses a Gaussian noise model since the dwell times are assumed to be long enough. The central limit theorem allows this noise model to be used as an approximation with long sampling times. However, in low XRF count regimes the Gaussian noise model is no longer an accurate approximation. Poisson noise instead becomes the dominant noise model.

Even at long dwell times though, oftentimes these algorithms do not account for the unidentical variance at each pixel. The variance of a Poisson process is equal to its mean, not constant. Mean-squared error (MSE) losses for example would prioritize minimizing the highest count rates instead of equally regardless of the count rate. There are means to still use the MSE, such as the Anscombe transformation and other similar transforms [56, 57] that approximately convert Poissonian data to Gaussian. Applying an inverse-variance weighting mechanism can be applied [58] so that all the data are equally weighted. For XRF data, Li *et al.* [59] propose using a log-log-square root operator as a preprocessing step. This heavily compresses the data to approximately the same order of magnitude while maintaining the shape of the data (*e.g.* peaks are preserved, etc.). The inverse log-log-square root is done as a post-processing step. These are approximations, however, so we suggest fitting the noise model to the data instead of the data to the noise model whenever possible.

Spectral denoising is only considered in these algorithms, but spatial correlations can also be leveraged to improve results [60, 61]. There are a plethora of spatial denoising algorithms, but only a portion are devoted to Poisson noise. Recent advancements include Poissonian Fractional Order Total Variation (FOTV) from Chowdhury *et al.* [62] is an

algorithm that uses a fractional derivative of the total variation (TV) of the image instead of the standard total variation regularizer. They show that this mitigates the quantization effects that often arise in TV minimization problems. This method was originally published using tests on conventional images, but to the best of our knowledge has not been applied to XRF data.

Neural networks have grown in popularity in the past decade. One of the more recent publicly available Poisson denoising algorithms is Noise2Void originally proposed by Krull *et al.* [63]. They show that it is possible to train a small neural network using the noisy image itself to remove the noise. This class of neural networks is useful for XRF data since there is no large, publicly available dataset to train a general XRF denoiser.

Many times, the goal of XRF sampling of paintings is generating high quality elemental maps. To this end, we propose to denoise the elemental maps directly instead of the XRF volume. This allows for tuning the denoising parameters to suit each maps need, which can vary by map as will be seen. Chapter 2 showed as well that the volume denoising algorithms can sometimes fail with regard to the elemental distributions; the three algorithms tested all had an elemental map of worse error than the raw data.

3.3. Denoising Formulation

In this section, we'll mathematically derive the elemental map denoising optimization. Assume that we have the XRF count volume $\mathbf{X} \in \mathbb{N}^{C \times H \times W}$ from a fast raster scan of sampling rate t seconds per pixel. Typically t is on the order of hundreds of milliseconds, but we will examine the cases of shorter times. There are C channels (*i.e.* energy bins), spatial height H, and spatial width W composing the volume of natural numbers $\mathbb{N}$.

We also assume that the XRF volume is drawn from an unknown, underlying Poisson distribution:

$$\mathbf{X} \sim \text{Poiss}\left(\mathbf{Y}\right) \tag{3.1}$$

$$\mathbf{Y} = t \cdot \mathbf{\Psi} \tag{3.2}$$

where $\mathbf{\Psi} \in \mathbb{R}^{C \times H \times W}$ is the ground truth photon arrival rate in counts per second, and $\mathbf{Y} \in \mathbb{R}^{C \times H \times W}$ is the expected number of photons given time t. While $\mathbf{\Psi}$ is unobtainable, it can be estimated provided that that dwell times are sufficiently long.

In practice, XRF analysis is performed on the counts $\mathbf{X}$ instead of the count rates since the time scale is just a multiplicative factor (assuming identical scan times per pixel). However, in this chapter, we will use XRF count rates so that we can fairly assess each algorithm across time scales. We will denote the measured XRF count rates as

$$\mathbf{\Lambda} = \frac{\mathbf{X}}{t}, \tag{3.3}$$

which will be used in place of $\mathbf{X}$, and should approximate $\mathbf{\Psi}$.

From $\mathbf{\Lambda}$, the M distinct elemental maps are then derived from $\mathbf{\Lambda}$ via the minimization of a nonnegative matrix factorization of $\mathbf{\Lambda}$:

$$\mathbf{A}^{\text{el}} = \underset{\mathbf{A} \geq 0}{\arg\min} \frac{1}{N_{\mathbf{A}}} \sum_{i \in (m,h,w)} \left(\mathbf{\Lambda}_i - \left(D^{\text{el}}\mathbf{A}\right)_i\right)^2 \tag{3.4}$$

where $D^{\text{el}} \in \mathbb{R}_+^{C \times M}$ is the dictionary composed of M spectral responses, $\mathbf{A}^{\text{el}} \in \mathbb{R}_+^{M \times H \times W}$ is the abundance matrix showing the location and relative quantities of each spectral response (*i.e.* the elemental distribution maps), and $N_{\mathbf{A}} = \frac{1}{MHW}$ is the number of entries

in $\mathbf{A}^{\text{el}}$. The matrix-tensor multiplication is carried out via

$$\left(D^{\text{el}}\mathbf{A}\right)_{c,h,w} = \sum_{m=1}^{M} D^{\text{el}}_{c,m} \cdot \mathbf{A}_{m,h,w}, \tag{3.5}$$

which is akin to matrix-matrix multiplication.

PyMca [15] is a widely used platform for generating the dictionary and elemental maps. Each column of D^{el} is a spectral response for a single atom, which can include any number of emission lines. An additional column in D^{el} is reserved for removing the background XRF signal. The background consists of incoherent x-rays from the source that were reflected back into the XRF detector. These dictionaries are made by associating peaks in the sum spectra to known elemental emission lines. Non-background endmembers of D^{el} are typically Gaussian curves fit to these peaks [15]. Once D^{el} is determined, the elemental maps $\mathbf{A}^{\text{el}}$ can be found via Eq. (3.4). Given elemental maps $\mathbf{A}^{\text{el}}$, our goal is to estimate the actual elemental rates, which we will denote as $\mathbf{B} \in \mathbb{R}_{+}^{M\times H\times W}$. These maps are found using Eq. (3.5) as well.

3.3.1. Poisson Negative Log Likelihood Loss

In order to evaluate the denoising performance, we need to establish an objective/loss function that minimizes the error. In many applications, the root mean squared error (RMSE) is used:

$$\text{RMSE}\left(\mathbf{A}^{\text{el}}, \mathbf{B}\right) = \sqrt{\frac{1}{N_{\mathbf{A}}} \sum_{i\in(m,h,w)} \left(\mathbf{A}^{\text{el}}_i - \mathbf{B}_i\right)^2}. \tag{3.6}$$

In using the RMSE as an error, there is an implicit assumption that the data being compared was drawn from a normal distribution with constant variance across all entries.

Our model assumes the data arrives according to a Poisson model, which has variance equal to the count rate. Accordingly, we need a loss function that takes this into account.

If the MSE is used for minimizing Gaussian noise, the Poisson negative log likelihood (PNLL) is the analogous log likelihood loss function for minimizing Poissonian noise. The minimum loss occurs when $\mathbf{A}^{\text{el}} = \mathbf{B}$, but the minimum value can vary; this is not a concern for optimization however. The PNLL loss for the elemental maps is

$$\mathcal{L}_{\text{NLL}}\left(\mathbf{A}^{\text{el}}, \mathbf{B}\right) = \frac{1}{N_{\mathbf{A}}} \sum_{i \in (m,h,w)} \mathbf{A}_i^{\text{el}} - \mathbf{B}_i \cdot \ln \mathbf{A}_i^{\text{el}}, \tag{3.7}$$

which prevents the rate estimation from deviating too far from the measurements.

The PNLL is akin to a relative error that depends on the magnitude of the counts. For example, if an entry $\mathbf{A}_i^{\text{el}} = 10$ and $\mathbf{B}_i = 15$, this would incur a larger error than if $\mathbf{A}_i^{\text{el}} = 110$ and $\mathbf{B}_i = 115$ despite the same absolute difference. During optimization, this forces small count rates to stay close to their original value, but allows for larger absolute deviations for high count rates. The RMSE would treat these errors equally since it is an absolute error.

3.3.2. Spatial Smoothness Regularizer

The PNLL we have defined operates pixel-wise, meaning there is an implicit assumption that the pixels are independent of one another (*i.e.* the PNLL is separable for each pixel). Intuitively, this is seldom the case—neighboring pixels should be highly correlated with one another. Take Fig. 3.1 for example, which has large areas of the same blue paint. Naturally, we would expect the spectra of these neighboring pixels to be correlated with one another. In local areas where the paint color changes, we expect less of a correlation.

If we first assume that for any pixel, all adjacent pixels are perfectly correlated with it, then we need a regularizer that penalizes the difference in count rate estimations. Oftentimes these regularizers penalize the absolute difference in some form, such as the ℓ_1 or ℓ_2 norm. Here, since we established that our data is Poissonian, we use the corresponding Kullback-Leibler divergence (KLD) to compare distribution $\mathbf{A}_i$ at entry i against spatially neighboring distribution $\mathbf{A}_j$ at entry j:

$$\mathcal{P}_{\mathrm{KLD}}\left(\mathbf{A}_i, \mathbf{A}_j\right) = \mathbf{A}_i - \mathbf{A}_j \cdot \left(\ln \mathbf{A}_i - \ln \mathbf{A}_j + 1\right). \tag{3.8}$$

The KLD is a statistical distance measure that compares any one probability distribution against any other. The Poissonian KLD (PKLD) of Eq. (3.8) happens to be identical to the PNLL if it is shifted such that the minimum value is zero for all rate estimations.

One item to note is that the PKLD not a symmetric function. The asymmetry poses an issue since spatial regularizers compare pairs of neighboring data. Instead of using the PKLD as a spatial regularizer, we use the Poissonian Jeffreys divergence (PJD), which is a sum of the two PKLD argument combinations:

$$\mathcal{P}_{\mathrm{JD}}\left(\mathbf{A}_i, \mathbf{A}_j\right) = \mathcal{P}_{\mathrm{KLD}}\left(\mathbf{A}_i, \mathbf{A}_j\right) + \mathcal{P}_{\mathrm{KLD}}\left(\mathbf{A}_j, \mathbf{A}_i\right) \tag{3.9}$$

$$= \left(\mathbf{A}_i - \mathbf{A}_j\right) \cdot \left(\ln \mathbf{A}_i - \ln \mathbf{A}_j\right). \tag{3.10}$$

Like the PNLL, the Poissonian Jeffreys divergence provides a relative error where the absolute difference is weighted by the difference in magnitude.

Now, when comparing neighboring pixels, the penalty depends on the magnitudes of the pixels being compared. This is important for denoising since there are often different

magnitude scales in elemental maps, but the same relative amounts of noise. Some areas may have counts in the thousands, while other areas may have counts in the tens. If we were to use the ℓ_1 or ℓ_2 norm, the denoising would be prioritized in the high count rate regions despite having similar relative noise levels.

With the PJD defined, we can define losses

$$\mathcal{L}_{\mathrm{JD}}(\mathbf{A}) = \mathcal{L}^{H}_{\mathrm{JD}}(\mathbf{A}) + \mathcal{L}^{W}_{\mathrm{JD}}(\mathbf{A}) \tag{3.11}$$

$$\mathcal{L}^{H}_{\mathrm{JD}}(\mathbf{A}) = \sum_{m=1}^{M}\sum_{h=1}^{H-1}\sum_{w=1}^{W}\mathcal{P}_{\mathrm{JD}}(\mathbf{A}_{m,h,w}, \mathbf{A}_{m,h+1,w}) \tag{3.12}$$

$$\mathcal{L}^{W}_{\mathrm{JD}}(\mathbf{A}) = \sum_{m=1}^{M}\sum_{h=1}^{H}\sum_{w=1}^{W-1}\mathcal{P}_{\mathrm{JD}}(\mathbf{A}_{m,h,w}, \mathbf{A}_{m,h,w+1}) \tag{3.13}$$

which penalize statistical distances from neighboring pixels both vertically in Eq. (3.12) and horizontally in Eq. (3.13). The sum of the two gives the total PJD loss in Eq. (3.11)

3.3.2.1. RGB Image Prior. One drawback with the spatial regularizers we defined is that edges are disregarded. If we are able to fuse other information with the optimization, we can mitigate over-smoothing of the XRF data. If available, we can exploit the color image $\mathbf{I} \in [0,1]^{3\times H\times W}$. The color image contains information from the visible spectrum, which can be used to identify pigments on the surface level of a painting. We can reasonably assume that where there is a transition in the color, there should be a transition in the pigments (and therefore the XRF data). By examining the spatial gradient of $\mathbf{I}$, higher gradients will indicate an edge, and therefore the PJD should be reduced in those areas. The weights we use are adapted from Dai *et al.* [44] and are found via

$$\Omega^H_{h,w} = \exp\left(-\beta \sum_{c=1}^{3} \left(\mathbf{I}_{c,h+1,w} - \mathbf{I}_{c,h,w}\right)^2\right) \tag{3.14}$$

and

$$\Omega^W_{h,w} = \exp\left(-\beta \sum_{c=1}^{3} \left(\mathbf{I}_{c,h,w+1} - \mathbf{I}_{c,h,w}\right)^2\right) \tag{3.15}$$

where Ω^H and Ω^W are the weights for the vertical and horizontal penalties. Hyperparameter $\beta \geq 0$ is used to control the edge sensitivity: the case where $\beta = 0$ is non-adaptive denoising. Note that these weights are small where the RGB gradient is large.

The weighted regularizers are then

$$\mathcal{L}_{\text{JD}}\left(\mathbf{A};\mathbf{I}\right) = \mathcal{L}^H_{\text{JD}}\left(\mathbf{A};\mathbf{I}\right) + \mathcal{L}^W_{\text{JD}}\left(\mathbf{A};\mathbf{I}\right) \tag{3.16}$$

$$\mathcal{L}^H_{\text{JD}}\left(\mathbf{A};\mathbf{I}\right) = \sum_{m=1}^{M}\sum_{h=1}^{H-1}\sum_{w=1}^{W} \Omega^H_{h,w} \cdot \mathcal{P}_{\text{JD}}\left(\mathbf{A}_{m,h,w}, \mathbf{A}_{m,h+1,w}\right) \tag{3.17}$$

$$\mathcal{L}^W_{\text{JD}}\left(\mathbf{A};\mathbf{I}\right) = \sum_{m=1}^{M}\sum_{h=1}^{H}\sum_{w=1}^{W-1} \Omega^W_{h,w} \cdot \mathcal{P}_{\text{JD}}\left(\mathbf{A}_{m,h,w}, \mathbf{A}_{m,h,w+1}\right). \tag{3.18}$$

Now, edges are better preserved in the optimization process.

3.3.3. Denoising Optimization

With the losses defined, we can create the full optimization equation. We have a weighted sum of Eqs. (3.7) and (3.16) to get

$$\mathbf{A}^* = \underset{\mathbf{A}\geq 0}{\arg\min}\ \mathcal{L}_{\text{NLL}}\left(\mathbf{A}, \mathbf{A}^{\text{el}}\right) + \alpha_{\text{JD}} \cdot \mathcal{L}_{\text{JD}}\left(\mathbf{A};\mathbf{I}\right). \tag{3.19}$$

Here, α_{JD} is a weighting parameter. Since we have no access to the ground truth information $\mathbf{B}$, we instead replace it with $\mathbf{A}^{\text{el}}$, which we do have. In order to solve Eq. (3.19), we first initialize $\mathbf{A} = \mathbf{A}^{\text{el}}$. Although this formulation is a candidate for the Alternating Direction Method of Multipliers [48], the update equations are difficult to minimize since there is no analytical solution. Thus, we turn to gradient descent based methods to solve the equation. We found that the Adam optimizer [51] works well. We determine that the algorithm has converged when the percent improvements in the loss function falls below a threshold.

Additionally, the amount of smoothing needed depends on the elemental map. Instead of optimizing the elemental maps with one α_{JD} value, it can be beneficial to optimize the maps individually (or at once with different α_{JD} per elemental map). The algorithm converges quickly on a CPU, so experimentation to find an appropriate α_{JD} is quick.

3.4. Experiments

For our experiments, we focus on Bernardo Biti's *Raising of the Cross* as shown in Fig. 3.1, which was painted using oil-based paints on a copper plate. Before being restored, there were some areas of the painting where the paint was chipped away, exposing the copper plate. Measurements were taken using the gantry designed by Pouyet *et al.* [12] every 0.5 mm for an overall dimension of 423×566 px. Dwell times were 100 ms per pixel. We treat this original XRF volume as the ground truth $\mathbf{Y}$ since it was scanned long enough where the Poisson noise is less apparent.

		2 ms/px				5 ms/px				10 ms/px			
		Raw	FOTV	N2V	Ours	Raw	FOTV	N2V	Ours	Raw	FOTV	N2V	Ours
Au L	RMSE	686.06	429.52	348.68*	**171.42**	490.12	318.53	305.21*	**183.25**	389.69	264.60*	264.70	**181.78**
	PKLD	348.00	201.00	135.52*	**41.14**	211.17	167.02	110.64*	**52.62**	150.19	90.53*	91.17	**42.72**
Ca K	RMSE	462.34	289.39	**153.61**	197.29*	298.40	180.57	**130.47**	150.10*	216.03	127.80	116.17*	**115.16**
	PKLD	267.57	208.90	**30.64**	63.49*	123.93	99.52	**25.28**	43.66*	72.41	46.76	**20.88**	23.79*
Cu K	RMSE	1976.04	**1304.30**	1488.46	1439.11*	1256.51	**961.30**	1572.19	1058.44*	893.58	843.10*	1383.68	**825.47**
	PKLD	306.77	79.87*	83.54	**78.56**	114.10	**44.31**	72.99	47.96*	56.81	**29.95**	59.30	31.68*
Fe K	RMSE	983.01	480.39	**448.20**	448.39*	625.33	**340.18**	391.03	348.81*	443.10	306.07*	364.30	**284.89**
	PKLD	302.67	124.28	**55.15**	67.65*	123.27	43.98	39.22*	**35.19**	60.68	24.44*	33.01	**21.33**
Hg L	RMSE	976.89	548.83	528.42*	**524.64**	691.69	**464.91**	490.53	466.65*	528.54	412.37*	457.43	**402.17**
	PKLD	225.57	89.78	**76.18**	77.80*	124.13	**64.85**	75.84	68.29*	82.99	**54.72**	61.93	56.18*
Hg M	RMSE	810.34	516.90	**374.80**	393.55*	583.41	385.80	353.23*	**335.63**	465.08	335.96	334.12*	**331.26**
	PKLD	285.46	179.59	**32.80**	76.10*	150.22	66.80	23.93*	**20.99**	84.34	26.23*	**19.64**	28.26
Mn K	RMSE	419.79	211.36	**160.59**	160.84*	267.32	135.20	134.22*	**123.87**	191.24	109.10*	118.85	**106.16**
	PKLD	146.73	75.24	**22.85**	27.09*	72.26	32.77	18.80*	**15.47**	39.89	15.65	14.16*	**11.56**
Ni K	RMSE	463.85	231.07	**189.72**	200.02*	303.16	**163.78**	170.02*	170.67	225.11	**146.63**	176.21	148.61*
	PKLD	137.20	47.17	21.24*	**20.31**	60.49	17.65	17.54*	**15.83**	33.05	12.94*	20.54	**12.92**
Pb L	RMSE	5841.67	**2379.70**	2730.38	2393.89*	3719.41	2180.30	1962.26*	**1881.18**	2637.82	1737.00	1708.85*	**1573.47**
	PKLD	311.90	**58.76**	73.87	62.57*	127.36	49.93	40.42*	**39.00**	64.54	31.54	30.29*	**26.30**
Pb M	RMSE	1709.02	929.21	**604.40**	775.63*	1128.15	580.62	574.07*	**559.77**	841.76	508.24*	533.40	**505.05**
	PKLD	640.90	244.55	**82.51**	201.51*	273.09	79.76*	**73.20**	86.09	147.61	**56.81**	62.72	57.37*
Sn L	RMSE	484.55	286.85	**135.18**	146.84*	311.32	187.34	**118.30**	120.84*	224.99	136.12	110.28*	**99.64**
	PKLD	260.53	176.81	**39.53**	55.63*	153.39	102.72	**32.61**	39.41*	94.42	52.01	29.25*	**27.81**
Ti K	RMSE	214.32	117.65	**81.13**	88.68*	134.76	80.11*	81.94	**68.79**	96.55	66.86*	118.77	**56.43**
	PKLD	165.34	72.72	**18.74**	33.78*	78.49	50.11	**15.39**	17.46*	47.73	34.77	15.35*	**12.88**

Table 3.1. Table of errors for each denoising algorithm. Each row denotes the elemental map to denoise and its corresponding errors. Bold values denote the best performing mask, and asterisked values indicate the second best performing mask.

3.4.1. Experimental Procedure

To get the fast scan, we find the ground truth rates via Eq. (3.3), then resample via Eqs. (3.1) and (3.2) varying the dwell time $t \in \{2, 5, 10\}$ ms/px. These represent speedups of 50x, 20x, and 10x respectively, all of which would result in scan times under an hour. The fastest time was chosen based on the gantry proposed by Pouyet *et al.* [12], which

they noted can scan at speeds of 600 mm/s while maintaining localization accuracy. For our experiments, this corresponds to a minimum dwell time of 0.83 ms/px, but we selected 2 ms/px as a minimum to be more comfortably above the maximum speed. We identified 20 elements present in the sum spectrum, 12 of which are of interest. Using PyMca, we generate the maps for each of the XRF volumes sampled at the four different rates, including the ground truth.

We then denoise the 12 elemental maps of interest using each algorithm, FOTV and N2V. As we mentioned before, we optimize the elemental maps individually instead of as a volume to achieve the best performance. This is the case for all methods. The optimization is carried out with knowledge of the ground truth—meaning we find the parameters that minimize the error between the denoised elemental maps and the ground truth elemental maps. This tests how well the algorithm is able to perform, even though in practice the ground truth would not be known.

3.4.2. Results Evaluation

Our evaluation criteria include the RMSE, which indicates how the average photon count rates of the denoised maps differ from the ground truth count rates. This is the intuitive, often-used metric. We also provide results for the PKLD, which indicates the average distance in a probabilistic sense. Recall that the PKLD measures a relative distance between two Poisson variables, while the RMSE measures an absolute distance regardless of the underlying distribution. Thus, we feel it is appropriate to provide a measure of the absolute error, as well as the average relative error.

Fig. 3.2 shows some elemental maps and their corresponding RMSEs and PKLDs for a $t = 10\,\text{ms/px}$ scan. We invite the reader to zoom into the images to see them in greater detail. For the other data, we provide the results in Table 3.1. Note that our method often outperforms the other techniques both in terms of the RMSE and PKLD starting at 5 ms/px. FOTV has mixed performance across all time scans in terms of both the RMSE and PKLD. N2V excels at very low photon counts, but is overtaken by our algorithm for higher dwell times. There appears to be a cutoff time that our algorithm requires where enough raw XRF data is collected to outperform N2V. We would suggest using our algorithm for longer dwell times, but for very short dwell times N2V appears to be the method of choice.

Given the option between the three algorithms, however, we argue that ours would be preferable. While it is easy to determine which algorithm performs best in each time scale, it is more difficult to choose the algorithm when the ground truth is not known. In addition to looking at the best performance, we should also see which method is the most reliable. The placement ranking should be considered to evaluate the overall performance. Our algorithm was outperformed by both FOTV and N2V in only three instances across all dwell times and all elemental maps (72 comparisons total). N2V and FOTV, however, came in last in 24 and 45 instances respectively. FOTV tended to perform worst in the 2 ms/px regime, and N2V as we noted performed worst in the 10 ms/px regime. Our algorithm only had the worst Ni K RMSE and Pb M PKLD for 5 ms/px, and the worst Hg M PKLD. Because our algorithm performed well across all the dwell times and elemental maps, we believe that our algorithm should be the method of choice for denoising elemental maps.

3.5. Conclusion

We introduced a straightforward denoising algorithm that shows that it is possible to drastically speed up the XRF acquisition technique (up to 50x in our experiments) while maintaining integrity in the elemental maps. It is a more robust algorithm that can be used at any Our algorithm uses a Poisson model to effectively remove noise present in these fast scans. Speedups of these high factors can not only ease time related issues for accessing works of art, but could also open the opportunity for researchers to scan more paintings in a single session. This allows more paintings to be analyzed for historical research and more quickly addressed for conservation concerns. Future research will test our algorithm on real fast XRF scans. The potential for sub 1-hour XRF scans of entire paintings is closer to reality.

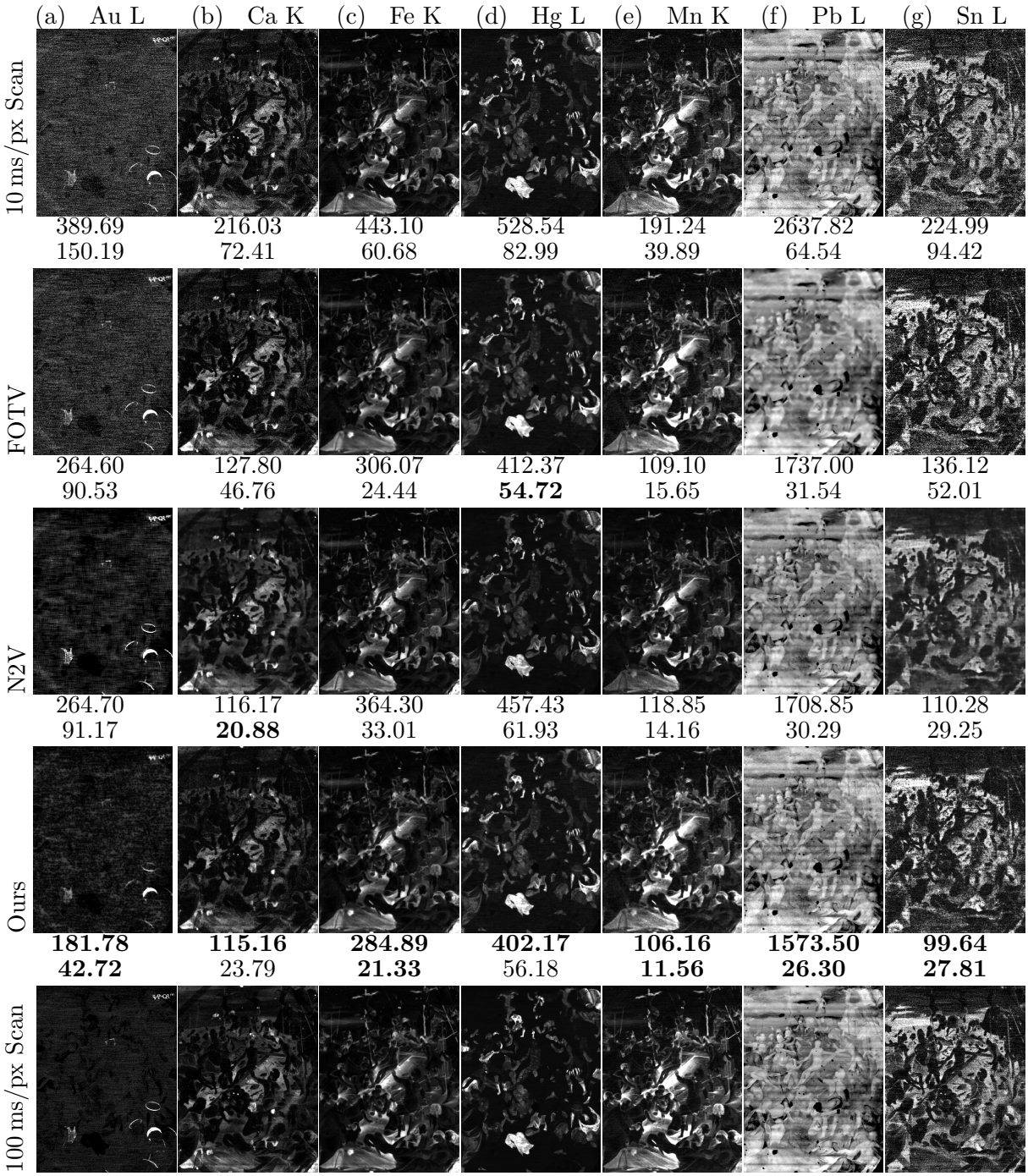


Figure 3.2. Visual comparison of seven elemental maps denoised using different algorithms. Numbers below the maps are the RMSE (top) and PKLD (bottom). Bold numbers indicate the minimum error.

CHAPTER 4

The Design of Supplementary X-Ray Fluorescence Scans of Paintings

Abstract

The time to acquire a macro x-ray fluorescence image of paintings is often a timely process on the order of hours or even days. This is sometimes intractable for spectroscopists and other parties with interest in the painting. Image processing algorithms exist that can address these concerns, but are rarely, if ever, widely implemented by spectroscopists. We propose a novel adaptive sampling technique that is easy to implement, and allows for broad customization depending on the goals of the scan. By quickly and iteratively collecting XRF data, higher quality data can be recorded with or without the use of denoising algorithms.

4.1. Introduction

XRF imaging has become a widely used method for the chemical analysis of historically significant paintings. In an effort to understand the chemical makeup of a painting, XRF provides valuable information of the individual elements that compose the paint pigments. XRF analysts including spectroscopists, art historians, and the like, can then for example inform conservation of the painting [7], or reveal under-paintings [4, 5] to name a few applications.

By illuminating a painting with x-rays, the atoms throughout all the paint layers (and even the canvas itself) will begin to fluoresce. Fluorescing atoms emit photons with a characteristic energy, which occurs when low energy level electrons are dislodged by the x-rays. Electrons occupying higher energy orbitals then transition to occupy the lower energy level left vacant by the dislodged electron. During transition, a photon with energy equal to the energy level difference is emitted to conserve energy. A detector then records the energy of the photons incident upon it. Tables containing the possible photon energies for each element are widely available and are used to identify elements that exist within the scanned object. XRF image data consists of multi-channel pixels that count the number of photons recorded at each energy level.

The popularity of XRF imaging as a cultural heritage tool has promoted research and development in XRF hardware to make it more affordable and more powerful [**8**, **10**, **11**, **12**]. Software development too has been growing for XRF analysis [**21**, **26**, **28**, **15**]. One characteristic that has remained constant, though, is the acquisition process.

Instead of a camera that can capture all the XRF responses at once, a gantry moves the XRF source and detector across the painting, capturing XRF data one pixel at a time. This process can be quite time intensive. Even with hardware improvements, the acquisition time can be quite long. Paintings of modest size often require many hours, if not days, to sample using XRF imaging. In a timescale for chemistry experiments this is not long at all, however these measurements are taken under special circumstances. Primarily, the samples in these experiments are one-of-a-kind and hold cultural significance. Particularly noteworthy paintings are typically housed in museums displaying them to the public, so access to them is limited. *Raising of the Cross* by Bernardo Biti shown in

Figure 4.1. Bernardo Biti, *Raising of the Cross*, $31 \times 23\,\mathrm{cm}^2$, The Thoma Foundation, inv. no. 2017.72, oil on copper.

Fig. 4.1, the painting we will examine in this chapter, is owned by the Thoma Foundation. Alfeld *et al.* for example had to leave the lower left corner of a painting unscanned due to the time limitations imposed by the museum housing the painting [**64**]. Museums may also require human supervision of the experiment to protect the painting from damage as best as possible. The time limitation and human capital required make XRF sampling a challenging task for spectroscopists.

Recently, work has been done to accelerate the XRF acquisition process via image processing as opposed to hardware upgrades. For example, denoising [**30, 31**], super-resolution [**35, 37**], and subsampling XRF algorithms [**36, 40**] have been proposed in

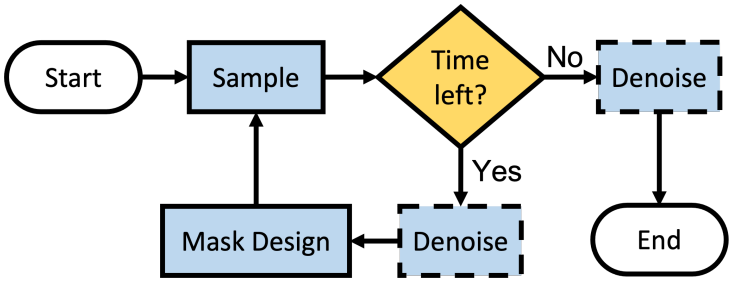


Figure 4.2. Flowchart for XRF sampling algorithm. Denoising is an optional step.

recent years. While these methods may not be explicitly used for speeding up the acquisition process, it is conceivable that they could be: denoising allows for fast, noisy scans, and super-resolution allows for more measurements to be approximated.

Yet, these papers have not been widely adapted in the XRF literature despite showing improvements in their respective goals. One can critique these papers in the following ways: (1) the post-processed data was never explicitly measured (how can the post-processed data be fully trusted?), (2) in cases where the post-processing algorithm fails, the original data may be very noisy, and (3) many experiments have been conducted under simulations, not in the real world. Other criticisms such as accessibility in terms of computational resources and algorithmic understanding steer spectroscopists away from the proposed work.

This chapter aims to alleviate these concerns by introducing a method for time-limited, supplementary scans of paintings that breaks from the traditional raster scan of constant time. We instead propose a smart allocation of the sampling time that effectively minimizes the error in the XRF volume estimations. The algorithm we present here is easy

to implement and quick in providing a sampling strategy, yet powerful for the community's efforts to accelerate XRF acquisition. It can be used in conjunction with denoising algorithms, or as a standalone algorithm working with only the collected data.

Fig. 4.2 shows a flowchart of the proposed algorithm. After an initial fast scan, a sampling mask is designed to strategically collect data that suits the goal of the spectroscopist. The mask is generated with the raw XRF data or the denoised data to mitigate the effects of noise on the mask. Multiple passes can be done within the time limit, or be stopped early if the gains from subsequent scans are not worth it as determined by the spectroscopist. We will look at a two-pass acquisition in this chapter.

Our work first addresses earlier work, then we formulate the sampling algorithms and the optimization process. Additionally, we propose a modification of an existing denoising algorithm to aid in the mask design and (optionally) in the final XRF volume. Our results are simulated from real data, but show the potential of the algorithm in fast XRF imaging.

4.2. Related Work

As was mentioned, there are algorithms that can already be used as a means to accelerate the XRF acquisition process. Subsampling algorithms explicitly address this concern.

Dai *et al.* [36] proposed a subsampling algorithm that uses a neural network to determine the top subset of pixel locations to sample. By feeding a color image of the painting into the network, they find the optimal binary sampling mask; positive pixels are scanned at a "regular" scan time as determined by the spectroscopist and their system. Every other pixel is skipped in an effort to reduce the scan time. These missing

pixels are inpainted via a dictionary-based optimization. The algorithm proposed allows for tight control of the sampling time since the percentage of pixels sampled is controlled (provided the neural network was trained with the same percentage). Sample locations for this method are solely based on the surface-level RGB image—hidden paintings may not be captured well if the underlying structures differ. Given this, along with the other general concerns, a spectroscopist may understandably opt to perform a standard raster scan with their time to ensure the raw data is as interpretable as possible.

Betterton *et al.* [40] proposed an algorithm that would work well in terms of the raw data quality. They too employ a multi-pass approach that uses progressively narrowing apertures. An initial wide-view scan of the painting in its entirety produces blurry results. Their reinforcement learning (RL) algorithm then suggests subsequent scans with more focused lenses in the areas of interest as determined by the neural network. Their simulation shows a well-sampled foreground in less time than that of a standard scan. Again, there are barriers here preventing spectroscopists from adapting this algorithm: (1) the RL algorithm was trained on a small image size (50×50 px) with a relatively small region of interest, and (2) the total scan time is not deterministic since it depends on a hyperparameter weighing a time penalty term. The spectroscopist would need to train a RL network prior to sampling their painting to ensure the correct dimensions are used. Training RL models are notoriously difficult, especially for scientists in other disciplines not well-versed in RL. Further, there is a lack of training data for generalizing this algorithm to a large scale. XRF data is slow to collect, so naturally there are very few samples compared to, say, RGB data. These issues may dissuade spectroscopists from adopting this algorithm in every scan.

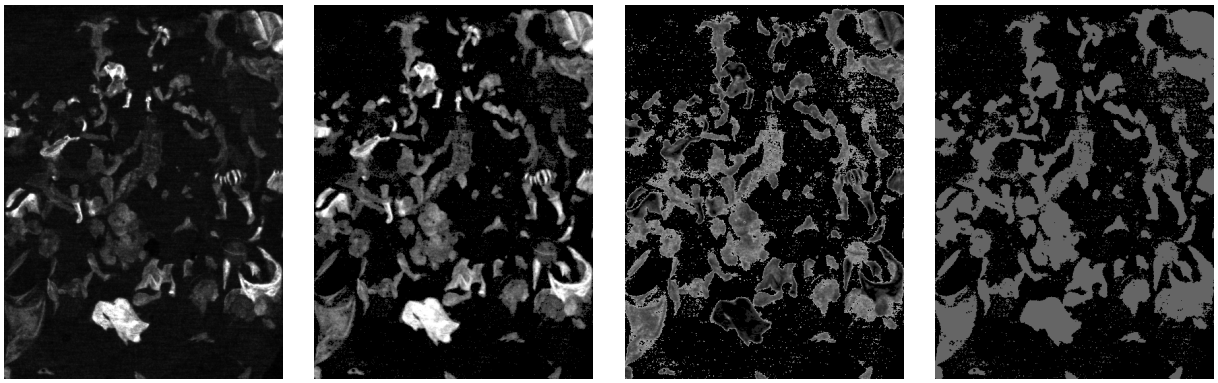
The method proposed by us [30] denoises quickly scanned XRF data using dictionary learning, and Yang *et al.* [31] use classic image processing denoising algorithms on XRF data. The output of these algorithms alter the raw data to denoise the data, which is unappealing to some spectroscopists due to the changing data and the potential for false positive/negative regions of interest. This dictionary denoising method is also computationally intensive, and would be very slow sans a GPU with enough memory for the XRF optimization.

Similarly, Dai *et al.* [35] and Yan *et al.* [37] use dictionary learning methods to super-resolve XRF images, but also suffer from requiring large computational resources, as well as the hallucination of data not sampled. In the field of image processing, inferring unsampled data is a widely acceptable practice, but may not be as acceptable in the cultural heritage and XRF fields.

The purpose of this chapter is, in part, to alleviate the aforementioned concerns. If we can find a method to collect the highest quality *raw* data, there is a chance for it to actually be implemented in XRF systems.

4.3. Formulation

We begin our formulation by mathematically defining the XRF acquisition process. Let $\mathbf{X} \in \mathbb{N}^{C\times H\times W}$ be the XRF data recorded after a scan where $\mathbb{N}$ is the set of natural numbers. The time it took to collect the data at each spatial location is denoted as $T^0 \in \mathbb{R}_+^{H\times W}$ where $\mathbb{R}_+$ is the set of nonnegative real numbers. In a typical scan, T^0 is constant across all entries, but, for completeness, we generalize T^0 such that each entry can be any nonnegative value.



(a) Hg L map (b) MSE Mask (c) PKLD Mask (d) Uniform Mask

Figure 4.3. Denoised Hg L map and the potential masks. The Hg L map was scanned under simulation at 5 ms/px. The masks were designed for a subsequent scan with an average dwell time of 5 ms/px. Each pixel is scanned for at least 1 ms. Count rates of at least 1200 are the set of pixels that can receive more time allocation. The display range is $[0, 11000]$ counts/s for the map, and $[1, 30]$ ms for the masks.

Since photons arrive at the sensor according to a Poisson process, we model $\mathbf{X}$ as being drawn from some unknown, underlying Poisson distribution $\mathbf{Y} \in \mathbb{R}_+^{C\times H\times W}$. The ground truth counts $\mathbf{Y}$ depends on the sampling time; longer dwell times produce more photons. This relation between time and counts is linear with an unknown count *rate* that does not depend on time. We denote this count rate $\mathbf{\Psi} \in \mathbb{R}_+^{C\times H\times W}$. The process for obtaining $\mathbf{X}$ is thus

$$\mathbf{X}_{c,h,w} \sim \text{Poiss}\left(\mathbf{Y}_{c,h,w}\right) \tag{4.1}$$

$$\mathbf{Y}_{c,h,w} = T^0_{h,w} \cdot \mathbf{\Psi}_{c,h,w}. \tag{4.2}$$

Our initial estimation for the count rates is then

$$\mathbf{\Lambda}_{c,h,w} = \frac{\mathbf{X}_{c,h,w}}{T^0_{h,w}} \tag{4.3}$$

where $\mathbf{\Lambda} \in \mathbb{R}_+^{C \times H \times W}$ is the measured count rates.

There will inevitably be some error between the measurements $\mathbf{\Lambda}$ and the underlying count rate distribution $\mathbf{\Psi}$ which we would like to minimize. Our problem statement allows us extra scan time $\tau \in \mathbb{R}_+$ after the initial scan T^0. Further sampling the painting improves upon the initial measurements. How should should we distribute time τ amongst the subsequent scan $T \in \mathbb{R}_+^{H \times W}$ such that we minimize the count rate error? To solve this, we take a probabilistic approach that depends on the chosen error metric. There are two errors we choose: (1) mean-squared error (MSE), and (2) mean Poisson Kullback-Leibler divergence (PKLD). These errors have interesting properties that prioritize sampling of different types of pixels.

4.3.1. Mean Squared Error Metric

Perhaps the most widely used error metric, the mean squared error is an intuitive way to measure the error between data. For our data, we formulate the MSE as

$$\boldsymbol{\mathcal{E}}^{\text{MSE}} = \frac{1}{N} \sum_{i \in (c,h,w)} (\mathbf{\Psi}_i - \mathbf{\Lambda}_i)^2 \tag{4.4}$$

where $N = CHW$ is the number of entries in $\mathbf{\Lambda}$. Since $\boldsymbol{\mathcal{E}}^{\text{MSE}}$ is itself a random variable, we can find the expected error:

$$\mathbb{E}\left[\boldsymbol{\mathcal{E}}^{\text{MSE}}\right] = \frac{1}{N} \sum_{c,h,w} \frac{\mathbf{\Psi}_{c,h,w}}{T^0_{h,w}} \tag{4.5}$$

where $\mathbb{E}[\cdot]$ is the expected value operator. Note that minimizing the expected MSE error is equivalent to minimizing the variance of $\mathbf{\Lambda}$. We readily find Eq. (4.5) with the

understanding that the expected value of $\mathbf{\Lambda}$ is

$$\mathbb{E}\left[\mathbf{\Lambda}_{c,h,w}\right] = \mathbf{\Psi}_{c,h,w}, \tag{4.6}$$

and the variance is

$$\text{var}\left(\mathbf{\Lambda}_{c,h,w}\right) = \frac{\mathbf{\Psi}_{c,h,w}}{T^0_{h,w}}. \tag{4.7}$$

The expected error of Eq. (4.5) suggests that the majority of the error occurs where the underlying rate $\mathbf{\Psi}$ is high and also where the sampling time $T^0_{h,w}$ is low. Of course, we can never be absolutely certain of $\mathbf{\Psi}$, but we can use $\mathbf{\Lambda}$ (or a denoised $\mathbf{\Lambda}$) as an appropriate estimation.

We then formulate a predictive model that estimates the MSE given additional scan time on top of the initial time:

$$\mathbb{E}\left[\boldsymbol{\mathcal{E}}^{\text{MSE}}\left(T\right)\right] \approx \frac{1}{N}\sum_{c,h,w}\frac{\mathbf{\Lambda}_{c,h,w}}{T^0_{h,w}+T_{h,w}}. \tag{4.8}$$

Since there is an initial dwell time, we incorporate the total dwell time to minimize the error of the total XRF volume, not just the subsequent volume. The best allocation of the next scan time τ can then be found by minimizing the expected error of Eq. (4.8):

$$T^* = \underset{\substack{\sum_{h,w} T_{h,w} \leq \tau \\ T^{\min}_{h,w} \leq T_{h,w} \leq T^{\max}_{h,w}}}{\arg\min} \frac{1}{N}\sum_{c,h,w}\frac{\mathbf{\Lambda}_{c,h,w}}{T^0_{h,w}+T_{h,w}}. \tag{4.9}$$

The first constraint ensures that the total scan time of the dwell map does not exceed the total available time. It is clear (both mathematically and intuitively) that without the

total time constraint, the dwell time at each pixel would tend towards infinity to minimize the measurement error; the minimization would then be trivial.

We introduce the second constraint to better control the design of the dwell map by establishing the minimum dwell time $T^{\min} \in \mathbb{R}_+^{H\times W}$ and the maximum dwell time $T^{\max} \in \mathbb{R}_+^{H\times W}$ where $T_{h,w}^{\min} \leq T_{h,w}^{\max}$. Fly scanning gantries for example are capable of recording measurements and moving at the same time. These gantries have a maximum speed limited by motor capabilities or other factors such as gantry localization or even safety concerns for the painting. A maximum speed corresponds to a minimum dwell time that the gantry must measure as the gantry translates across the painting. Maximum dwell times can also be used to ignore certain regions of the painting by setting the maximum to zero. Oversampling a pixel may occur without setting a maximum dwell time too. Spectroscopists often scan paintings in the low hundreds of milliseconds because those times provide acceptable signal-to-noise ratios. Capping dwell times forces the scan time to be redistributed to other less sampled pixels.

Fig. 4.3a shows a sample image of the Hg L line that was generated from simulated XRF data acquired at 5 ms/px. Using the Hg L data for the mask optimization, we show a corresponding MSE mask in Fig. 4.3b where the average dwell time in the next scan is also 5 ms/px. To ensure that each pixel has some additional scan time, we set the minimum dwell time per pixel at 1 ms. Note that the MSE mask appears roughly as a gamma adjusted Hg L map, which will be explained in the solution of the optimization.

4.3.2. Poisson Kullback-Leibler Divergence

In the previous subsection, we formulated a way to minimize the MSE error. The implicit assumption with using the MSE is that the data is drawn from a Gaussian of unit variance. This is not the case with XRF data since it follows a Poisson distribution. Instead, an appropriate way to compare two Poisson rates is via the Kullback-Leibler divergence, which measures the statistical distance between two random variables. The average Poisson Kullback-Leibler divergence can be derived as

$$\boldsymbol{\mathcal{E}}^{\mathrm{PKLD}} = \frac{1}{N} \sum_{i \in (c,h,w)} \boldsymbol{\Lambda}_i - \boldsymbol{\Psi}_i \cdot (\ln \boldsymbol{\Lambda}_i - \ln \boldsymbol{\Psi}_i + 1) \,. \tag{4.10}$$

Similar to the derivation of the MSE mask, we find the expected PKLD error:

$$\mathbb{E}\left[\boldsymbol{\mathcal{E}}^{\mathrm{PKLD}}\right] = \frac{1}{N} \sum_{i \in (c,h,w)} \boldsymbol{\Psi}_i \left(\ln \boldsymbol{\Psi}_i - \mathbb{E}\left[\ln \boldsymbol{\Lambda}_i\right]\right) . \tag{4.11}$$

The expectation of a logarithm of a random variable is difficult to calculate, but can be approximated using a Taylor series expansion of $\ln \boldsymbol{\Lambda}_i$ about $\mathbb{E}\left[\boldsymbol{\Lambda}_i\right]$:

$$\begin{aligned} \mathbb{E}\left[\ln \boldsymbol{\Lambda}_i\right] &= \mathbb{E}\left[\sum_{k=0}^{\infty} \frac{\ln^{(k)} \mathbb{E}\left[\boldsymbol{\Lambda}_i\right]}{k!} \left(\boldsymbol{\Lambda}_i - \mathbb{E}\left[\boldsymbol{\Lambda}_i\right]\right)^k\right] \\ &= \ln \boldsymbol{\Psi}_i + \sum_{k=2}^{\infty} \frac{(-1)^{k+1}}{k \cdot \boldsymbol{\Psi}_i^k} \mathbb{E}\left[\left(\boldsymbol{\Lambda}_i - \boldsymbol{\Psi}_i\right)^k\right] \end{aligned} \tag{4.12}$$

where $\ln^{(k)}(\cdot)$ is the k^{th} derivative of the natural logarithm. If we approximate the expected value with a maximum order of $\left(\boldsymbol{\Psi}_{c,h,w} \cdot T^0_{h,w}\right)^{-1}$, we obtain

$$\mathbb{E}\left[\ln \boldsymbol{\Lambda}_{c,h,w}\right] \approx \ln \boldsymbol{\Psi}_{c,h,w} - \frac{1}{2\, \boldsymbol{\Psi}_{c,h,w} \cdot T^0_{h,w}} . \tag{4.13}$$

There are other terms that are of smaller order, but we ignore them for now. If we substitute Eq. (4.13) into Eq. (4.11), we can approximate the error as

$$\mathbb{E}\left[\boldsymbol{\mathcal{E}}^{\mathrm{PKLD}}\right] \approx \frac{1}{N}\sum_{c,h,w}\frac{1}{2\,T^{0}_{h,w}}. \tag{4.14}$$

Now, just like the MSE optimization, we can form a PKLD predictive model by substituting the ground truth rates with our measured rates and the dwell time by the total dwell:

$$\mathbb{E}\left[\boldsymbol{\mathcal{E}}^{\mathrm{PKLD}}\left(T\right)\right] \approx \frac{1}{N}\sum_{c,h,w}\frac{1}{2\left(T^{0}_{h,w}+T_{h,w}\right)}. \tag{4.15}$$

Given our constraints, we can then formulate the minimization for the PKLD mask:

$$T^{*} \approx \underset{\substack{\sum_{h,w}T_{h,w}\leq\tau \\ T^{\min}_{h,w}\leq T_{h,w}\leq T^{\max}_{h,w}}}{\arg\min} \frac{1}{N}\sum_{c,h,w}\frac{1}{2\left(T^{0}_{h,w}+T_{h,w}\right)}. \tag{4.16}$$

Note that this optimization does not depend on the count rates at all. In fact, if the first scan was constant, the optimization of Eq. (4.16) is also a uniformly sampled volume:

$$T^{*}_{h,w} \approx \frac{\tau}{HW}. \tag{4.17}$$

The uniform is perhaps a gross simplification since there are the lower order terms that were dropped in Eq. (4.12). These terms are of $\left(\boldsymbol{\Psi}_{c,h,w}\cdot T^{0}_{h,w}\right)^{-2}$ and lower order. Since the rate is in the denominator in the expansion, the optimal dwell time allocation is inverse to the count rate.

Instead of the uniform mask, we suggest a cost function that takes this inverse relation into account. If we include more terms in the Taylor series expansion, the optimization becomes untenable in terms of calculation speed. This cannot be discounted since oftentimes the spectroscopists conducting the experiments have a limited time window to scan the painting. Therefore, we propose a different cost function simplification that accounts for the inverse relationship between count rate and sampling time allocation:

$$T^* \approx \underset{\substack{\sum_{h,w} T_{h,w} \leq \tau \\ T^{\min}_{h,w} \leq T_{h,w} \leq T^{\max}_{h,w}}}{\arg\min} \frac{1}{N} \sum_{c,h,w} \frac{\mathbf{\Lambda}^{-1}_{c,h,w}}{T^0_{h,w} + T_{h,w}}. \tag{4.18}$$

We will show in our experiments that this loss is able to better optimize the PKLD than that of a constant scan in some cases. Like the MSE mask, our intuition still holds that the more time that is allocated to any pixel, the lower the expected error.

Fig. 4.3c shows a PKLD mask for the same Hg L mask in Fig. 4.3a. This mask, like the MSE mask of Fig. 4.3b, has an average dwell time of 5 ms/px. Since the mask favors low count rates, we need to exclude the background. Only rates over 1200 counts/s were considered for the optimization. Slower rates are automatically given the minimum dwell time of 1 ms/px. Notice how the PKLD mask appears complementary to the MSE mask: the bright garment in the bottom is now dark, signaling that the PKLD is (likely) relatively low compared to the other regions. Thus, it requires less scan time to minimize the PKLD.

Perhaps an added benefit of this mask is that the edges are sometimes sampled longer than the interior of the regions of interest. This is due in part to the machine's point spread function that mix neighboring signals and the denoising algorithm, both of which

blur edges. The edges of these regions will inevitably sharpen with the longer sampling times.

4.3.3. Joint MSE and PKLD Mask

Given the MSE optimization of Eq. (4.9) and the PKLD optimization of Eq. (4.18), the two are in conflict with one another: the MSE mask prioritizes high count rates, while the PKLD mask prioritizes low count rates. What happens with a convex combination of the two mask designs? We need to analyze the effect of the weighting parameter for the joint optimization

$$T^* \approx \underset{\substack{\sum_{h,w} T_{h,w} \leq \tau \\ T^{\min}_{h,w} \leq T_{h,w} \leq T^{\max}_{h,w}}}{\arg\min} \frac{1}{N} \sum_{c,h,w} \frac{\rho\, \mathbf{\Lambda}_{c,h,w} + (1-\rho)\, \mathbf{\Lambda}^{-1}_{c,h,w}}{T^0_{h,w} + T_{h,w}} \tag{4.19}$$

for some $\rho \in [0, 1]$. One can quickly verify that count rates approaching zero or infinity incur an infinite penalty in the loss. In between, there exists some rate that is least prioritized, which can be found by minimizing the summand. The least penalized rate occurs where

$$\mathbf{\Lambda}_{c,h,w} = \sqrt{\frac{1-\rho}{\rho}}. \tag{4.20}$$

It can be difficult to know which rate should be given the smallest weight, so we instead weigh not by a convex combination, but by a gamma correction of the rates. Note that the exponent of the weighted rates $\mathbf{\Lambda}$ is 1 for the MSE mask in Eq. (4.9) and -1 for the PKLD mask in Eq. (4.18). We can instead use $\rho \in [-1, 1]$ as a weighting mechanism

between the two masks. The generalized mask optimization is then

$$T^* \approx \underset{\substack{\sum_{h,w} T_{h,w} \leq \tau \\ T^{\min}_{h,w} \leq T_{h,w} \leq T^{\max}_{h,w}}}{\arg\min} \frac{1}{N} \sum_{c,h,w} \frac{\mathbf{\Lambda}^{\rho}_{c,h,w}}{T^0_{h,w} + T_{h,w}} \tag{4.21}$$

where ρ controls the preference for the MSE mask ($\rho = 1$) or the PKLD mask ($\rho = -1$).

What is interesting is the case where $\rho = 0$, which is the uniform sampling mask that arose in Eq. (4.16). Here, the optimization does not depend on the rates at all. Perhaps this gives some comfort to spectroscopists in knowing that the default uniform raster scan attempts to weight the MSE and PKLD equally. Fig. 4.3d shows a uniform sampling mask applied to pixels whose rate is at least 1200 counts/s. There is a minimum dwell time of 1 ms/px, similar to the other masks in Fig. 4.3.

4.4. Optimization

As was mentioned previously, finding the optimal mask in a short amount of time is not only ideal for any algorithm, but strictly required for it to be adopted by XRF spectroscopists in the context of performing fast scans. Algorithms that do not immediately provide a sampling pattern takes valuable time away from collecting more data.

Before we begin the solution, note that the cost function is not indexed by the channel in the denominator. This is beneficial in an optimization sense since the weights can be added together prior to carrying out the minimization. Instead of optimizing a mask based on N values, we instead only need $N/C = HW$ values. We strategically rewrite

Algorithm 1 Sampling Mask Generation

Require: $\mathbf{\Lambda}, \tau, T^0, T^{\min}, T^{\max} \geq 0$
Ensure: $\sum_{h,w} T^*_{h,w} \leq \tau$

function MaskOpt($\mathbf{\Lambda}, \rho, \tau, T^0, T^{\min}, T^{\max}$)
$\mathbf{\Lambda}^{\rho}_{h,w} \leftarrow \sum_c \mathbf{\Lambda}^{\rho}_{c,h,w}$
$Q_{h,w} \leftarrow \sqrt{\mathbf{\Lambda}^{\rho}_{h,w}}$
$\tau_{\text{net}} \leftarrow \tau - \sum_{h,w} T^{\min}_{h,w}$

while $\tau_{\text{net}} > 0$ and any $\left(T^{\min}_{h,w} < T^{\max}_{h,w}\right)$ **do**

$D_i \leftarrow (h,w) : \frac{-\mathbf{\Lambda}^{\rho}_{h,w}}{\left(T^0_{h,w}+T^{\min}_{h,w}\right)^2} \leq \frac{-\mathbf{\Lambda}^{\rho}_{h,w}}{\left(T^0_{j,k}+T^{\min}_{j,k}\right)^2} \; \forall\,(j,k) \;\text{ and }\; T^{\min}_{h,w} < T^{\max}_{h,w}$
$R_i \leftarrow \sum_{j=1}^{i} T^0_{D_j} + T^{\min}_{D_j}$
$S_i \leftarrow \sum_{j=1}^{i} Q_{D_j}$

$K_i \leftarrow \frac{S_i}{Q_i} \cdot \left(T^0_{D_i} + T^{\min}_{D_i}\right) - R_i$
$k^* \leftarrow \max_{K_i \leq \tau} \; i$

$V_i \leftarrow \frac{Q_{D_i}}{S_{k^*}} \cdot (\tau_{\text{net}} + R_{k^*}) - \left(T^0_{D_i} + T^{\min}_{D_i}\right), \quad i \leq k^*$
$T^{\min}_i \leftarrow \min\left(T^{\min}_i + V_i,\; T^{\max}_i\right)$

$\tau_{\text{net}} \leftarrow \tau - \sum_{h,w} T^{\min}_{h,w}$
end while

$T^* \leftarrow T^{\min}$
return T^*

end function

Eq. (4.21) as

$$T^* \approx \underset{\substack{\sum_{h,w} T_{h,w} \leq \tau \\ T^{\min}_{h,w} \leq T_{h,w} \leq T^{\max}_{h,w}}}{\arg\min} \; \frac{1}{N} \sum_{h,w} \frac{\sum_c \mathbf{\Lambda}^{\rho}_{c,h,w}}{T^0_{h,w} + T_{h,w}} \tag{4.22}$$

with the channel summation separated.

There is independence of each entry in T from one another, which also aids in the optimization. The approach we take to solve Eq. (4.22) is therefore one of incremental allocation: keep allocating time to the pixels where the error will be reduced the most until there is no time left to allocate. We will first focus on the case where we ignore the minimum and maximum time constraints, *i.e.* $T_{h,w}^{\min} = 0$ and $T_{h,w}^{\max} = \infty$ for all (h, w).

4.4.1. No Individual Time Constraint Optimization

In order to find the pixels to allocate time to, we first denote the generic expected error as

$$\mathbb{E}\left[\boldsymbol{\mathcal{E}}\left(T\right)\right] = \frac{1}{N}\sum_{h,w}\frac{\sum_c \mathbf{\Lambda}_{c,h,w}^{\rho}}{T_{h,w}^{0} + T_{h,w}} \tag{4.23}$$

for simplification. The greatest loss reduction occurs at the pixel with the greatest negative derivative. These derivatives of T are quickly found as:

$$\frac{\partial}{\partial T_{h,w}}\mathbb{E}\left[\boldsymbol{\mathcal{E}}\left(T\right)\right] = \frac{-\sum_c \mathbf{\Lambda}_{c,h,w}^{\rho}}{N\left(T_{h,w}^{0} + T_{h,w}\right)^2}. \tag{4.24}$$

Again, notice that the derivatives of the dwell entries are also independent of one another. Weighted rates $\mathbf{\Lambda}^{\rho}$ are not dependent on T.

If we sort the pixels by their derivative in ascending order at $T_{h,w} = 0$ for all entries, we now have a list of how the pixels should be prioritized in the optimization—the first entry has the greatest negative derivative, *i.e.* the greatest expected reduction in error given an incremental time allotment $\delta \ll 1$. How much time should then be allocated to this first pixel? We keep allocating time until the derivative of the first pixel is equivalent to the derivative of the second pixel without any time allocation. At this point, it is

equally beneficial to allocate δ to these pixels. Let us denote $i \in \{1, 2, \ldots, HW\}$ as the indexing scheme where i corresponds to the i^{th} most negative derivative at location (h_i, w_i). Mathematically, we need to know when

$$\frac{\partial}{\partial T_i}\mathbb{E}\left[\boldsymbol{\mathcal{E}}\left(T_i\right)\right] = \frac{\partial}{\partial T_j}\mathbb{E}\left[\boldsymbol{\mathcal{E}}\left(0\right)\right], \tag{4.25}$$

where $i < j$. The dwell time T_i in Eq. (4.25) determines how much time to allocate to pixel i before considering pixel j. Fortunately, the equation is a quadratic, which can be easily solved. The nonnegative solution

$$T_i = \sqrt{\frac{\sum_c \boldsymbol{\Lambda}^{\rho}_{c,i}}{\sum_c \boldsymbol{\Lambda}^{\rho}_{c,j}}} \cdot T_j^0 - T_i^0, \tag{4.26}$$

is when there is equal reduction in the loss between pixels i and j.

Of course, there are more than two pixels to optimize. For any pixel k, the amount of time that needs to be allocated to pixels $1, \ldots, k-1$ prior to pixel k is a sum of the the results found in Eq. (4.26):

$$\tau_k = \sum_{i=1}^{k} \left(\sqrt{\frac{\sum_c \boldsymbol{\Lambda}^{\rho}_{c,i}}{\sum_c \boldsymbol{\Lambda}^{\rho}_{c,k}}} \cdot T_k^0 - T_i^0 \right). \tag{4.27}$$

We include pixel k in the summation since the summand when $i = k$ is zero, and it simplifies the optimization in the coming steps.

This length of time τ_k must not exceed the total scan time. Thus, we need to find the maximum value for k given the total time constraint. Denote k^* as the maximal k that

satisfies

$$\sum_{i=1}^{k} \left(\sqrt{\frac{\sum_c \mathbf{\Lambda}^{\rho}_{c,i}}{\sum_c \mathbf{\Lambda}^{\rho}_{c,k}}} \cdot T^0_k - T^0_i \right) \leq \tau. \tag{4.28}$$

All pixels with indices $i \leq k^*$ are considered for new time allocation, but all indices $i > k^*$ are excluded and assigned zero time.

With the set of pixels to optimize known, all that is left is to find the appropriate time allocation that uses all the time available. We find some average value λ that would maximize the time usage. Hypothetically, if $\mathbf{\Lambda}^{\rho}_{c,k^*} = \lambda$ for all c, this would be the average weighted rate where all the time is allocated. This value is found via equality of Eq. (4.28)

$$\sum_{i=1}^{k^*} \left(\sqrt{\frac{\sum_c \mathbf{\Lambda}^{\rho}_{c,i}}{\sum_c \lambda}} \cdot T^0_{k^*} - T^0_i \right) = \tau \tag{4.29}$$

with a solution of

$$\lambda = \frac{1}{C} \left(\frac{T^0_{k^*} \cdot \sum_{i=1}^{k^*} \sqrt{\sum_c \mathbf{\Lambda}^{\rho}_{c,i}}}{\tau + \sum_{i=1}^{k^*} T^0_i} \right)^2 . \tag{4.30}$$

If we substitute Eq. (4.30) into Eq. (4.29), we have a simplified summation that uses all the available time to scan:

$$\sum_{i=1}^{k^*} \left[\frac{\sqrt{\sum_c \mathbf{\Lambda}^{\rho}_{c,i}}}{\sum_{j=1}^{k^*} \sqrt{\sum_c \mathbf{\Lambda}^{\rho}_{c,j}}} \cdot \left(\tau + \sum_{j=1}^{k^*} T^0_j \right) - T^0_i \right] = \tau \tag{4.31}$$

where each summand is the optimal amount of time to scan each pixel. We have a final minimization solution of

$$T_i^* = \begin{cases} \frac{\sqrt{\sum_c \mathbf{\Lambda}_{c,i}^{\rho}}}{\sum_{j=1}^{k^*} \sqrt{\sum_c \mathbf{\Lambda}_{c,j}^{\rho}}} \cdot \left(\tau + \sum_{j=1}^{k^*} T_j^0\right) - T_i^0, & i \leq k^* \\ 0, & i > k^* \end{cases} \tag{4.32}$$

which includes the scan time of every pixel. The fractional component of Eq. (4.32) is a weighting term that sums to unity across the k^* pixels. What is being weighted is the total scan time *including* the initial scan times of the pixels included in the optimization set. The subtraction of the initial scan time ensures that the subsequent scan does not exceed the time allotted. No matter the optimization objective, we can find the optimal allocation using this algorithm given that there are no individual pixel time constraints.

4.4.2. Minimum and Maximum Dwell Time Constraints

There are only a few modifications needed to incorporate the minimum dwell times $T^{\min}$ and maximum dwell times $T^{\max}$. The minimum dwell time constraint can be accounted for in the start of the optimization steps. If we assign

$$T^0 \leftarrow T^0 + T^{\min} \tag{4.33}$$

and

$$\tau \leftarrow \tau - \sum_{h,w} T_{h,w}^{\min}, \tag{4.34}$$

then we optimize the sampling mask over the time that is left over after allocating the mandatory minimum time. The optimal dwell mask is then found using the remaining time. To get the true optimal sampling mask, we simply add the minimum dwell time back to the optimal mask:

$$T^* \leftarrow T^* + T^{\min} \tag{4.35}$$

calculates the final mask accounting for the minimum scan time.

Accounting for the maximum dwell time is also a simple solution by repeating the optimization. First, carry out the optimization steps as if there were no maximum time constraints. Next, assign the optimized values or the maximum values to the minimum dwell time, whichever is lesser:

$$T^{\min} \leftarrow \min\left(T^{\min}_{h,w} + T^*_{h,w},\ T^{\max}_{h,w}\right). \tag{4.36}$$

If there is no clipping by the maximum dwell time, then we are done and assign the new minimum dwell time as the optimal dwell time. Otherwise, there is remaining time that can be allocated to pixels that have not reached the maximum dwell time:

$$\tau \leftarrow \tau - \sum_{h,w} T^{\min}_{h,w}, \tag{4.37}$$

and an additional iteration of the algorithm is needed. Pixel locations where $T^{\min}_{h,w} = T^{\max}_{h,w}$ are excluded/ignored from the next iteration since the maximum amount of time is already allocated. This process is repeated until all the time is allocated or all the pixels are allotted their maximum dwell times. Algorithm 1 provides pseudocode for the mask

computation. The algorithm is quite fast, regularly producing results for our XRF volume in under one second.

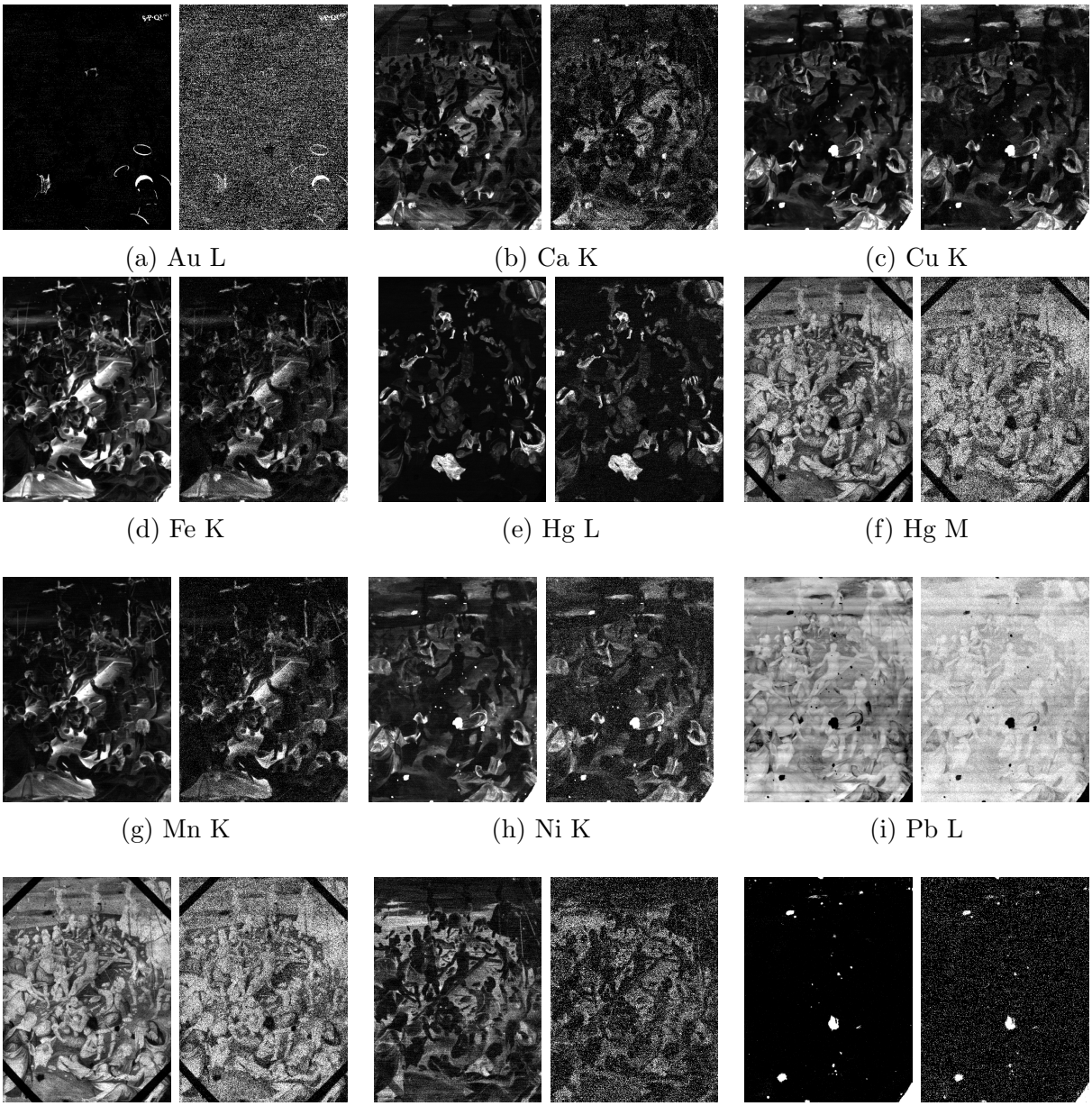
4.5. XRF Preprocessing

While we have derived the algorithm for generating the sampling mask, there are additional considerations before using it. After collecting the XRF data, it is decomposed into elemental maps that show the location and relative amount of the element present. Generating high quality elemental maps is very often the goal for further XRF analysis. Some elements typically have a strong presence (*i.e.* high PSNR), while other elements appear noisier in the elemental maps. Instead of an optimization over the entire XRF volume, it would be more beneficial for XRF analysts to select which elemental maps should be further sampled to reduce noise.

Aside from the elemental maps, the noise in the initial scan also may pose a problem. The rates measured $\mathbf{\Lambda}$ are not perfect, and this can affect the performance of the mask sampling algorithm. Prior to applying the sampling algorithm, it may be beneficial to apply a denoising algorithm to the data that will be optimized. We will discuss both of these concerns.

4.5.1. Elemental Maps

The elemental maps are generated from a nonnegative matrix factorization of the XRF volume. Fig. 4.3a provides the denoised, 5 ms/px Hg L elemental map for example. Recall that each element emits its own characteristic XRF spectrum; this aids in identifying elements present in the painting. One of copper's strongest peaks named Kα_1 for example



(a) Au L (b) Ca K (c) Cu K

(d) Fe K (e) Hg L (f) Hg M

(g) Mn K (h) Ni K (i) Pb L

(j) Pb M (k) Sn L (l) Ti K

Figure 4.4. XRF maps of *Raising of the Cross*, depicted in Fig. 4.1. These maps are from the original XRF scan of 100 ms/px (left) and a simulated scan of 2 ms/px (right). We invite the reader to zoom in to better view the maps and the effects of noise.

is around 8.05 keV, and Pb's Lα_1 peak is around 10.55 keV. Once an XRF volume is acquired, the spectroscopist looks at the sum spectrum, which is a summation of the spectra at all pixel locations. The sum spectrum is used instead of identifying elements in each individual pixel since it is less noisy.

Once the peaks in the sum spectrum are assigned an elemental line, a platform like PyMca [15] generates a dictionary $D^{\text{el}} \in \mathbb{R}_+^{C\times M}$ where the M columns are Gaussian curves fitted to the peaks in the sum spectrum. The abundance $\mathbf{A}^{\text{el}} \in \mathbb{R}_+^{M\times H\times W}$ is found by a nonnegative factorization of the XRF volume $\mathbf{\Lambda}$ by the dictionary D^{el}:

$$\mathbf{A}^{\text{el}} = \arg\min_{\mathbf{A}\geq 0} \frac{1}{N_{\mathbf{A}}} \sum_{i\in(c,h,w)} \left(\mathbf{\Lambda}_i - \left(D^{\text{el}}\,\mathbf{A}\right)_i\right)^2. \tag{4.38}$$

To avoid unnecessarily introducing more variables, the volume $D\mathbf{A}$ is found similarly to matrix multiplication:

$$\left(D^{\text{el}}\mathbf{A}\right)_{c,h,w} = \sum_{m=1}^{M} D^{\text{el}}_{c,m} \cdot \mathbf{A}_{m,h,w}. \tag{4.39}$$

The factorization is nonnegative since negative elemental quantities bears no physical meaning.

Once the maps are generated, XRF analysis can be conducted since the location of the elements are known as well as the relative quantities. Typically, there are a select few elements that are of importance such as copper, iron, mercury, and lead. Rhodium for instance is a common anode material to generate x-rays for XRF measurements. Photons originating from rhodium can reflect back into the detector and appear as peaks in the sum spectrum. Rhodium is included in D^{el} with the understanding that the painting

does not contain it. We identified twelve elemental lines in the original 100 ms/px scan of *Raising of the Cross*, the maps of which are shown in Fig. 4.4 on the left of each subfigure. These elemental maps are used in part for XRF analysis of, say, pigment identification.

The entire or partial XRF volume $\mathbf{\Lambda}$ can be used as the input to the mask design, but if there are certain maps of $\mathbf{A}^{\text{el}}$ that are of more interest to the spectroscopist, it is best to use those maps in place of the volume. These maps can optionally be weighted since the magnitude scale of the maps can vary; Ca K for example has an average count rate of 310 counts/s while Pb L has a mean of 52600 counts/s. An MSE mask would highly favor the Pb L map if the two were jointly optimized. Since elemental map quality is often the highest priority for XRF spectroscopists and analysts, we can perform a substitution of

$$\mathbf{\Lambda} \leftarrow \text{diag}\left(\mathbf{w}\right) \mathbf{A}^{\text{el}} \tag{4.40}$$

as the input to the mask optimization. Weights $\mathbf{w} \in \mathbb{R}_{+}^{M \times 1}$ is a weighting vector; $\text{diag}\left(\mathbf{w}\right)$ is a diagonal matrix where the diagonal values are $\mathbf{w}$. The product is calculated exactly as the matrix-tensor multiplication of Eq. (4.39). Maps can be excluded from the optimization by zeroing the corresponding entry in $\mathbf{w}$. The magnitude of $\mathbf{w}$ also has no influence on the optimization as scaling factors do not change the optimization results.

4.5.2. Denoising

In a fast scan, fewer photons are recorded, so the resulting maps are noisier. The images on the right in Fig. 4.4 show the results of a fast XRF scan of 2 ms/px, a 50x speedup in the acquisition. The Hg M and Pb M maps (Figs. 4.4f and 4.4j respectively) show high levels of noise in the foreground due to the fast scan. If we were just to use these maps as

inputs to, say, the MSE mask algorithm, we would oversample the overestimated rates, and undersample the underestimated rates. In order to equitably distribute the sampling time, we propose to denoise the maps after each pass of the XRF gantry as a correction mechanism. Even if denoising is not done in the final step, it is important to obtain a better prediction of where the high and low count rates are.

We provide a volumetric denoising algorithm [30] detailed in Chapter 2. As discussed in the related work, there are serious computational barriers that makes the algorithm infeasible considering the time constraint to scan the painting. A subsequent work of ours performs denoising on the elemental maps directly. This is a faster calculation that often completes in under a minute on a CPU. We will adapt their algorithm to our work, noting that this can be replaced by another denoising algorithm.

There are two terms in their optimization: (1) the data fidelity term, and (2) a regularization term. Both of these terms were designed for Poisson image optimization. The data fidelity term is the zero-shifted Poisson negative log likelihood (PNLL), which is defined to penalize deviations of the free variable from the observed data. For elemental maps, the PNLL at any pixel $i \in (m, h, w)$ is

$$\mathcal{P}_{\text{NLL}}\left(\mathbf{A}_i, \mathbf{A}_i^{\text{el}}\right) = \mathbf{A}_i - \mathbf{A}_i^{\text{el}} \cdot \ln \mathbf{A}_i \tag{4.41}$$

where $\mathbf{A}_i$ is the free variable, and $\mathbf{A}_i^{\text{el}}$ contains the measured count rates. The zero-shifted PNLL is equivalent to the summand of the PKLD defined in Eq. (4.10). The work by us previously assumed that the dwell time is constant for each pixel, but our algorithm now clearly does not assume this. We decide to weigh the PNLL by the dwell time to account

for the confidence in the measurements. Our PNLL loss is defined as

$$\mathcal{L}_{\mathrm{NLL}}\left(\mathbf{A}, \mathbf{A}^{\mathrm{el}}; T^0\right) = \frac{1}{N_{\mathbf{A}}} \sum_{m,h,w} \mathcal{P}_{\mathrm{NLL}}\left(T^0_{h,w} \cdot \mathbf{A}_{m,h,w},\ T^0_{h,w} \cdot \mathbf{A}^{\mathrm{el}}_{m,h,w}\right) \tag{4.42}$$

where $N_{\mathbf{A}} = MHW$ is the number of entries in the map volume. The incorporation of time as a multiplier is identical to using the PNLL of the elemental map *counts* instead of the count rates.

The regularization term is included to promote spatial smoothness in the image. The PKLD is a Poisson penalty term that finds the statistical distance from one rate to another, but it is asymmetric. The Jeffreys divergence of two Poisson variables is a symmetrized version of the PKLD. This divergence is a way to compare two spatially neighboring values when neither rate is considered a "ground truth" measurement. For two rates $\mathbf{A}_i$ and $\mathbf{A}_j$, the Poissonian Jeffreys divergence (PJD) is

$$\mathcal{P}_{\mathrm{JD}}\left(\mathbf{A}_i, \mathbf{A}_j\right) = \left(\mathbf{A}_i - \mathbf{A}_j\right) \cdot \left(\ln \mathbf{A}_i - \ln \mathbf{A}_j\right). \tag{4.43}$$

The absolute difference is weighted by the difference in magnitude, which accounts for the varying count rate magnitudes within an elemental map.

The authors provide an optional weighting scheme of the PJD. Based on an RGB image of the painting $\mathbf{I} \in [0,1]^{3\times H\times W}$ registered to the XRF data, adaptive weights

$$\Omega^H_{h,w} = \exp\left(-\beta \sum_{c=1}^{3} \left(\mathbf{I}_{c,h+1,w} - \mathbf{I}_{c,h,w}\right)^2\right) \tag{4.44}$$

and

$$\Omega_{h,w}^{W} = \exp\left(-\beta \sum_{c=1}^{3} \left(\mathbf{I}_{c,h,w+1} - \mathbf{I}_{c,h,w}\right)^2\right) \tag{4.45}$$

suppress large smoothing penalties that arise in edges. The edges are detected in the vertical and horizontal directions. Hyperparameter $\beta \geq 0$ controls the sensitivity of the edge suppression ($\beta = 0$ is the unweighted case). During the denoising optimization, this mitigates smoothing in regions where there are known edges—we expect XRF changes in these areas since the edges often correspond to pigment changes.

The average weighted PJD loss is then

$$\mathcal{L}_{\text{JD}}\left(\mathbf{A};\mathbf{I}\right) = \mathcal{L}_{\text{JD}}^{H}\left(\mathbf{A};\mathbf{I}\right) + \mathcal{L}_{\text{JD}}^{W}\left(\mathbf{A};\mathbf{I}\right) \tag{4.46}$$

where

$$\mathcal{L}_{\text{JD}}^{H}\left(\mathbf{A};\mathbf{I}\right) = \frac{1}{N_{\mathbf{A}}} \sum_{m=1}^{M} \sum_{h=1}^{H-1} \sum_{w=1}^{W} \Omega_{h,w}^{H} \cdot \mathcal{P}_{\text{JD}}\left(\mathbf{A}_{m,h,w}, \mathbf{A}_{m,h+1,w}\right) \tag{4.47}$$

for the vertical dimension, and

$$\mathcal{L}_{\text{JD}}^{W}\left(\mathbf{A};\mathbf{I}\right) = \frac{1}{N_{\mathbf{A}}} \sum_{m=1}^{M} \sum_{h=1}^{H} \sum_{w=1}^{W-1} \Omega_{h,w}^{W} \cdot \mathcal{P}_{\text{JD}}\left(\mathbf{A}_{m,h,w}, \mathbf{A}_{m,h,w+1}\right) \tag{4.48}$$

for the horizontal dimension.

The full denoising optimization sums the losses from Eqs. (4.42) and (4.46) to get

$$\mathbf{A}^{*} = \underset{\mathbf{A}\geq 0}{\arg\min}\ \mathcal{L}_{\text{NLL}}\left(\mathbf{A}, \mathbf{A}^{\text{el}}\right) + \alpha_{\text{JD}} \cdot \mathcal{L}_{\text{JD}}\left(\mathbf{A};\mathbf{I}\right) \tag{4.49}$$

where $\alpha_{\text{JD}} \in \mathbb{R}_+$ determines the smoothing strength. The optimization is carried out via gradient descent and on each elemental map separately as the authors suggest. The optimized maps can then be used in place of the raw count rates:

$$\mathbf{\Lambda} \leftarrow \text{diag}\,(\mathbf{w})\,\mathbf{A}^* \tag{4.50}$$

where $\text{diag}\,(\mathbf{w})$ are the elemental map weights as previously defined, but as the diagonal entries of a square matrix. Now, the dwell times for the next scan are based on a more accurate estimate of the elemental map rates.

We have outlined our adaptive sampling algorithm. After an initial fast XRF scan, the (denoised) elemental distribution maps determine the sampling strategy for the next scan. Depending on the choice of ρ in Eq. (4.21), the sampling mask is designed for minimizing the MSE and/or PKLD error. In the next section, we provide results.

4.6. Experiments

Here, we examine the mask's performance on the XRF data of *Raising of the Cross*. The data was originally scanned at 100 ms/px to get an XRF volume of $\mathbf{\Lambda} \in \mathbb{R}_+^{4096\times566\times423}$. Scanning this volume took about 6 hr 40 min. Imagine a scenario where we have not yet scanned this painting, but we only have 40 min before the painting is needed back for public viewing. This is one tenth of the original scan time, so the average dwell time is 10 ms/px. Our spectroscopist wants the highest quality raw data for XRF analysis. We will compare the quality of the maps generated from our method against a constant dwell time. The image quality is measured in terms of the root mean squared error (RMSE), which measures absolute error, as well as the PKLD, which measures relative error.

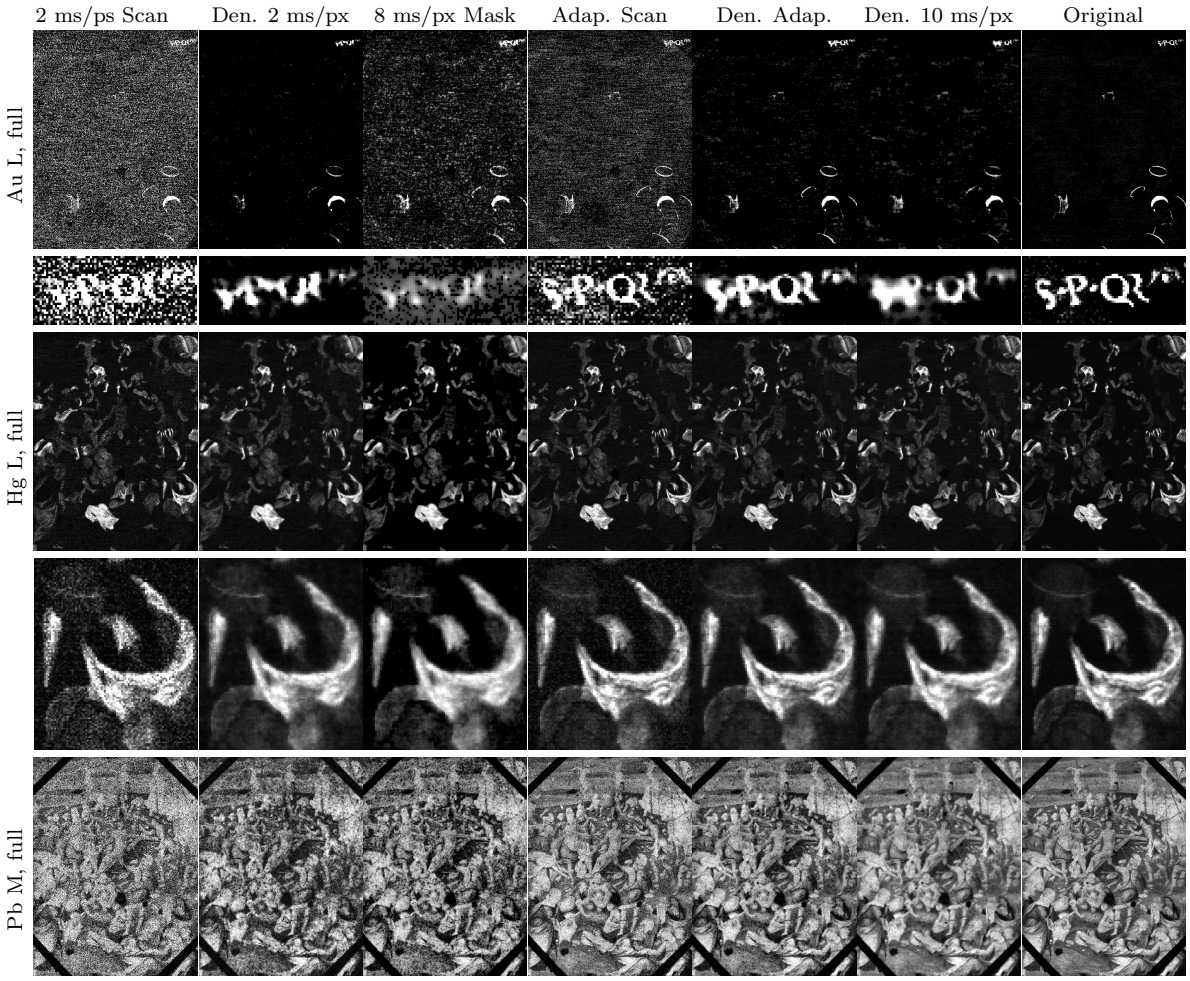


Figure 4.5. Select results showing the process of the algorithm. Column 1 is the initial 2 ms/px scan. Column 2 is the denoised Column 1 used to generate the mask. Column 3 is the MSE mask for 8 ms/px average. Column 4 is the raw, cumulative data. Column 5 is the final denoised data. Column 6 is a standard 10 ms/px raster scan for comparison. Column 7 is the maps from the original 100 ms/px scan. We invite the reader to compare the last three columns. Results for Au L are in the first two rows, Hg L then next two, and Pb M the last.

4.6.1. Data Updating

There are some update steps we must first cover. Once a new XRF volume is sampled, we need to update the previous data. Let the superscript of the following variables denote

the iteration number; these variables are accumulated amounts. Variables without a superscript will denote the newly collected data independent of prior iterations. We first update the time by

$$T^{k+1} = T^k + T, \tag{4.51}$$

then the XRF volume by

$$\mathbf{\Lambda}^{k+1}_{c,h,w} = \frac{T^k_{h,w} \cdot \mathbf{\Lambda}^k_{c,h,w} + T_{h,w} \cdot \mathbf{\Lambda}_{c,h,w}}{T^{k+1}_{h,w}}. \tag{4.52}$$

These variables accumulate the total dwell time and measured rate over all iterations. The elemental maps are then updated via Eq. (4.38) with the new rates $\mathbf{\Lambda}^{k+1}$. They are *not* updated similarly to Eq. (4.52) since the endmembers of D^{el} are adjusted with new data.

4.6.2. Optimizing One Elemental Map

The optimization of a single elemental map (*i.e.* $\mathbf{w}$ is a standard unit basis vector) as opposed to multiple maps provides the greatest gains for that map. Generally, the more maps that are included in the optimization, the smaller the gains since the elemental maps have different high and low rate regions (and therefore higher and lower priority regions). Single map optimization has grounding in reality too—sometimes the spectroscopist is interested in a specific elemental distribution [53]. Here, we show the maximum potential gains that can be achieved using our algorithm on one elemental map.

We compare different adaptive scans against a fast raster scan of 10 ms/px, which is ten times faster than the original scan. There are two different initial uniform scans: a 2

ms/px scan and a 5 ms/px scan. After each scan, we denoise the elemental maps; these are used for the mask design instead of the raw elemental maps. The remaining time to reach an average of 10 ms/px is reserved for the adaptive mask. Within each time distribution, we apply three different masks: the PKLD mask ($\rho = -1$), uniform mask ($\rho = 0$), and MSE mask ($\rho = 1$).

Since the PKLD mask focuses on low counts, we establish a count rate minimum for each elemental map to avoid oversampling the background. The pixels that meet or exceed this threshold are considered to be a part of the foreground. The threshold is applied to the uniform and MSE masks as well. Additionally, we set the minimum dwell time for every pixel as 1 ms to guarantee that each pixel receives some additional scan time in an effort to simulate the maximum speed of the gantry. We do not set the maximum dwell time for foreground pixels, but note that no dwell time ever exceeded the original scan time of 100 ms/px. Once the mask for a single elemental map is found, we resample the XRF volume, following the steps outline in Eqs. (4.51) and (4.52) for updating the time and XRF volume respectively. The elemental maps are updated using PyMca with the new XRF volume, then denoised as a final step.

There are two evaluation criteria: root mean-squared error (RMSE) and PKLD, which provide a measure of the absolute and relative error respectively. There are also two sets of pixel data used for comparison: the foreground pixels only, as well as all the pixels. The foreground pixel comparison is appropriate since there are regions in all the elemental maps where there are negligible rates; knowledge of an element's absence in a region of interest is usually sufficient. We note that the foreground error is calculated using the foreground pixels of the ground truth, which may have discrepancies with the foreground

determination in the denoised maps. The results for each map of interest are detailed in Table 4.1 for foreground errors and Table 4.2 for all errors. Four different intermediary results are shown in Fig. 4.5. Further, we test the algorithm when there is no denoising after the initial scan. The adaptive scanning mask is then based on the raw elemental map data. These results are in Table 4.3 for foreground errors and Table 4.4 for all errors. Two additional tables, Tables 4.5 and 4.6, show the foreground and all errors respectively using denoising after the first scan, but not after the second scan like Tables 4.1–4.4. The errors here are calculated from the raw count rates of all the collected data that the spectroscopist would actually measure.

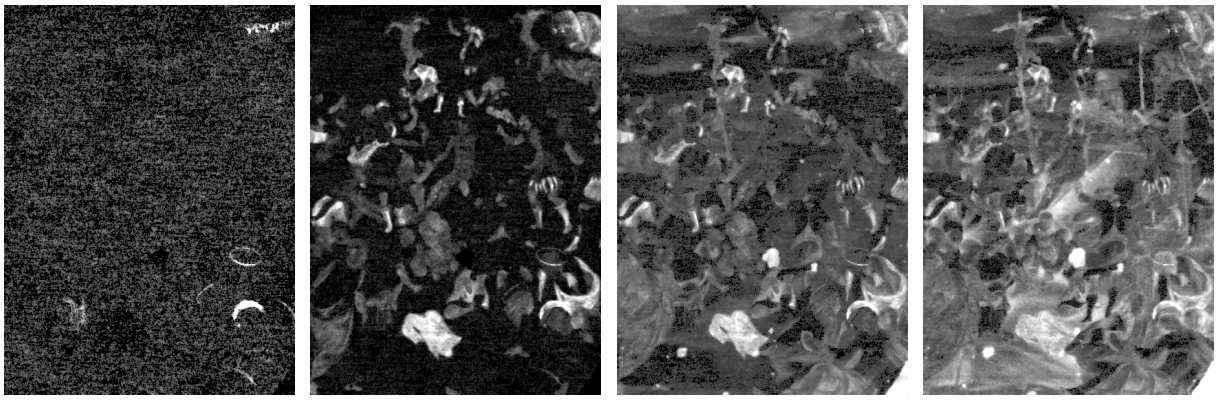
4.6.2.1. Analysis. We start our analysis by comparing across Tables 4.1–4.6. The benefits of denoising at the final step are apparent since the non-denoised errors of Tables 4.5 and 4.6 are all greater than that of the denoised maps. However, we do acknowledge that denoising can be unappealing for some spectroscopists. With this assumption, we still tend to outperform, particularly with an initial scan of 5 ms/px and an MSE mask for the remaining 5 ms/px average scan. While this mask performed best in 13 errors to the 10 ms/px uniform's 5, the 5 ms/px adaptive scan outperformed the uniform mask in 18 out of 24 errors. Our mask shows that it can often produce the best raw data compared to a standard raster scan.

Our next analyses focus on Tables 4.1–4.4 where we perform denoising on the data after the second scan. First note that the denoised masks tend to outperform the non-denoised masks. By removing some of the noise prior to generating the mask, more time is allocated to underestimated pixel locations. Without the denoising, the mask assumes that the actual rate is quite low, and assigns an incorrect prioritization. This is

particularly noticeable in maps with high levels of noise (*e.g.* Sn L). In maps of low noise, we see minor improvements (*e.g.* Fe K), or even minor detriments to using the denoised map (*e.g.* Cu K). For fast XRF scans, we recommend denoising the elemental maps. However, as the scans last longer and longer, it may be acceptable to forgo the denoising steps. Cu K and Fe K have a high PSNR, so denoising offers minimal improvements to the mask design.

When comparing the different initial dwell times, the results vary by mask. In maps of low levels of noise, it is more beneficial to design the mask based on a faster scan (*e.g.* Cu K). Of course, if the data is less noisy, the mask will be closer to the true underlying mask, and therefore will have a better overall performance. Maps with higher noise levels prefer the longer dwell times. Even after using the denoising algorithm, the noise levels may still be too high to get an accurate count rate estimation. The Hg M map is a good example where the noise levels are so high that it is best to perform a uniform scan given the allotted time. The Ca K map has a moderate level of noise, and given an appropriate initial scan time of 5 ms/px, the adaptive mask performs better than the 10 ms/px uniform scan in terms of the MSE. An initial scan time of 2 ms/px is too fast for this mask as it performs worse compared to the 5 ms/px initial scan.

Within each time split, the MSE mask often outperforms the other masks in both RMSE and even PKLD; this is particularly true for the maps with low levels of noise. Noisier maps tend to have a preference for either the uniform mask or the PKLD mask. Relative errors tend to be amplified in these maps since the order of magnitudes can differ between the estimates and ground truth. The lower count regions mostly have these large



(a) Au L mask, display range $[1, 40]$ ms

(b) Au L + Hg L mask, display range $[1, 31]$ ms

(c) Au L + Hg L + Cu K mask, display range $[1, 22]$ ms

(d) Au L + Hg L + Cu K + Fe K mask, display range $[1, 18]$ ms

Figure 4.6. MSE masks with an average of 8 ms/px designed from denoised 2 ms/px scan elemental maps. Each following mask includes an additional elemental map. (a) Au L mask alone. (b) Joint Au L and Hg L mask. (c) Joint Au L, Hg L, and Cu K mask. (d) Joint Au L, Hg L, Cu K, and Fe K mask.

magnitude differences, which also contribute to a larger RMSE. If the masks focus the dwell time here, they can effectively minimize the RMSE and PKLD at the same time.

4.6.3. Optimizing Multiple Elemental Maps

Multiple maps can be optimized at the same time, but there are diminishing returns with each additional map. We choose to optimize the maps that are the most susceptible to our algorithm to illustrate the case for using multiple maps. Elemental maps Au L, Cu K, Fe K, and Hg L showed some of the greatest potential for our algorithm.

We compare four different sampling masks to the standard uniform scan: (1) Au L alone, (2) Au L and Hg L, (3) Au L, Hg L, and Cu K, and (4) Au L, Hg L, Cu K, and Fe K. Specifically, each 8 ms/px average mask was found from the denoised elemental maps $\mathbf{A}^*$ at an initial rate of 2 ms/px. We only consider the MSE mask optimization here

since it performed the best for these elements in the single map optimization case. In order to balance the weights between each map, we found it best to normalize each map such that the maximum rates of each map are identical. If we let Θ denote the maps to design a mask around, map weight vector $\mathbf{w}$ is defined as

$$\mathbf{w}_i = \frac{\max\left(\{\mathbf{A}_j^*\}_{j\in\Theta}\right)}{\max\left(\mathbf{A}_i^*\right)} \text{ for } i \in \Theta \tag{4.53}$$

and is zero otherwise.

The four masks are displayed in Fig. 4.6. Notice how the high count regions of each map are represented in the joint mask. The Au L and Hg L mask of Fig. 4.6b for example primarily resembles the Hg L mask, but it also includes the lettering from the Au L map in the top right for example (which is not present in the Hg L map). Similarly, there are many high and medium count rate regions in the Cu K map represented in Fig. 4.6c such as the blue sky. The Fe K map has high count rates in the wooden crosses as well as the weaponry, and these details are highlighted in the mask of Fig. 4.6d.

As more maps are added even beyond the four here, it becomes clear why a uniform mask generally works so well: there are differing areas of high count rates in each map. This decreases the range of dwell times with an increasing number of mask elements. This is especially true when the weighting scheme of Eq. (4.53) is applied because we normalize the rates so that no one map can overpower any other. The standard deviation of the dwell times understandably decreases as well—from 6.86 ms in the Au L map to 3.18 ms in the four map mask. Both of these facts suggest that with more maps added, the more each dwell time will converge towards the mean.

4.6.3.1. Analysis. Table 4.7 contains the foreground errors of these masks. A primary conclusion is that the best performing mask for each element is when that element is first included in the mask optimization. The Au L mask alone is best for Au L, but as more elements are included in the mask design, the Au L map increases in RMSE. The dwell times are more and more diverted from the high Au L count rate regions. The other masks also enjoy improvements, albeit diminishing, when their respective map is added to the optimization. When Hg L is included, there is a 21% decrease in RMSE over the standard uniform scan, but this is only a portion of the 32% decrease when Hg L is optimized alone. For Cu K, there is a 14% decrease when three maps are included in the mask, which is less of a decrease than 34% for the sole Cu K mask. Fe K when it is added to a four-map mask sees a 7% decrease as opposed to a 17% decrease. Interestingly, even with four different elemental maps incorporated into the mask, they all outperformed the standard raster scan in terms of RMSE, and all but Au L show improvements in the PKLD.

We also illustrate the effects of these masks on the elements that were *not* included in the mask. The Fe K map decreases in RMSE and PKLD with each additional map, which supports the theory around the strength of the uniform mask: every additional map pushes the mask closer to a uniform scan. The same can be said for the Cu K map in that it performs worst with the Au L map alone, yet sees an error reduction when the Hg L map is included.

Overall, these results show that there is great potential for our algorithm in terms of optimizing certain elemental maps and in terms of a speedy acquisition. Future work will test the mask design in a real world scenario.

4.7. Conclusion

We have provided an algorithm to smartly acquire XRF samples. After a first scan, an adaptive mask can be designed based on the collected XRF data. The choice of mask is also adjustable in many ways: we are able to design masks that focus on high count rates, low count rates, or equally amongst the rates. Control of the mask is even more refined as it allows for setting a dwell time range for each individual pixel. This allows spectroscopists to adjust the sampling mask to best suit their gantries. These masks work well, particularly the MSE-based mask in minimizing the RMSE and PKLD errors. A negligible amount of time is needed to carry out the mask optimization, allowing for a practical implementation in existing gantry systems.

Denoising plays a key role as well. After the first scan, it is often beneficial to denoise the maps prior to generating the mask. In particular, noisy maps see a benefit in a denoised sampling mask to mitigate erroneous oversampling and undersampling. Once the second batch of XRF is recorded according to the adaptive sampling mask, further denoising provides the greatest benefit.

While we did not discuss here, future work will look at finding the optimal number of scans. Perhaps three, four, or even more scans would be even more beneficial than a two pass scheme. The number of maps included in the mask design at each stage would be an interesting study as well. Testing our mask design in domains outside of XRF imaging is also a possibility that should be explored. Until then, we believe our algorithm is a great step in making adaptive XRF scanning more practical for spectroscopists, whether it is used for speedy acquisitions or reducing error in the elemental maps in full length scans.

Foreground Pixel Errors with Denoising after each Scan								
		2 + 8 ms/px Mask			5 + 5 ms/px Mask			10 ms/px
		PKLD	Uniform	MSE	PKLD	Uniform	MSE	Uniform
Au L	RMSE	194.76	190.27	**154.63***	225.44	197.15	173.13*	214.52
	PKLD	50.99	49.26	47.12*	51.40	50.42	49.25*	**41.67**
Ca K	RMSE	191.09	187.56	185.84*	154.69	144.51	**143.21***	148.76
	PKLD	63.65*	63.84	65.00	37.42	36.84*	39.65	**28.20**
Cu K	RMSE	1201.85	848.17	**749.06***	1239.68	962.72	766.23*	1082.28
	PKLD	33.61	26.33	**24.84***	37.58	29.03	26.27*	35.51
Fe K	RMSE	364.12	336.56	**323.57***	402.87	348.88	325.87*	389.95
	PKLD	24.67	23.68*	24.03	21.28	**18.79***	18.93	22.56
Hg L	RMSE	553.56	476.50	**435.86***	626.46	527.57	485.46*	639.67
	PKLD	45.22	38.80	**36.82***	53.41	46.23	44.59*	64.97
Hg M	RMSE	261.50*	262.84	266.54	207.13	198.27*	200.73	**184.99**
	PKLD	76.30*	78.66	81.21	37.13	35.83*	40.42	**27.42**
Mn K	RMSE	134.58	129.86	115.08*	140.56	123.00	**113.51***	129.42
	PKLD	13.89	13.78	13.04*	11.89	**10.56***	10.61	10.98
Ni K	RMSE	171.51	152.15	**138.62***	176.23	155.84	138.72*	163.48
	PKLD	11.27	10.09	9.52*	11.46	10.09	**9.34***	10.58
Pb L	RMSE	1642.26	1646.78	1640.72*	1561.92*	1587.95	1640.64	**1552.87**
	PKLD	25.34*	25.55	25.48	**22.84***	23.98	26.22	23.03
Pb M	RMSE	584.16*	590.69	592.05	**518.53***	524.42	534.38	518.96
	PKLD	122.92*	125.09	128.53	**56.22***	62.00	70.30	57.15
Sn L	RMSE	128.02	126.66*	127.30	129.36	120.80	**119.06***	119.61
	PKLD	43.56*	43.65	45.57	44.31	39.26	38.30*	**32.39**
Ti K	RMSE	132.58	106.38	101.77*	120.23	92.64	**83.43***	158.87
	PKLD	72.52	71.63*	73.86	61.93	60.78*	61.26	**34.82**

Table 4.1. Table of foreground pixel errors for single map optimization *with* denoising prior to generating the mask *and* after the second scan. Each row denotes the elemental map to optimize and its corresponding errors. Amongst each element, asterisks denote the best performing mask for a given time length. Bold values denote the best performing mask across all time lengths.

All Pixel Errors with Denoising after each Scan								
		2 + 8 ms/px Mask			5 + 5 ms/px Mask			10 ms/px
		PKLD	Uniform	MSE	PKLD	Uniform	MSE	Uniform
Au L	RMSE	162.60	161.22	**141.12***	191.21	172.99	159.24*	181.78
	PKLD	43.93	43.79	43.28*	51.07	49.95	49.50*	**42.72**
Ca K	RMSE	143.74	141.43	140.43*	118.76	112.07	**111.26***	115.16
	PKLD	44.24*	44.36	45.05	29.75	29.47*	30.95	**23.79**
Cu K	RMSE	914.24	679.49	**619.67***	939.71	750.54	627.52*	825.47
	PKLD	34.39	30.75	29.98*	35.39	29.95	**29.19***	31.68
Fe K	RMSE	280.53	264.56	257.93*	296.35	263.37	**251.00***	284.89
	PKLD	37.79	37.59*	37.98	26.17	25.17*	25.66	**21.33**
Hg L	RMSE	346.90	321.68	**316.11***	390.89	356.64	347.19*	402.17
	PKLD	44.32	**44.25***	46.87	52.51	52.02*	53.38	56.18
Hg M	RMSE	360.13*	360.33	363.95	338.97	336.57*	339.38	**331.26**
	PKLD	72.19*	74.32	76.74	37.61	36.69*	40.84	**28.26**
Mn K	RMSE	109.69	107.05	96.35*	114.18	101.81	**95.60***	106.16
	PKLD	14.62	14.75	14.28*	12.58	11.88*	12.11	**11.56**
Ni K	RMSE	150.70	139.01	**131.22***	157.91	145.29	135.93*	148.61
	PKLD	12.84	12.43	**12.28***	13.77	13.11	12.95*	12.92
Pb L	RMSE	1685.24*	1690.87	1686.73	1591.01*	1620.38	1677.23	**1573.47**
	PKLD	32.83*	33.07	33.07	27.78*	29.09	31.48	**26.30**
Pb M	RMSE	567.18*	573.55	574.95	**504.34***	510.17	519.75	505.05
	PKLD	120.35*	122.54	125.89	**56.85***	62.54	70.54	57.37
Sn L	RMSE	105.53	104.40*	104.62	106.23	99.38	**98.09***	99.64
	PKLD	35.70	35.59*	36.54	35.25	31.56	30.77*	**27.81**
Ti K	RMSE	52.03	45.85	44.21*	48.75	42.56	**40.80***	56.43
	PKLD	23.85	23.74	23.09*	25.00	24.92*	25.13	**12.88**

Table 4.2. Table of all pixel errors for single map optimization *with* denoising prior to generating the mask *and* after the second scan. Each row denotes the elemental map to optimize and its corresponding errors. Amongst each element, asterisks denote the best performing mask for a given time length. Bold values denote the best performing mask across all time lengths.

Foreground Pixel Errors with Denoising after Scan 2 only								
		2 + 8 ms/px Mask			5 + 5 ms/px Mask			10 ms/px
		PKLD	Uniform	MSE	PKLD	Uniform	MSE	Uniform
Au L	RMSE	218.42	174.48	**162.65***	406.40	354.16	279.25*	214.52
	PKLD	48.83	48.06*	48.75	93.90	86.11	80.41*	**41.67**
Ca K	RMSE	198.42	195.74	194.92*	178.33	171.98	171.09*	**148.76**
	PKLD	68.65*	69.22	69.81	60.07	59.54*	60.84	**28.20**
Cu K	RMSE	1240.07	857.44	**713.31***	1242.86	945.05	754.81*	1082.28
	PKLD	36.02	27.67	**25.95***	38.47	29.67	26.80*	35.51
Fe K	RMSE	375.92	339.06	**323.58***	406.19	352.76	325.24*	389.95
	PKLD	24.55	22.83*	23.46	22.15	19.58	**19.14***	22.56
Hg L	RMSE	607.14	523.59	**468.00***	661.91	567.63	504.57*	639.67
	PKLD	54.31	45.86	**41.40***	61.02	53.14	47.19*	64.97
Hg M	RMSE	275.29*	275.62	279.94	223.58	223.20*	224.18	**184.99**
	PKLD	85.94*	87.25	90.75	54.89*	55.54	56.48	**27.42**
Mn K	RMSE	129.82	120.31	**116.92***	155.39	135.54	123.84*	129.42
	PKLD	13.35	13.13*	13.32	11.95	10.19	**9.75***	10.98
Ni K	RMSE	169.43	146.99	**135.76***	191.38	161.08	141.63*	163.48
	PKLD	10.87	9.99	9.60*	11.51	9.66	**8.77***	10.58
Pb L	RMSE	1649.94	1638.62	1636.63*	1618.72	1586.83*	1647.80	**1552.87**
	PKLD	25.55	25.30*	25.39	24.48	23.95*	26.09	**23.03**
Pb M	RMSE	604.47	599.03	597.38*	532.99	530.13	529.48*	**518.96**
	PKLD	135.96	135.72*	137.22	67.92	67.68*	69.25	**57.15**
Sn L	RMSE	141.91	140.17*	142.12	148.51*	149.18	149.61	**119.61**
	PKLD	45.05*	45.48	47.67	57.89*	58.67	60.06	**32.39**
Ti K	RMSE	137.06	115.75	**106.57***	392.67	240.87	166.17*	158.87
	PKLD	83.39	82.99	82.69*	51.73	34.38	**30.31***	34.82

Table 4.3. Table of foreground pixel errors for single map optimization *without* denoising prior to generating the mask but *with* denoising after the second scan. Each row denotes the elemental map to optimize and its corresponding errors. Amongst each element, asterisks denote the best performing mask for a given time length. Bold values denote the best performing mask across all time lengths.

All Pixel Errors with Denoising after Scan 2 only								
		2 + 8 ms/px Mask			5 + 5 ms/px Mask			10 ms/px
		PKLD	Uniform	MSE	PKLD	Uniform	MSE	Uniform
Au L	RMSE	179.54	150.22	**143.15***	180.53	168.76	153.90*	181.78
	PKLD	43.71	**42.54***	42.82	49.79	49.08	48.45*	42.72
Ca K	RMSE	148.71	146.84	146.34*	134.36	130.05	129.40*	**115.16**
	PKLD	46.96*	47.23	47.56	41.62	41.30*	41.80	**23.79**
Cu K	RMSE	941.42	688.32	**602.30***	937.06	736.57	621.10*	825.47
	PKLD	35.42	31.26	31.16*	34.65	29.88	**29.58***	31.68
Fe K	RMSE	286.31	264.73	256.61*	296.58	264.33	**249.16***	284.89
	PKLD	36.94	36.33*	37.02	25.93	25.07*	25.44	**21.33**
Hg L	RMSE	366.85	336.37	**320.88***	397.19	362.78	345.95*	402.17
	PKLD	43.46	**42.71***	43.68	49.36	48.55*	49.73	56.18
Hg M	RMSE	366.91	365.30*	367.77	348.18	344.60	343.07*	**331.26**
	PKLD	80.61*	81.74	84.99	53.50*	54.10	54.90	**28.26**
Mn K	RMSE	106.80	100.11	97.78*	113.31	101.94	**95.80***	106.16
	PKLD	14.42	14.31*	14.55	12.77	12.13*	12.22	**11.56**
Ni K	RMSE	151.08	137.02	**129.92***	156.00	140.50	131.04*	148.61
	PKLD	13.05	12.60	12.36*	13.15	12.43	**12.07***	12.92
Pb L	RMSE	1711.31	1701.20	1700.70*	1655.79	1625.09*	1672.20	**1573.47**
	PKLD	34.12	33.93*	34.07	29.98	29.41*	29.98	**26.30**
Pb M	RMSE	586.77	581.56	580.02*	517.22	514.77	514.28*	**505.05**
	PKLD	132.85	132.68*	134.22	68.91	68.80*	70.31	**57.37**
Sn L	RMSE	118.25	116.74*	118.13	109.28*	109.62	109.58	**99.64**
	PKLD	39.78*	39.97	41.40	38.12*	38.41	38.79	**27.81**
Ti K	RMSE	53.66	48.44	46.50*	61.63	48.86	**44.13***	56.43
	PKLD	28.55*	28.63	28.61	26.94	26.74	26.69*	**12.88**

Table 4.4. Table of all pixel errors for single map optimization *without* denoising prior to generating the mask but *with* denoising after the second scan. Each row denotes the elemental map to optimize and its corresponding errors. Amongst each element, asterisks denote the best performing mask for a given time length. Bold values denote the best performing mask across all time lengths.

Foreground Pixel Errors with Denoising after Scan 1 only								
		2 + 8 ms/px Mask			5 + 5 ms/px Mask			10 ms/px
		PKLD	Uniform	MSE	PKLD	Uniform	MSE	Uniform
Au L	RMSE	428.73	378.51	359.55*	510.68	404.43	**349.15***	428.20
	PKLD	128.61	114.07	**108.09***	172.83	131.42	108.65*	136.55
Ca K	RMSE	304.98	288.61	284.33*	325.59	263.68	**248.66***	277.37
	PKLD	107.16	104.48*	104.84	97.79	85.07	83.86*	**78.88**
Cu K	RMSE	1209.07	1005.01	941.87*	1234.00	996.89	**911.73***	1122.26
	PKLD	72.72	66.86*	67.10	58.90	**50.10***	51.19	56.51
Fe K	RMSE	624.00	547.78	520.91*	657.87	524.16	**483.14***	597.52
	PKLD	104.13	98.30	97.73*	65.14	**55.50***	55.21	56.51
Hg L	RMSE	821.84	649.65	**572.58***	936.20	696.69	594.53*	859.05
	PKLD	99.84	80.32	73.88*	104.56	75.61	**65.07***	104.83
Hg M	RMSE	452.72	423.56	418.90*	457.04	393.76	**373.09***	377.56
	PKLD	125.72	121.85*	123.33	113.24	101.76*	102.82	**84.44**
Mn K	RMSE	246.49	211.58	201.61*	274.11	212.30	**189.98***	230.95
	PKLD	61.04	57.43*	57.92	48.78	39.90	39.31*	**36.39**
Ni K	RMSE	256.14	238.35	231.22*	281.62	231.14	**212.53***	242.11
	PKLD	44.64	42.30	42.06*	37.53	29.72	**27.61***	28.39
Pb L	RMSE	2690.87	2646.71	**2629.81***	2987.48	2780.17	2699.75*	2648.69
	PKLD	66.46	64.95	64.87*	78.92	71.13	70.03*	**64.65**
Pb M	RMSE	1008.63	920.43	908.07*	1003.69	891.38	**839.64***	866.64
	PKLD	244.66	229.91*	242.41	184.54	170.71*	177.14	**148.38**
Sn L	RMSE	291.13	256.48	253.58*	324.42	242.21	**222.55***	271.18
	PKLD	102.10	92.86*	94.26	109.97	89.56	**86.59***	92.08
Ti K	RMSE	189.96	156.47	146.46*	241.75	149.67	**117.98***	221.27
	PKLD	**43.11***	43.14	43.63	50.73	44.74	44.08*	58.34

Table 4.5. Table of foreground pixel errors for single map optimization *with* denoising prior to generating the mask, but *without* denoising after the second scan. Each row denotes the elemental map to optimize and its corresponding errors. Amongst each element, asterisks denote the best performing mask for a given time length. Bold values denote the best performing mask across all time lengths.

All Pixel Errors with Denoising after Scan 1 only								
		2 + 8 ms/px Mask			5 + 5 ms/px Mask			10 ms/px
		PKLD	Uniform	MSE	PKLD	Uniform	MSE	Uniform
Au L	RMSE	358.27	346.93	**343.01***	405.20	399.41	398.38*	389.69
	PKLD	144.15	142.21	**141.25***	157.42	156.43*	156.78	150.19
Ca K	RMSE	225.33	215.25	212.37*	222.26	206.80	**204.29***	216.03
	PKLD	93.38	90.24*	91.72	76.51	73.33*	74.88	**72.41**
Cu K	RMSE	1017.20	912.98	882.57*	987.43	848.18	**817.02***	893.58
	PKLD	114.80	113.13*	114.75	70.56	67.02*	71.12	**56.81**
Fe K	RMSE	470.57	435.21	425.21*	477.25	406.08	**393.10***	443.10
	PKLD	125.59	124.94*	126.21	77.69	74.22*	78.12	**60.68**
Hg L	RMSE	502.25	462.37	**452.51***	530.41	471.56	458.41*	528.54
	PKLD	98.23	96.68*	98.90	86.65	84.23*	85.81	**82.99**
Hg M	RMSE	489.95	476.52	470.26*	499.64	470.81	**462.75***	465.08
	PKLD	102.09	100.88*	101.11	96.68	87.47	87.38*	**84.34**
Mn K	RMSE	203.76	175.19	**168.77***	218.81	184.40	178.05*	191.24
	PKLD	49.44	47.37*	49.49	46.08	42.94*	47.79	**39.89**
Ni K	RMSE	242.11	235.33	234.56*	241.04	230.03*	233.65	**225.11**
	PKLD	53.06	52.96*	54.10	37.87	37.68*	41.45	**33.05**
Pb L	RMSE	2651.60	2636.88	**2620.87***	2836.43	2774.80	2734.71*	2637.82
	PKLD	65.67	65.43	65.06*	72.40	71.44*	71.77	**64.54**
Pb M	RMSE	912.70	859.23	842.96*	933.46	864.86	**838.97***	841.76
	PKLD	181.16	173.49*	181.59	164.97	158.71*	166.71	**147.61**
Sn L	RMSE	251.47	218.82	**206.33***	242.27	220.82*	221.85	224.99
	PKLD	99.84	90.84	**88.85***	101.29	92.07*	93.65	94.42
Ti K	RMSE	100.55	94.82	**94.41***	104.08	97.56	96.99*	96.55
	PKLD	65.88*	65.97	66.38	61.20	60.96*	61.53	**47.73**

Table 4.6. Table of all pixel errors for single map optimization *with* denoising prior to generating the mask, but *without* denoising after the second scan. Each row denotes the elemental map to optimize and its corresponding errors. Amongst each element, asterisks denote the best performing mask for a given time length. Bold values denote the best performing mask across all time lengths.

		Au L	+ Hg L	+ Cu K	+ Fe K	Uniform
Au L	RMSE	**154.63***	178.39	180.39	189.75	214.52
	PKLD	47.12*	50.59	48.85	49.81	**41.67**
Hg L	RMSE	704.24	**504.75***	558.60	586.83	639.67
	PKLD	70.59	**47.29***	55.60	59.81	64.97
Cu K	RMSE	1448.58	1436.60	**926.46***	945.65	1082.28
	PKLD	51.27	50.67	**32.06***	32.90	35.51
Fe K	RMSE	438.15	427.97	399.09	**362.72***	389.95
	PKLD	29.28	26.79	24.05	**21.62***	22.56

Table 4.7. Foreground errors for the masks in Fig. 4.6. Each column represents a mask. Asterisks denote the best performing adaptive mask, and bold values denote the best performing mask including the regular uniform scan.

CHAPTER 5

Adaptive Macro X-Ray Fluorescence Scanning via Predicted Image Volumes

Abstract

The time to acquire a macro x-ray fluorescence image of paintings is often a timely process on the order of hours or even days. This is sometimes intractable for spectroscopists and other parties with interest in the painting. Image processing algorithms exist that can address these concerns, but are rarely, if ever, widely implemented by spectroscopists. We propose a novel adaptive sampling technique that is easy to implement, and allows for broad customization depending on the goals of the scan. By quickly and iteratively collecting XRF data, higher quality data can be recorded with or without the use of denoising algorithms.

5.1. Introduction

In the past decade, cultural heritage research has relied more and more on x-ray fluorescence (XRF) imaging. Of all the historical artifacts and objects, paintings in particular are the focus of many an XRF-related publication since they are (relatively) flat objects easy to image. This imaging modality is powerful in its ability to find the distribution of chemical elements throughout the paint layers of a painting. Other analytical methods involve extracting paint samples from the painting itself—clearly an undesirable approach

since these paintings are unique and often hold significant cultural and even monetary value. XRF imaging captures the elemental distributions across the entire painting (not just a small sample) without damaging it. These data include the response from paint layers below the surface as X-rays can penetrate the painting further than visible light. The distributions are then analyzed according to the goal of the XRF scan. For example, XRF has been used for painting conservation [7] or authentication [6].

Modern XRF systems are only able to record data one pixel at a time, often requiring many hours or even days to obtain a full sample [9, 10, 11, 12]. Many factors determine the total scan time, such as the painting size, the XRF spot and step sizes, and dwell time at each pixel. Fig. 5.1 shows a color image of the painting *Raising of the Cross* by Bernardo Biti; it was originally scanned with a step size of 0.5 mm and a dwell time of 100 ms/px. With these settings, over 6.5 hours of was required for a painting not much larger than a standard A4- or Letter-sized paper.

Recent software and algorithmic advancements quicken the collection process by employing signal processing techniques [54] regardless of the XRF hardware. Subsampling and inpainting for example allows for step size increases or irregular sampling patterns [35, 36, 31, 37]. Other efforts optimize the dwell time at each pixel outside of a binary sample/skip approach to reduce the scan time [39, 40, 30].

In this chapter, we propose an extension of the work by us. Our work uses a two-stage sampling procedure that strictly complies with time requirements often imposed by outside owners of the painting. A first fast uniform scan collects initial XRF data. Then, depending on the chemical elements of interest, a second adaptive mask is designed to better sample areas of interest using the remaining time. Our work aims at replacing the

initial uniform mask with a mask that predicts where to sample to further reduce the error.

Instead of a uniform mask to begin sampling, we propose sampling a small subset of the pixels strategically selected to represent a painter's palette. Then, we "repaint" the painting by mixing the colors in the palette (and therefore mix the XRF spectra). We then extract the elemental distribution maps from the predicted XRF volume. These predicted maps determine the first full sampling mask.

We recognize that there are issues with our approach, namely that the color image only captures surface level information; hidden paintings can easily make the XRF estimation quite poor. Other fast imaging modalities outside of RGB that can capture sub-surface data may provide beneficial supplementary information. However, this chapter is meant to be exploratory in whether it is possible to use a few samples to design an initial sampling pattern. We will show that it is, in fact, possible, with some reliance on prior knowledge in pigment science, art history, and related fields.

5.2. Related Work

The primary contribution of this work is the use of a predicted XRF volume to inform a real XRF scan. We first discuss XRF prediction when no or a minimal amount of XRF samples are available. Little research on this front has been conducted since underpaintings are known to exist in some historical paintings [**1**, **65**, **66**, **4**]. These subsurface layers can contain not only different pigments, but different structures altogether that depict entirely different scenes. This makes it challenging, if not near impossible, to estimate the XRF signals using the color image alone should an underpainting exist. If there is no

underpainting, we examine if we can harness any of the predicted XRF data for better acquiring actual XRF data.

We also address some existing XRF sampling algorithms that researchers proposed to shorten the total scan time. These publications are more abundant than ones on XRF prediction, and each have their own strengths and drawbacks that we enumerate.

5.2.1. XRF Prediction

In XRF applications, there are two stages: acquiring the volume, and analyzing the volume. Oftentimes the goal of XRF analysis oftentimes is identifying the pigments used and where they exist in the painting. This has been studied extensively across many different paintings. Recent developments use learning-based methods to automatically identify these pigments [**55**, **67**, **68**]. Our problem here is roughly the inverse where we need to generate XRF data from color data.

Some work has been proposed for predicting XRF data. Martin-Ramos and Chiari for example provide a simple tool to predict the XRF data using any combination of appropriate imaging modalities (*e.g.* RGB, X-ray diffraction, etc.) [**69**]. They assume registered, full volume data are available for the chosen modalities except XRF. The XRF data in their experiments are sparse with tens to low hundreds of samples, from which their pigments are found. A clustering algorithm assigns pigments to the unsampled pixels via a straightforward least-squares minimization of all the imaging data as well as the location.

This algorithm inpaints pigment data, but it is mainly applicable when other sub-surface image data is available. A recent proposal by Bombini [**70**] attempts to recreate

the XRF signal using a color image and a pigment database. Bombini suggests using a color similarity score between all the entries in the pigment database and each pixel of the color image. For each pixel, these scores become the weights for mixing the known XRF signals in the pigment database via a Monte Carlo method. Once all the mixtures at each point are found, the XRF volume can be analyzed as if it were true data.

We want a merge between the two options whereby we are able to mix the XRF signals given a few samples from the painting. This would provide a best-of-both-worlds approach where the XRF data is not entirely hallucinated, and it allows for a mixture of signals. We note that no external information outside of the RGB image is required for our proposed method.

5.2.2. XRF Subsampling

There are many factors in choosing the direction for XRF subsampling algorithms. As was mentioned, there are generally two classes of subsampling techniques: binary/discrete dwell times and continuous dwell times. Binary on and off sampling occurs in inpainting [36] and super-resolution [35, 71]. Here, the unsampled locations are inferred using image processing techniques from the existing data. Typically these algorithms are based in dictionary learning. These techniques may not be ideal for XRF analysis since the predicted pixels have not been sampled. In certain applications such as conservation, there should be some samples at every point to ensure that the predicted data has some grounding in reality. This provides extra insurance of sorts to ensure the elemental data is as we would expect.

Other algorithms such as the one proposed by Betterton *et al.* [40] sample the entire painting at different discrete time steps. A first unfocused pass collects blurry data, but at a quick speed. Their reinforcement learning algorithm then determines where to sample for the next pass using progressively more focused apertures. This is a good algorithm, although strict time constraints cannot reliably be met, and some paintings have enough detail where the entire painting could be the region of interest.

We choose to extend the work by us because it (1) can follow strict time requirements, (2) sample every location at the desired resolution, and (3) is fast, modular, and easily customizable for any painting. We designed a sampling mask based on data collected from a fast scan, then denoise the accumulated data. Instead of a standard scan to initialize our algorithm, we want to use the predicted XRF data for a more informed first scan.

5.3. Formulation

We begin the formulation by outlining our strategy. First, we want to capture the XRF emissions of a select few pixels. This set of pixels should try to capture the paint palette that the painter used as best as possible. Next, we try a paint mixing strategy: using the paint palette, we recreate the color image using those pixels alone. The amount of mixing for each paint should be correlated with an equivalent mixture in the XRF domain.

5.3.1. Pixel Selection

In order to establish a paint palette, we need to select representative pixels in the color image, $\mathbf{I}^{\text{RGB}} \in [0,1]^{3\times H\times W}$ where H is the image height and W is the image width. There

Figure 5.1. Bernardo Biti, *Raising of the Cross*, $31 \times 23\,\mathrm{cm}^2$, The Thoma Foundation, inv. no. 2017.72, oil on copper.

are a multitude of ways to choose representative pixels, primarily via clustering algorithms or matrix factorization approaches. Ideally, the selection algorithm we use has two special properties. First, the cluster centers (or matrix endmembers) are themselves data points within the painting. This is here in order to make the selection process less ambiguous; if, for example, a center is not represented in the data, which location should we sample? We aim to avoid this ambiguity with our algorithm selection. Further, the selected points should be well-suited for nonnegative mixing. Because we are mixing paints, it does not make physical sense to have negative amounts of mixing given the palette. This suggests

that we pick the colors most unlike every other color so that we span the color space as best as possible.

With these characteristics in mind, we turn to simplex volume maximization (SiVM) [72], which satisfies both of our criteria. SiVM's goal is to select the best P data points that maximize the volume of the convex hull. It has been applied in XRF data analysis by Alfeld *et al.* [28]. This is an ideal algorithm for our purposes since the selected data exist in the color image, and it selects the extreme points of the dataset, which can be used for better image reconstruction. We refer the reader to the work by Thurau *et al.* for the full details of their SiVM algorithm.

The RGB data may work well, but we look to other color spaces. Instead of using the RGB data, we opt for the CIELAB color space to mimic how people perceive colors. Small differences in color in the RGB space may be larger in the CIELAB space and vice-versa. This should be taken into account since different perceptual colors are more likely to be different paints (and therefore have different XRF responses). If the paints were not very perceptually different, it is more understandable that the paints arose from a mixture. Our palette selection represented in RGB space, $D^{\text{RGB}} \in [0,1]^{3\times P}$, is

$$D^{\text{RGB}} = \text{RGB}\left(\text{SiVM}\left(\text{CIELAB}\left(\mathbf{I}^{\text{RGB}}\right)\right)\right) \tag{5.1}$$

where $\text{RGB}(\cdot)$ converts the CIELAB data to RGB. The CIELAB image is found through a conversion $\text{CIELAB}(\cdot)$ from the RGB image.

5.3.2. Mixture Model: Color

With a palette selected, we now want to "repaint" the painting. In order to do this, it is important to remember some elementary color theory. Namely, we are working with paints, which undergo *subtractive* color mixing. Mixing a red paint with yellow paint produces an orange paint, for example. This is different in RGB space: red light and green light for example produce yellow light. Thus, we need to once again change the color representation from the additive RGB to a subtractive color space. CMY is that subtractive color space, often used in color printing. It is complementary to the RGB space:

$$\mathbf{I}^{\mathrm{CMY}}_{i,h,w} = 1 - \mathbf{I}^{\mathrm{RGB}}_{i,h,w} \tag{5.2}$$

$$D^{\mathrm{CMY}}_{i,p} = 1 - D^{\mathrm{RGB}}_{i,p} \tag{5.3}$$

convert the RGB image and dictionary respectively into CMY space. When we mix the paints in CMY, it mixes similarly to paints.

Now, we can repaint the painting. Using our palette as a dictionary, we want to determine the amount of mixing that occurred. This is a nonnegative matrix factorization problem with one of the matrices known. The abundance (*i.e.* mixture) for each pixel is found by

$$\mathbf{A}^{\mathrm{CMY}} = \underset{\mathbf{A}\geq 0}{\arg\min}\ \frac{1}{PHW} \sum_{j\in(i,h,w)} \left(\mathbf{I}^{\mathrm{CMY}}_{j} - \left(D^{\mathrm{CMY}}\mathbf{A}\right)_{j}\right)^{2} \tag{5.4}$$

where $\mathbf{A}^{\mathrm{CMY}} \in \mathbb{R}_+^{P\times H\times W}$ is the abundance volume. The matrix-tensor multiplication is carried out similarly to matrix multiplication as

$$\left(D^{\mathrm{CMY}}\mathbf{A}\right)_{i,h,w} = \sum_{p=1}^{P} D_{i,p}^{\mathrm{CMY}} \cdot \mathbf{A}_{p,h,w}. \tag{5.5}$$

A plethora of solvers exist to solve the nonnegative minimization problem of Eq. (5.4). Once we have the mixture amounts, we can now move into the XRF domain.

5.3.3. Mixture Model: XRF

When the pixels were selected for the color dictionary, we collect the XRF data there. The underlying, unknown XRF rates $\mathbf{\Psi} \in \mathbb{R}_+^{C\times H\times W}$ determine the samples we measure. The set of nonnegative numbers is denoted by $\mathbb{R}_+$, and the channels C denote the number of different counting bins split by photon energy. The measurements are also dictated by the initial time spent sampling, namely $T^0 \in \mathbb{R}_+^{H\times W}$; the longer the sampling time, the more photons arrive. The collection is modeled by a Poisson process that factors in these traits:

$$\mathbf{X}_{c,h,w} = \mathrm{Poiss}\left(T_{h,w}^0 \cdot \mathbf{\Psi}_{c,h,w}\right) \tag{5.6}$$

where $\mathbf{X}$ is the XRF count volume. Of course, we initially only sample P pixels as determined by the CMY dictionary. We represent these XRF data as a dictionary as well: $D^{\mathrm{XRF}} \in \mathbb{R}_+^{C\times P}$ is the XRF dictionary where each column corresponds to the columns of the CMY dictionary. The choice of dwell time for these pixels should be at least as long as a standard scan since there are only a few points selected. Noisy readings hamper the initial mask design.

Our theory is that we can mix the XRF spectra just as we mix the paints. We acknowledge that this poses numerous problems, especially if underlying paintings are present. However, our end goal is not to estimate the XRF volume. Rather, we want to know if the estimation effectively informs an initial scan.

The estimated XRF volume is built from the P pixels of the XRF dictionary and the CMY abundance:

$$\widetilde{\mathbf{X}} = D^{\mathrm{XRF}}\mathbf{A}^{\mathrm{CMY}}. \tag{5.7}$$

The dwell times are also mixed by the CMY abundance. If $D^{\mathrm{Time}} \in \mathbb{R}_+^{1\times P}$ contains the dwell times for the sampled locations, then the mixed dwell times $\widetilde{T}^0 \in \mathbb{R}_+^{H\times W}$ are

$$\widetilde{T}^0 = D^{\mathrm{Time}}\mathbf{A}^{\mathrm{CMY}}. \tag{5.8}$$

The mask design algorithm requires that the count rates be used, so we normalize the predicted counts by time:

$$\widetilde{\mathbf{\Lambda}}_{c,h,w} = \frac{\widetilde{\mathbf{X}}_{c,h,w}}{\widetilde{T}^0_{h,w}}. \tag{5.9}$$

With the predicted rates, we can decompose the XRF volume into the elemental distribution maps. These maps show the relative quantities and locations of each element present in the painting. Because each element emits a characteristic XRF spectrum, we can identify the elements present using a lookup table matching photon energy to an element and its emission line. PyMca [**15**] is an open-source software that factors the XRF volume by a dictionary of elemental responses, $D^{\mathrm{el}} \in \mathbb{R}_+^{C\times M}$, where M is the number

of chemical elements identified as part of the painting. Each column is a Gaussian (or a Gaussian mixture) fitted to the XRF data. The elemental distribution maps are thus found via

$$\widetilde{\mathbf{A}}^{\mathrm{el}} = \underset{\mathbf{A}\geq 0}{\arg\min}\ \frac{1}{N_{\mathbf{A}}} \sum_{i\in(m,h,w)} \left(\widetilde{\mathbf{\Lambda}}_i - \left(D^{\mathrm{el}}\mathbf{A}\right)_i\right)^2, \tag{5.10}$$

which is identical to how $\mathbf{A}^{\mathrm{CMY}}$ is calculated in Eq. (5.5). Each channel of the elemental maps $\widetilde{\mathbf{A}}^{\mathrm{el}} \in \mathbb{R}_{+}^{M\times H\times W}$ contains the distribution for some corresponding elemental emission line. With these distributions, we will see that so long as we are able to predict the high and low count rate areas, we can better design an initial sampling pattern.

5.3.4. Mask Design

The mask design we follow is based on the count rates of the elemental maps that we predict. It also depends on the minimization goal, typically the mean-squared error (MSE) between the underlying rates and the predicted rates. The Poisson Kullback-Leibler divergence (PKLD) is also used to measure the statistical distance between the two rates, although this mask tends to underperform compared to the MSE mask. We will therefore focus on the design of the MSE mask, which is the result of an optimization. Given a total time constraint τ and pixel-wise dwell minima and maxima of $T^{\min}$ and $T^{\max}$ respectively, the optimal scanning pattern is

$$T^0 = \underset{\substack{\sum_{h,w} T_{h,w}\leq\tau \\ T^{\min}_{h,w}\leq T_{h,w}\leq T^{\max}_{h,w}}}{\arg\min}\ \frac{1}{N_{\mathbf{A}}} \sum_{m,h,w} \mathbf{w}_m \frac{\widetilde{\mathbf{A}}^{\mathrm{el}}_{m,h,w}}{T_{h,w}} \tag{5.11}$$

where $\mathbf{w} \in \mathbb{R}_+^M$ weighs the contribution for each elemental map. Not all the elemental maps need to be optimized, so a subset of $\widetilde{\mathbf{A}}^{\text{el}}$ can be selected by setting entries of $\mathbf{w}$ to zero. Eq. (5.11) is solved by allocating time primarily to the pixels where the MSE is expected to decrease the greatest. We do not incorporate the predicted dwell time into the denominator as a prior dwell time since it is just that—predicted, not an actual dwell time. Once the mask is found, we perform an initial XRF scan.

5.4. Experiments

Our goal is to find an initial sampling mask that can outperform a traditional raster scan. These masks will be designed for fast raster scans. Since the XRF predictions are mixed based on surface level information, there should not be a heavy reliance on the predictions. Thus, these masks can be beneficial for dictating the parameters of a first XRF pass. Subsequent scans can be found using the same mask design instead based on the data collected from the first scan.

5.4.1. Elemental Map Predictions

There are plenty of colors present in the painting, indicating a multitude of different pigments and XRF spectra. We found that a palette of $P = 25$ colors chosen via SiVM is sufficient to recreate the painting. The palette and locations of the samples are shown in Fig. 5.2. In fact, the root mean squared error (RMSE) of the reconstruction in Eq. (5.4) is 8.34×10^{-4}.

These 25 locations are then sampled at a rate of 100 ms/px, which is the original sampling rate. After mixing the XRF samples according to Eqs. (5.7)), (5.8), and (5.9),

Figure 5.2. (Left) Palette locations. (Right) Palette colors.

we decompose the count rate volume via Eq. 5.10. We identified 12 different elements present in the volume of interest. Fig. 5.3 shows the predicted elemental distributions on the right as well as the original, ground truth scan on the left for reference.

Clearly, there are discrepancies between the original scan and the predicted scan. Some maps are more informative than others, though. For example, the predicted Hg L map of Fig. 5.3e seems to be the most correlated: the high count rates of the ground truth match the high count rates of the predicted map. There are, however, many overestimated regions in the predicted map. The Sn L map, despite its noisiness, tends to overestimate areas as well, but in this case is able to roughly predict areas without Sn L. The flag in

the top right and the garment in the bottom middle are the clearest examples, but the people also tend to have lower levels of Sn L than the average. Other predicted maps may be visually appealing, but are anti-correlated to the ground truth. The predicted Pb L map in Fig. 5.3i shows relatively low count rates for the people, but the ground truth has them at higher count rates. Without knowing beforehand that there is this negative correlation, it would be unwise to rely on a Pb L-based sampling mask.

We analyze these intermediary results by looking at the adjusted cosine similarity (ACS) between the predicted and ground truth maps. This is formulated as

$$s_m = \frac{\sum_{h,w} \left(\widetilde{\mathbf{A}}^{\text{el}}_{m,h,w} - \bar{\mathbf{A}}^{\text{el}}_{m} \right) \left(\mathbf{B}_{m,h,w} - \bar{\mathbf{B}}_{m} \right)}{\sqrt{\sum_{h,w} \left(\widetilde{\mathbf{A}}^{\text{el}}_{m,h,w} - \bar{\mathbf{A}}^{\text{el}}_{m} \right)^2} \sqrt{\sum_{h,w} \left(\mathbf{B}_{m,h,w} - \bar{\mathbf{B}}_{m} \right)^2}} \tag{5.12}$$

where

$$\bar{\mathbf{A}}^{\text{el}}_{m} = \frac{1}{HW} \sum_{h,w} \widetilde{\mathbf{A}}^{\text{el}}_{m,h,w} \tag{5.13}$$

$$\bar{\mathbf{B}}_{m} = \frac{1}{HW} \sum_{h,w} \mathbf{B}_{m,h,w}. \tag{5.14}$$

The ACS is equivalent to the Pearson correlation coefficient for a sample population. We denote the ground truth elemental maps as $\mathbf{B}$. In our experiments, $\mathbf{B}$ are the elemental maps from an XRF volume originally scanned at 100 ms/px. This metric is beneficial since we primarily care about where the high and low count rates exist within the painting for each elemental map without regard to any scale factor. The ACS is used over the cosine similarity since the vectorized elemental maps will always occupy the orthant

containing nonnegative values. Negative correlations are therefore impossible with the standard cosine similarity metric for our data. The cosine similarity always suggests high correlation, while the ACS is more nuanced in its ability to determine the true correlation. The raw ACS scores for each elemental map are shown in the third column of Table 5.1 along with the correlation type.

Additionally, we note that some of the elemental maps are noisy. These maps are the basis for the optimization, so we apply a denoising algorithm to maximize the ACS score. The results of the denoising are shown in the last column of Table 5.1. Denoising shows improvements for elements with a non-negligible positive correlation. Elements with no correlation (Au L, Hg M, and Ti K) may or may not see improvements in their respective ACS scores; elements with negative correlations (Pb L and Pb M) understandably see lowered ACS scores post-denoising.

These results back up our prior analysis of the Hg L map with a positive correlation and the Pb L map with a negative correlation. We can rely on some other maps as well that have a substantially positive ACS, namely Ca K, Cu K, Fe K, Hg L, Mn K, Ni K, and Sn L. Of course, these scores cannot be known a priori; however there are some indicators pointing towards a positive, negative, or no correlation. For example, mercury-based red pigments are often present in older paintings. With the color image, we can see that where there are reds, there is a high count rate for Hg L in the predicted and ground truth maps. Similar art historical or materials science-based arguments can be applied for other paints as well to determine how plausible the predicted elemental maps align with the true maps.

Map	Correlation	ACS Raw	ACS Denoised
Au L	0	0.0809	0.0899
Ca K	+	0.2623	0.2921
Cu K	+	0.2493	0.2523
Fe K	+	0.4199	0.5263
Hg L	+	0.6333	0.6437
Hg M	0	-0.0408	-0.0283
Mn K	+	0.3191	0.5022
Ni K	+	0.1971	0.2166
Pb L	–	-0.2838	-0.2838*
Pb M	–	-0.2289	-0.2289*
Sn L	+	0.2941	0.4302
Ti K	0	-0.0426	-0.0426*

Table 5.1. ACS scores for each elemental map of interest. The correlation type is the second column. The raw and denoised ACS scores are the last two columns respectively. Asterisked values indicate that the raw map performed best.

5.4.2. Single Shot Mask Design

When we design the mask, there are some practical considerations to make, particularly our reliance on the predicted map. Placing too much reliance on the predictions could result in egregious oversampling or undersampling that may be more detrimental than the standard raster scan since the predictions are not perfectly correlated with the ground truth.

Our first experiment determines how much weight we can place on these predicted maps. We accomplish this by controlling the minimum and maximum dwell times in the mask design. For each element and average dwell time combination, we progressively increase the dwell time range. We choose to have an equal raise in the maximum dwell

time as the decrease in the minimum dwell time, beginning with a standard raster scan. Both the maximum and minimum are adjusted jointly so that the dwell time can be allocated appropriately. Mathematically, for an average dwell time x, we want to find the best dwell mask where we cap the minimum dwell time as $T^{\min} = x - r$ and the maximum dwell time as $T^{\max} = x + r$ for some $r \in [0, x]$. Once the mask is found, we sample accordingly, then evaluate the RMSE error.

Fig. 5.4 illustrates the effect of the dwell time range on the elemental maps. The plots show the percent change in the RMSE when compared to the standard raster scan where all the pixels are sampled for the same amount of time. We first analyze across different average dwell times. Note that the respective curves are roughly identical in shape and in magnitude. This suggests that the optimal range is proportional to the average dwell time regardless of the underlying scan time. The optimal Fe K line for example has a dwell time range that is approximately 80% of the average dwell time.

For the positively correlated elements, we see a pattern in the minima: the higher the correlation, the larger the dwell range that should be utilized. Further, with the (welcomed) exception of the Fe K map, the higher the correlation, the more improvements in the RMSE we can expect. These patterns follow intuition as well: high correlation means high confidence in the areas to sample, thus resulting in reduced errors.

Many of these positively correlated curves have a valley shape with apparent minima. At small dwell ranges typically less than 80% of the average dwell time, the dwell masks oftentimes are binary in nature: either sample for the minimum dwell time or the maximum dwell time. Masks with longer dwell ranges break from the binary nature and contain dwell times on a spectrum. At some point, the plots plateau. Masks along the

plateau are identical, which happens when the dwell range exceeds the minimum and maximum of the unconstrained case. Fig. 5.5 shows the progression of the Fe K mask with increasing dwell times. Note that there are only two dwell times for the first four dwell ranges, but becomes non-binary afterwards. Additionally, we see that at the final mask (Fig. 5.5f), it more closely resembles a uniform sampling mask compared to, say, the mask of Fig. 5.5d even though the dwell range is lesser. This explains why many of these plots converges closer to the uniform mask results at high dwell ranges.

The non-positively correlated maps experience a similar pattern, but inverted. While the positively correlated maps have different dwell times that minimize the error, this is not the case for the other elemental maps. For example, Pb L has an ACS score of -0.28, and TI K has a score of -0.04, but both peak around 80% to 100% of the average dwell time. The maximum error instead occurs where the masks become no longer binary. When this happens, more pixels are sampled at times greater than the minimum dwell time, and less than the maximum dwell time. These characteristics push the mask towards a uniform sampling pattern, which is the preferred mask for non-positively correlated maps. Fig 5.6 shows the Pb L-based dwell maps for different sampling ranges. At a 4 ms dwell range, the dwell mask also has the greatest actual dwell range of 4 ms (between 3 and 7 ms/px). At a 6 ms dwell range, the allowed dwell range is between 2 and 8 ms/px, but the actual dwell range after optimization is between 3.1 and 5.9 ms/px. This smaller range, and that all pixels are sampled above the allowed minimum, results in a smaller RMSE than the 4 ms dwell range mask. The standard deviation of the 4 ms and 6 ms dwell range masks are 1.67 ms and 0.48 ms respectively; the latter indicates the dwell mask is closer to a uniform mask.

The challenge in real experiments is not only predicting which elemental maps are positively correlated with the ground truth, but the strength of the correlation. Opting for small dwell ranges minimizes errors should the map be negatively correlated, but the potential gains may be reduced. Large dwell ranges may see greater gains, but the losses are greater as well. Although, there should be some solace in the fact that ridding the minimum and maximum dwell time constraints can still provide decent results for positively correlated maps—non-positively correlated maps grow in error, but oftentimes not maximally.

5.4.3. Multi Shot Mask Design

We acknowledge that predictive scans of this nature can be quite risky with regard to the available time to scan a painting. While we demonstrated that it is possible to use the predicted elemental map predictions to design more effective sampling patterns than uniform scans, subsequent scans provide another layer of insurance. Instead of using the entirety of the scan time on the predicted mask, we can split our time into two scans: the first scan based on the predicted mask, then a second scan based on data actually collected.

In our experiments, we assume that we have a total of 10 ms/px on average to scan a painting. We test different divisions of time: (1) 2 ms/px for the first scan and 8 ms/px for the second, (2) 5 ms/px for the first scan and 5 ms/px for the second scan, and (3) 10 ms/px for the only scan. There are two options for the first scan: the predicted scan or a uniform scan. For predictive scans, we select the optimal dwell range. The second scan is based on the data collected in the first scan with a minimum dwell time per pixel of 1

Element	Metric	Mask #1	2 + 8 ms/px Scan				5 +5 ms/px Scan				10 ms/px Scan	
			Scan 1	Den. 1	Scan 2	Den. 2	Scan 1	Den. 1	Scan 2	Den. 2	Scan 1	Den.1
Ca K 0.2921	RMSE	Predicted	578.42	254.62	260.70	183.80	376.67	199.15	233.21	151.34	272.49	145.63
		Uniform	590.13	257.62	263.03	185.84	382.57	201.81	243.05	**143.21***	277.37	148.76
	PKLD	Predicted	227.39	83.28	76.26	63.79	123.20	68.33	66.98	37.63	76.72	**27.34**
		Uniform	233.82	83.72	76.46	65.00	126.16	63.82*	67.30	39.65	78.88	28.20
Cu K 0.2523	RMSE	Predicted	2434.56	1858.32	865.05	**748.32**	1524.11	1420.90	896.51	766.37	1090.86	1082.40
		Uniform	2485.23	1914.78	864.25*	749.06	1579.71	1396.85*	910.77	766.23*	1122.26	1082.28*
	PKLD	Predicted	277.44	94.23	48.85	**24.72**	106.75	55.14	48.56	26.06	53.68	34.45
		Uniform	287.40	96.81	48.61*	24.84	112.82	56.01	48.59	26.27	56.51	35.51
Fe K 0.5263	RMSE	Predicted	1196.02	588.41	450.85	323.64	760.80	444.46	454.08	**316.79**	539.53	373.28
		Uniform	1330.87	616.22	452.68	323.57*	845.67	479.49	475.92	325.87	597.52	389.95
	PKLD	Predicted	263.68	70.62	61.36	23.98	98.72	31.74	47.32	**18.53**	47.77	21.38
		Uniform	326.80	73.49	62.31	24.03	117.78	34.28	46.61*	18.93	56.51	22.56
Hg L 0.6437	RMSE	Predicted	1456.07	838.84	514.35	**435.29**	1017.51	712.06	572.26	472.69	757.97	581.07
		Uniform	1636.88	873.83	521.59	435.86	1152.33	762.10	604.76	485.46	859.05	639.67
	PKLD	Predicted	280.22	100.80	50.49	**36.66**	146.21	79.49	59.14	42.38	88.44	56.79
		Uniform	335.87	105.45	51.59	36.82	171.18	86.79	64.90	44.59	104.83	64.97
Mn K 0.5022	RMSE	Predicted	481.88	187.11	187.32	115.29	306.19	147.53	187.48	**112.29**	219.15	125.47
		Uniform	508.56	196.31	187.62	115.08*	323.26	153.04	197.96	113.51	230.95	129.42
	PKLD	Predicted	141.62	26.04	39.48	13.67	67.67	15.63	36.09	10.89	34.96	10.70
		Uniform	155.88	27.16	40.47	13.04*	71.84	15.96	36.93	**10.61***	36.39	10.98
Sn L 0.4302	RMSE	Predicted	570.97	163.55	246.08	128.02	369.12	145.31	224.96	119.37	267.43	**118.32**
		Uniform	580.82	165.70	244.37*	127.30*	374.16	144.73*	256.58	119.06*	271.18	119.61
	PKLD	Predicted	208.45	52.92	85.98	44.75	136.42	52.96	74.38	31.95	89.99	**31.14**
		Uniform	211.85	51.81*	85.96*	45.57	138.97	46.23*	83.80	38.30	92.08	32.39

Table 5.2. Foreground errors across different 10 ms/px average scans. Each mask is found to minimize the error of its respective map alone. Bold values indicate the best performing RMSE and PKLD across all columns. Asterisked values indicate where the uniform Mask #1 outperformed the predicted Mask #1 in any column.

ms. We would like to know how much of a benefit a predictive first scan provides over a uniform sampling pattern.

Our evaluation methodology is slightly adjusted here since we include the PKLD as an evaluation tool. The PKLD is highly sensitive to noise in the background areas, so for each element we determine a threshold to distinguish between the foreground and background signals. We only consider pixels in the foreground to ensure the background

PKLD errors are not dominant. All the masks, however, are designed to minimize the RMSE.

From the single-shot experiments, we know that the positively correlated elements are the only maps that have a chance of decreasing the error. Thus, we test the elements Ca K, Cu K, Fe K, Hg L, Mn K, and Sn L. Although Ni K is positively correlated, it did not show improvements from the uniform scan. For each element, we assigned the dwell range that provided the best RMSE in our single-shot experiments.

Table 5.2 enumerates each experiment at each stage of the process. Each elemental row had masks designed to minimize the RMSE errors of that element alone. We report the results of the raw and denoised data for full clarity in our experiments. The baselines for the experiments are the denoised uniform scans for the full 10 ms/px (last column, uniform rows). Notice that of the twelve optimal RMSE and PKLD errors, ten favor the use of the predicted map over the uniform scan for the first mask. The Sn L map even favored full reliance on the predicted map. Regardless of which full scan to use though, the improvements are often not enough to justify the means of using a predictive mask over a uniform mask. No final RMSE improvement exceeds 3% compared to the uniform scan. The PKLD has greater percent improvements, but on average is still relatively small.

The greatest improvements are seen if we were to just perform a single scan using the predictive maps. This aligns with the single shot experiment, although here we also have denoised results to compare. In general, the raw data after the first scan has the largest differences between the predicted mask and the uniform mask. In terms of the foreground RMSE, the top performer is the Hg L predicted map with an average 11.5% improvement

over the uniform mask across the three first scan times. The Fe K maps were similarly improved by 10.0%, and Mn K by 5.2%. The remaining maps were only improved by less than 3%.

Other benefits can be seen after denoising the first scan, but these gains are not as strong as the ones seen with the raw data. For example, the denoised Hg L map using the predictive mask saw an average improvement of 6.6% compared to the uniform mask. The other elements saw similar cuts in improvement: Fe K has a 5.4% decrease in error, and Mn K has a 3.8% decrease. The Ca K map has a 1.5% improvement, and Cu K and Sn L have sub 1% improvements.

We have two conclusions from these results. First, if a predictive mask is used, it should be the only scan performed; otherwise, the resulting gains may not be enough to justify employing the predictive scan procedure. Second, we recommend applying this technique when the ACS is at least 0.5. For the same reason why we recommend one predicted scan, the potential gains of elemental maps with an ACS less than 0.5 may not be beneficial enough. With other works of art, the primary issue becomes predicting what the ACS score is without knowledge of the ground truth. Yet, as we mentioned earlier, there are art historical and chemical indicators that can aid ACS prediction.

5.4.4. Multiple Map Optimization

In some XRF applications, there may be interest in finding more than one elemental distribution map. Our final experiment analyzes the impact of optimizing multiple maps in the predictive scan. Previously we saw that the Fe K, Hg L, and Mn K maps performed well with their high ACS score. Thus, we will optimize these three elements.

Element	Metric	Mask #1	2 + 8 ms/px Scan				5 + 5 ms/px Scan				10 ms/px Scan	
			Scan 1	Den. 1	Scan 2	Den. 2	Scan 1	Den. 1	Scan 2	Den. 2	Scan 1	Den.1
Fe K 0.5263	RMSE	Predicted	1223.78	593.92	484.89	**344.02**	780.10	465.17	522.29	347.73	551.54	373.30
		Uniform	1330.87	616.22	509.58	344.86	845.67	479.49	526.24	347.78	597.52	389.95
	PKLD	Predicted	281.01	68.34	50.57	21.24	106.03	33.74	54.25	20.58	50.35	21.38
		Uniform	326.80	73.49	51.77	**20.70***	117.78	34.28	55.06	20.61	56.51	22.56
Hg L 0.6437	RMSE	Predicted	1596.07	867.99	719.39	566.12	1104.27	743.92	696.34	**550.77**	821.08	621.87
		Uniform	1636.88	873.83	694.12*	550.82*	1152.33	762.10	722.84	559.14	859.05	639.67
	PKLD	Predicted	327.24	104.98	84.16	56.30	163.68	84.43	82.04	54.69	99.41	62.19
		Uniform	335.87	105.45	80.49*	**54.32***	171.18	86.79	86.50	55.94	104.83	64.97
Mn K 0.5022	RMSE	Predicted	474.00	189.08	200.19	**116.74**	301.76	147.58	204.85	117.78	215.71	124.01
		Uniform	508.56	196.31	201.81	123.15	323.26	153.04	209.84	118.27	230.95	129.42
	PKLD	Predicted	138.09	28.20	36.55	12.70	67.04	16.34	37.74	11.08	35.22	10.96
		Uniform	155.88	27.16*	36.29*	11.95*	71.84	15.96*	38.43	**10.93***	36.39	10.98

Table 5.3. Foreground errors across different 10 ms/px average scans. Each mask is found to minimize the errors of Fe K, Hg L, and Mn K jointly. Bold values indicate the best performing RMSE and PKLD across all columns. Asterisked values indicate where the uniform Mask #1 outperformed the predicted Mask #1 in any column.

We begin by repeating the dwell range experiment in Section 5.4.2, except here we want to find the optimal dwell range for a multiple element mask. With multiple maps, we need a weighting scheme, especially since the magnitude of the maps differ. We first normalize each predicted map by their respective maximum values. Next, we note the similarity in the predicted maps of Fe K and Mn K. They themselves have an ACS of 0.9715, indicating very high agreement not just with where the high and low rates are, but how relatively high and low. Instead of equally weighing all three normalized maps, we choose to equally weigh the predicted Hg L map with an averaged Fe K and Mn K predicted map. This is the basis for the mask optimization.

Fig. 5.7 provides error plots for the joint predictive mask for each of the three elements. Again, the shape of the curves are roughly identical in shape and magnitude across each of the average dwell times. However, other than the optimal dwell range, the curves

are not identical to the single element masks. The Hg L map, for instance, has a lesser improvement compared to its single mask counterpart—it no longer outperforms the other elements when the full dwell range is used. This is to be expected on some level since the dwell time focus is diverted to include multiple maps. There are also many more high count rate areas in the predicted Fe K and Mn K maps compared to the predicted Hg L map, which heavily dilutes the Hg L gains. On the other hand, the Fe K and Mn K map errors are not as affected for the same reason. Fig. 5.8 shows the different masks tested with varying dwell time ranges. Amongst the Fe K and Mn K mask, there are highlights of the Hg L mask such as the red garment in the bottom center, which is not in either the Fe K or Mn K predictions.

With these plots, we found that the greatest average decrease amongst the three maps is when the dwell range is 60% of the average dwell time. This is true across each of the three different average dwell times. Once we found the optimal dwell range, we continue through the same procedure as the multishot approach in Section 5.4.3. The same 10 ms/px average dwell time splits are tested with denoising after each collection of XRF data. Table 5.3 enumerates these results. Note that the predicted mask #1 is identical for all elements unlike in Table 5.2 where the masks were independently found.

After the last denoising step, we arrive at similar conclusions as the multishot optimization. The gains, where they appear, are not significant enough to justify deviating from an initial uniform scan. In this case, we only see the best improvements in terms of the RMSE; the best PKLD is always better with an initial uniform mask.

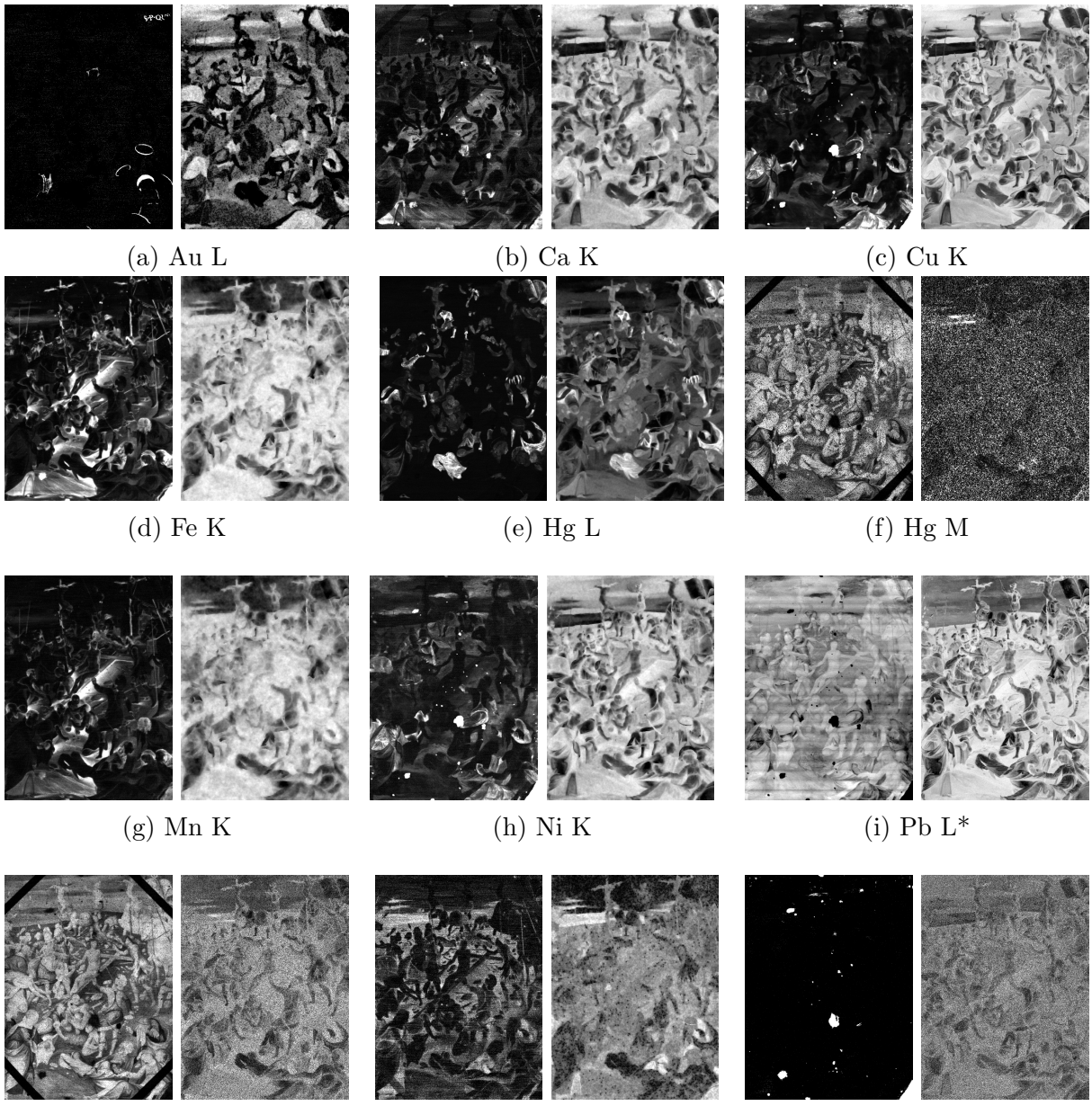
Looking at the first sampling stage, we again see the greatest gains. Here, the Hg L map has the smallest change in percent error with an average 3.7% improvement in the raw

data and 1.9% in the denoised data. The other two maps enjoy greater gains than the Hg L map. For Fe K, the percent decreases are 7.8% and 3.6% for the raw and denoised results respectively. The Mn K map sees 6.7% and 3.8% decreases for the raw and denoised. Since we are optimizing multiple maps, these percent decreases are understandably lower than in the single map cases.

5.5. Conclusion

We proposed an updated method for fast XRF sampling. Using only a few samples, we are able to provide rough estimates of the painting's elemental maps. Some of these estimates, we show, are accurate enough to identify high and low count rate regions. These predictive data dictate how to first scan the painting. Our results show comparable, yet slight improvements of a two-pass system with this predicted data over the raster scan without predicted data. The potential for our algorithm on a first scan is primarily in a first scan since there is a noticeable decrease in error for both the raw data and the denoised data.

Predicting the XRF response is a monumental task, especially with only surface-level information. Yet, there is still enough information that we are able to extract to improve upon fast XRF scans for some elemental maps.



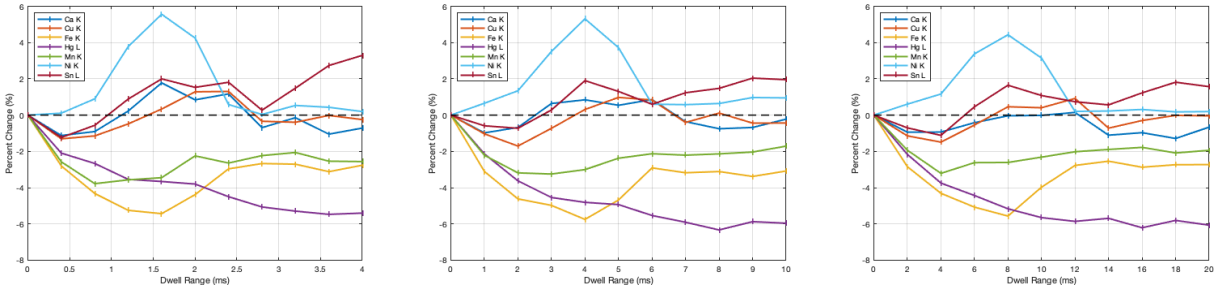
(a) Au L (b) Ca K (c) Cu K

(d) Fe K (e) Hg L (f) Hg M

(g) Mn K (h) Ni K (i) Pb L*

(j) Pb M* (k) Sn L (l) Ti K*

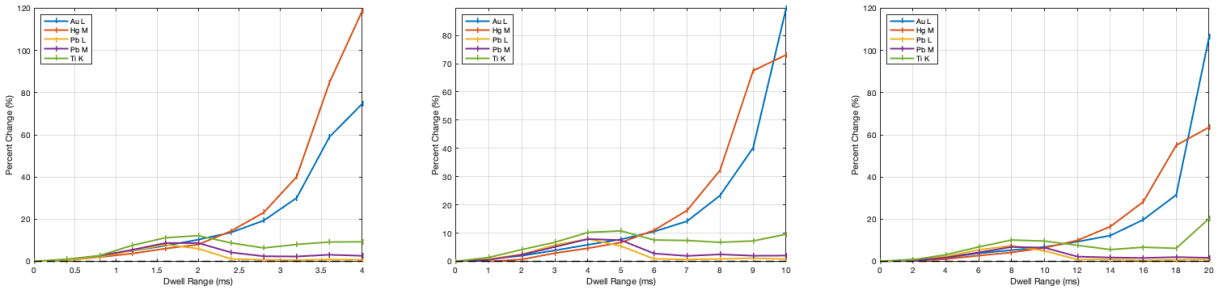
Figure 5.3. Elemental maps of *Raising of the Cross*. These maps are from the original XRF scan of 100 ms/px (left) and a predicted, denoised scan using 25 XRF samples (right). Asterisked elements show the raw predicted scan. The display ranges for each elemental map are adjusted individually to maximize the contrast. We invite the reader to zoom in to better view the maps and the effects of noise.



(a) 2 ms/px avg., Positive correlation

(b) 5 ms/px avg., Positive correlation

(c) 10 ms/px avg., Positive correlation

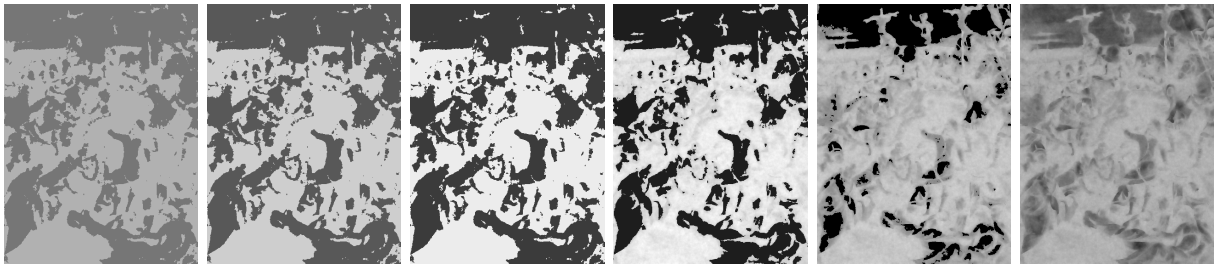


(d) 2 ms/px avg., Nonpositive correlation

(e) 5 ms/px avg., Nonpositive correlation

(f) 10 ms/px avg., Nonpositive correlation

Figure 5.4. Percent change in RMSE errors for each mask as the allowable dwell time range increases. Plots (a)–(c) contain only the elements with high ACS values are plotted. Plots (d)–(f) contain the remaining elements. Negative values indicate improvements from the raster scan.

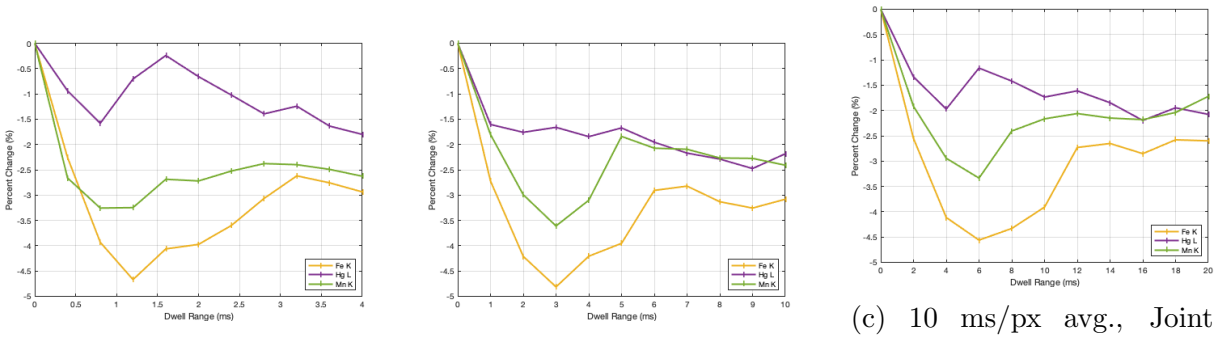


(a) 1 ms (b) 2 ms (c) 3 ms (d) 4 ms (e) 5 ms (f) ≥6 ms

Figure 5.5. Dwell masks based on the predicted Fe K map with an average dwell time of 5 ms/px. The dwell range for each subsequent mask is increased by 1 ms. Masks with dwell ranges of at least 6 ms are all identical. The display range for these masks is between 2.5 and 6.8 ms. The optimal mask is found to be the 4 ms range of Fig. 5.5d.

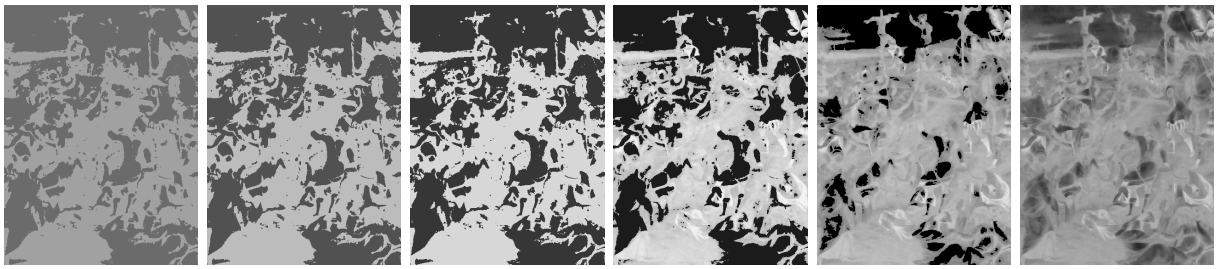
(a) 1 ms (b) 2 ms (c) 3 ms (d) 4 ms (e) 5 ms (f) ≥6 ms

Figure 5.6. Dwell masks based on the predicted Pb L map with an average dwell time of 5 ms/px. The dwell range for each subsequent mask is increased by 1 ms. Masks with dwell ranges of at least 6 ms are all identical. The display range for these masks is between 2.5 and 7.0 ms.



(a) 2 ms/px avg., Joint mask (b) 5 ms/px avg., Joint mask (c) 10 ms/px avg., Joint mask

Figure 5.7. Percent change in RMSE errors for the joint mask as the allowable dwell time range increases. Negative values indicate improvements from the raster scan.



(a) 1 ms (b) 2 ms (c) 3 ms (d) 4 ms (e) 5 ms (f) ≥6 ms

Figure 5.8. Dwell masks based on the predicted joint Fe K, Hg L, and Mn K maps with an average dwell time of 5 ms/px. Masks with dwell ranges of at least 6 ms are all identical. The display range for these masks is between 2.5 and 7.2 ms.

CHAPTER 6

CAVE: A Class of Activation Functions that Constrains the Average, Variance, and Extrema

Abstract

Oftentimes in machine learning and optimization problems, the output must follow certain constraints known a priori. Activation functions at the output of a network are typically used to enforce these constraints. For example, the sigmoid function limits the output range in image and video applications, and the softmax function ensures that the output vector in classification tasks does not violate axioms of probability. While sigmoid constrains the output range, and softmax constrains the output sum to unity, it is difficult to find a function where one can specify the desired range and sum (or equivalently the mean). In this chapter, we present the benefits of using a shifted sigmoid in image processing tasks as the final activation function. We detail how this shift is found, and show its benefits in classification and autoencoding tasks.

6.1. Introduction

In the past decade, deep learning has rapidly risen in popularity as a means to solve a variety of tasks. The foundation of deep learning neural networks is based on learning weights such that a loss function is minimized over a set of known input/output pairs. These weights appear in the compositions of both linear and nonlinear differentiable

(activation) functions that constitute the neural network. Of the activation functions in a network, the final activation is of particular importance in that it is often used to constrain the output to conform with prior knowledge. In color image generating tasks for example, the display range is $[0, 1]$. The sigmoid function is typically used here as a final activation function since it maps an input in $\mathbb{R}^N$ onto the range $(0, 1)^N$. In single label classification tasks, the probability vector at the output should follow a likelihood measure where the elements are all nonnegative and the ℓ_1 norm is unity. The output of a softmax activation function satisfies these restrictions, which explains its ubiquity as a final activation function in classification problems.

The last activation function can be thought of as a constraining function characterizing the neural network's search space. A problem may occur, however, when incorporating all known constraints into the last activation function. In cases where the known constraints cannot all be encompassed in a single activation function, the unaddressed constraints can be relaxed via the method of Lagrangian multipliers. These constraints are instead added into the loss function and weighted by the Lagrangian multiplier. The issue is then choosing the value of the multiplier, not to mention the output of the neural network may not be feasible if the constraints are not allowed to be relaxed.

In this chapter, we introduce the Constrained Average, Variance, and Extrema (CAVE) activation function, whose output has the desired range, mean, and variance specified by the user. It is a type of constrained optimization layer that optimizes just up two to variables in an efficient manner both in terms of speed and memory usage. While this is a very specific scenario, we show that it is more beneficial to use over a softmax function in classification tasks. First, we briefly review some work on constrained optimization using

neural networks. Second, we introduce the CAVE functions and how they are calculated. Finally, we apply it to an image classification problem to show its potential in these tasks.

6.2. Related Work

Constrained optimization, while not a new topic, has recently made advances as it relates to neural networks. We should first point out that most neural networks implicitly incorporate some constraint via the last activation layer. As was mentioned, sigmoid, ReLU, and other such layers constrain the output range to ensure inequality constraints are met. In image classification, equality constraints are implied in the softmax function to ensure the sum of the elements is unity and the values are nonnegative. These functions appear everywhere in neural network architectures as they are generally sufficient for the task at hand.

Aside from image processing, other problems require a different set of constraints to be met. Outputs satisfying ordinary differential equations [73] have been incorporated into neural networks for example. Generalized frameworks to address equality and/or inequality constraints have also been introduced [74, 75, 76, 77], although they are generally computationally heavy. One way that it has been introduced is in bilevel optimization by

$$\begin{aligned} \min_{\Omega} \quad & \mathcal{L}\left(\mathbf{X}, \theta^*; \Omega\right) \\ \text{s.t.} \quad & \theta^* \in \arg\min_{\theta} \; f\left(\mathbf{X}, \theta\right) \end{aligned} \tag{6.1}$$

where $\mathcal{L}$ is a loss function, $\mathbf{X}$ is the input data, θ^* is some optimized parameter with respect to function f, and Ω is the set of weights in the neural network. Bilevel optimization is often thought of as a leader and follower optimization. The leader (upper objective)

optimizes its parameters based on the decision of the follower (lower objective). In a machine learning context, the leader resembles the network itself, and the follower resembles an activation function. The network must learn to adapt to the activation function.

While there are different ways to solve this problem, often these solutions are not differentiable [78, 79], rendering them inaccessible in machine learning environments that rely on gradient information to learn. Gould *et al.* [74, 77] revisited the solution of bilevel optimization problems using gradient descent approaches, the latter citation of which is for neural networks. This is an elegant solution, but requires exact (or near exact) optimization solutions for the gradients to be properly calculated for back propagation. If we can maintain clearly differentiable operations, the network is more likely to learn a better mapping function.

Throughout this chapter, let $\mathbf{X} \in \mathbb{R}^N$ be some input data. Additionally, let $\Theta \subseteq \{l, h, \mu, \nu\}$ be the set of desired output constraints where l denotes the allowable minimum, h denotes the allowable maximum, μ denotes the mean, and ν denotes the variance. By "allowable", we cannot guarantee that the minimum of the output is l, but that it will not be less than l. This is similar to what we mean by the allowable maximum. Let $f(\mathbf{X}, \Theta) : \mathbb{R}^N \to \mathbb{R}^N$ be some activation function such that the output satisfies the constraints set forth by Θ. Mathematically, the set of constraints are

$$f(\mathbf{X}; \Theta)_i \geq l \quad \forall i \tag{6.2}$$

$$f(\mathbf{X}; \Theta)_i \leq h \quad \forall i \tag{6.3}$$

$$\mathbb{E}[f(\mathbf{X}; \Theta)] = \mu \tag{6.4}$$

$$\text{var}(f(\mathbf{X}; \Theta)) = \nu. \tag{6.5}$$

There are plenty of functions that constrain the output in some manner. The softplus function $S^{+}(\cdot)$ has a semi-infinite range in $(0, \infty)$ which can be used to constrain the minimum or maximum, while the sigmoid function $\sigma(\cdot)$ with range $(0, 1)$ constrains both the minimum and maximum. These are seen in rows 1–3 of Table 6.1 by using simple linear transforms to reach the target range. Statistical constraints are shown in rows 4–6, which are well-known linear transforms that can be used to constrain the mean and/or variance for any output. Nothing is particularly noteworthy with these functions.

More interesting functions begin to arise when both range and statistical constraints are required. The next four rows 7–10 are activation functions that overlap range and moment specifications (one each). They are found using a linear transform of the softplus activation to ensure the mean or variance constraints are met. Proofs that the constraints are met for a given Θ are omitted, but can be easily verified. Dai *et al.* proposed using row 7 of Table 6.1 with a sigmoid function instead of a softplus function in the specific case where $l = 0$ to ensure nonnegativity in the output with a constrained mean [**36**]. The maximum for that case, while bounded by the sigmoid initially, is then inadvertently scaled. This is why finite range constraints cannot be combined with statistical constraints using linear transformations alone.

When $\|\Theta\| > 2$, linear transforms applied after the range-limited activation functions cannot satisfy all the constraints. One could use Lagrangian regularization to promote the unmet constraints. For example, if the desired output range is $l = 0$, $h = 1$, and $\mu = 0.5$, one could design a loss function such as

$$\mathcal{L}(\mathbf{X}, \mathbf{Y}) = \|\sigma(\mathbf{X}) - \mathbf{Y}\|_2^2 + \lambda \left(\mathbb{E}\left[\sigma(\mathbf{X})\right] - 0.5\right)^2 \tag{6.6}$$

where $\sigma(\mathbf{X})$ resembles the sigmoidal output of a neural network, $\mathbf{Y}$ is the target output, and λ is a hyperparameter (Lagrangian multiplier). The output range would be valid by nature of the sigmoid, but there is no guarantee that the resulting mean will be μ. Small λ would prioritize minimizing the mean-squared error (MSE), but the result may not be feasible with the error in the mean; high λ would prioritize matching the mean, but would likely result in a high MSE compared to the low λ case. We speculate that the network with high λ would output values very close to the mean with low variance.

CAVE functions are the solution to the remaining constraint combinations. They are beneficial in that (1) they reduce the number of regularizers in the loss function, and (2) the constraints are met, not simply promoted. Using a CAVE function, the loss in Eq. (6.6) would be rewritten as

$$\mathcal{L}(\mathbf{X}, \mathbf{Y}) = \|\mathrm{CAVE}(\mathbf{X}; \Theta) - \mathbf{Y}\|_2^2 \tag{6.7}$$

where $\mathrm{CAVE}(\mathbf{X}; \Theta)$ is defined in rows 11–15 of Table 6.1 based on Θ. In the following section, we delve into how the activation framework works.

6.3. CAVE Functions

In order to satisfy at least three of Constraints (6.2)–(6.5), we propose to apply a linear transform before and after a range-limited activation function. We denote the inner linear transform by scalars a and b, which need to be solved for in order to constrain the mean and/or variance.

Notice that the CAVE functions are the first three functions in Table 6.1 with a linearly transformed input. It's easy to see that the minimum and maximum constraints

Row #	Θ	$f(\mathbf{X};\Theta)$	Min	Max	Mean	Var
1	$\{l\}$	$S^+(\mathbf{X})+l$	l			
2	$\{h\}$	$-S^+(\mathbf{X})+h$		h		
3	$\{l,h\}$	$(h-l)\,\sigma(\mathbf{X})+l$	l	h		
4	$\{\mu\}$	$\mathbf{X}-\mathbb{E}[\mathbf{X}]+\mu$			μ	
5	$\{\nu\}$	$\sqrt{\frac{\nu}{\mathrm{var}(\mathbf{X})}}\mathbf{X}$				ν
6	$\{\mu,\nu\}$	$\sqrt{\frac{\nu}{\mathrm{var}(\mathbf{X})}}(\mathbf{X}-\mathbb{E}[\mathbf{X}])+\mu$			μ	ν
7	$\{l,\mu\}$	$\frac{\mu-l}{\mathbb{E}[S^+(\mathbf{X})]}S^+(\mathbf{X})+l$	l		μ	
8	$\{h,\mu\}$	$\frac{\mu-h}{\mathbb{E}[S^+(\mathbf{X})]}S^+(\mathbf{X})+h$		h	μ	
9	$\{l,\nu\}$	$\sqrt{\frac{\nu}{\mathrm{var}(S^+(\mathbf{X}))}}S^+(\mathbf{X})+l$	l			ν
10	$\{h,\nu\}$	$-\sqrt{\frac{\nu}{\mathrm{var}(S^+(\mathbf{X}))}}S^+(\mathbf{X})+h$		h		ν
11	$\{l,\mu,\nu\}$	$S^+(a\,\mathbf{X}+b)+l$	l		μ	ν
12	$\{h,\mu,\nu\}$	$-S^+(a\,\mathbf{X}+b)+h$		h	μ	ν
13	$\{l,h,\mu\}$	$(h-l)\,\sigma(\mathbf{X}+b)+l$	l	h	μ	
14	$\{l,h,\nu\}$	$(h-l)\,\sigma(a\,\mathbf{X}+b)+l$	l	h		ν
15	$\{l,h,\mu,\nu\}$	$(h-l)\,\sigma(a\,\mathbf{X}+b)+l$	l	h	μ	ν

Table 6.1. List of activation functions constraining all combinations of the minimum, maximum, mean, and variance. Proposed CAVE functions are in rows 11–15. Empty entries indicate values that cannot be constrained.

are satisfied since the inner linear transform does not affect the output of the nonlinear activation.

Here, we will only examine the case where $\Theta=\{l,h,\mu,\nu\}$. The CAVE function from here on is thus

$$\mathrm{CAVE}(\mathbf{X};\Theta)=(h-l)\,\sigma(a\,\mathbf{X}+b)+l. \tag{6.8}$$

In the following subsections, we first show how a and b are found via an optimization. Then, we establish bounds on the mean and variance. Appendix A examines the CAVE

function in greater detail, including a proof that no matter $\mathbf{X}$, μ, or ν, an a and b can be found such that $\mathbb{E}\left[\text{CAVE}\left(\mathbf{X};\Theta\right)\right]=\mu$ and $\text{var}\left(\text{CAVE}\left(\mathbf{X};\Theta\right)\right)=\nu$.

6.3.1. Finding the Linear Transform

Solving for the correct a and b is non-trivial. By separately taking the expected value and variance of Eq. (6.8) and isolating a and b as much as possible, we get two equations:

$$\mathbb{E}\left[\sigma\left(a\,\mathbf{X}+b\right)\right]=\frac{\mu-l}{h-l} \tag{6.9}$$

$$\text{var}\left(\sigma\left(a\,\mathbf{X}+b\right)\right)=\frac{\nu}{\left(h-l\right)^{2}}. \tag{6.10}$$

Variables a and b cannot be solved for analytically, so we turn to numerical approximations. We first define the CAVE loss function $\mathcal{C}$ to minimize the sum of squared errors between the actual and target means and variances:

$$a^{*},b^{*}=\underset{a,b}{\arg\min}\ \mathcal{C} \tag{6.11}$$

$$\mathcal{C}=\mathcal{C}_{\mu}+\mathcal{C}_{\nu} \tag{6.12}$$

$$\mathcal{C}_{\mu}=\left(\mathbb{E}\left[\sigma\left(a\,\mathbf{X}+b\right)\right]-\frac{\mu-l}{h-l}\right)^{2} \tag{6.13}$$

$$\mathcal{C}_{\nu}=\left(\text{var}\left(\sigma\left(a\,\mathbf{X}+b\right)\right)-\frac{\nu}{\left(h-l\right)^{2}}\right)^{2}. \tag{6.14}$$

In the case where the mean or variance alone is specified (rows 13 and 14 of Table 6.1), the CAVE loss is just $\mathcal{C}_{\mu}$ or $\mathcal{C}_{\nu}$.

Since we are optimizing only two scalar variables, we first use gradient descent to approach the minimum, then Newton's second order optimization to quickly and accurately

reach the minimum. The gradient $\nabla_{a,b}\mathcal{C}$ and Hessian $\nabla^2_{a,b}\mathcal{C}$ with respect to a and b can be found analytically. Taking a few gradient descent steps and Newton's method steps finds the appropriate linear transform weights. Gradient descent here is defined as

$$\begin{bmatrix} a \\ b \end{bmatrix}_{k+1} = \begin{bmatrix} a \\ b \end{bmatrix}_{k} - \eta_{\text{gd}} \nabla_{a,b}\mathcal{C} \tag{6.15}$$

and Newton's method is defined as

$$\begin{bmatrix} a \\ b \end{bmatrix}_{k+1} = \begin{bmatrix} a \\ b \end{bmatrix}_{k} - \eta_{\text{nm}} \left(\nabla^2_{a,b}\mathcal{C}\right)^{-1} \nabla_{a,b}\mathcal{C} \tag{6.16}$$

where η_{gd} and η_{nm} are the respective learning rates. Since neural networks learn via the gradient with respect to $\mathbf{X}$, we note that the gradient and Newton steps to optimize a and b are themselves differentiable with respect to $\mathbf{X}$. This allows learning to occur in neural networks and other optimization tasks. Appendix A includes a derivation of these CAVE optimization steps. Appendix B includes a derivation of the gradient of the CAVE optimization steps.

In order to ensure the best possible convergence, we first standardize normalize the input to roughly center the data. We initialize $a_0 = 1$ and $b_0 = 0$, which corresponds to a standard sigmoid activation. We found that learning rates of 1 work well for both gradient descent and Newton's method. The more optimization steps, the more $\mathcal{C}$ is minimized, but not many steps are needed for convergence as will be seen in the experiments. We provide an efficient PyTorch [80] implementation of the CAVE functions.[1]

[1] https://github.com/Henchopp/CAVE

6.3.2. Solution Space of CAVE

To better understand how CAVE operates, we need to study the solution space as defined by Eqs. (6.2))–(6.5). This is necessary because CAVE will optimize Eq. (6.11) regardless of whether the mean and variance are feasible. For example, if we specify $l = 0$, $h = 1$, and $\mu = 2$, there is no possible output where the minimum CAVE error is zero. Here we discuss these bounds on the mean and variance given the output range.

The bounds on the mean are straightforward: the smallest possible mean is l, which occurs when $b \to -\infty$. When $b \to -\infty$, each entry of the output in Eq. (6.8) is l. A similar argument for the maximum mean can be shown, except $b \to \infty$ in this case. Since the CAVE function is smooth with respect to b, we can use the intermediate value theorem to show that

$$l \leq \mu \leq h \tag{6.17}$$

are the only valid options for the mean.

The variance is more difficult to establish bounds on, but we note that Popoviciu's inequality of variance [81] establishes a bound on the variance of random variables as

$$\nu \leq \frac{(h-l)^2}{4} \tag{6.18}$$

which can roughly be followed for sample populations. The maximum variance occurs when half of the output takes on the low l, and the other half is the high value h. This is true only when the variance is constrained, but not the mean. If the mean were constrained, tighter bounds are needed.

Later derivations provided by Bhatia and Davis [82] further constrain the variance when the mean is provided:

$$0 \leq \nu \leq (h - \mu)(\mu - l). \tag{6.19}$$

We see that as the mean approaches either the high or low values, the allowable variance goes to zero. This is in line with our prior analysis of the mean: where the mean is equal one of the extrema, all the output values are identical. Of course, when all values are identical, there is no variance. On the other hand, the maximum possible variance occurs when the mean is an average of the extrema. In this case, the Bhatia-Davis bounds of Eq. (6.19) simplifies to Popovichu's inequality of variance of Eq. (6.18). Like Popovichu's inequality, these bounds are for random variables, but can be reasonably approximated for sample populations. Appendix A provides variance bounds for sample populations.

6.4. Experiments

We perform a small experiment demonstrating where CAVE can be used. In single label classification, the output of the classifier $\hat{\mathbf{Y}}$ is typically a vector of size N for N classes. It has the additional properties that it must follow a probability distribution where $\hat{\mathbf{Y}} \in [0, 1]$ and $\mathbb{E}\left[\hat{\mathbf{Y}}\right] = 1/N$. This is a perfect candidate for CAVE as the final activation function.

Typically in classification tasks, a neural network would use a softmax function as a means to estimate class probabilities at the final output. The softmax function is

$$S^{\max}(\mathbf{X})_i = \frac{e^{\hat{\mathbf{X}}_i}}{\sum_j^N e^{\hat{\mathbf{X}}_j}}. \tag{6.20}$$

Note that the output is constrained: $l = 0$, $h = 1$, and $\mu = \frac{1}{N}$.

We trained two identical shallow convolutional neural networks except for the final activation function: one experiment uses the softmax activation, and the other uses CAVE activation. The dataset we use is CIFAR-100 [83], which provides image data of size 32×32 for classification with 100 different classes. About 500 training images and 100 testing images are included for each class. We therefore set the CAVE constraints with values of $l = 0$, $h = 1$, and $\mu = 1/100$.

The CAVE neural network used no gradient descent steps and seven Newton's method steps to minimize the CAVE loss. The neural network has two convolutional layers each with ReLU activation. This is followed by two linear layers where the first layer has ReLU activation, but the second layer is the final activation. Shallow networks often do not perform as well as deep neural networks, but the shallowness in this case can better highlight the differences in activation functions. Lastly, we use the negative log likelihood loss as the loss function. Table 6.2 shows the results of the two different activation functions.

Metric	Softmax	CAVE
Test Loss	5.546	**3.394**
Top 1 Accuracy	22.9%	**25.5**%
Top 5 Accuracy	47.3%	**50.5**%
Avg. Time/Epoch (s)	**62.07**	62.96
Epochs Trained	25	**7**
Total Time (s)	1551.75	**440.72**

Table 6.2. CIFAR-100 classification results comparing softmax and CAVE activation functions

CAVE enjoys improvements in both accuracy and convergence speed. It provides a three percentage point boost in top 1 and top 5 metrics. While the average time per epoch

is understandably longer due to the CAVE optimization every batch, it adds less than one second per epoch. This is considering seven Newton steps with a batch size of 600 samples where each sample has 100 values for CAVE to optimize individually. We have a relatively fast implementation since we implemented the forward and backward methods of one step of gradient descent and Newton's method. Regardless, CAVE is shown here to converge must faster in terms of the number of epochs.

This is curious in a way since both CAVE and softmax functions evaluate the spread of the data. These functions are both independent of the mean of the input data. We postulate the reason behind the improvements is because CAVE provides more stable gradient information than the softmax. CAVE appears to provide the same spread of gradient information regardless of the spread of the input data. Softmax on the other hand tends to have gradient spreads that are dependent on the input data spread.

For example, we tested the gradients of a random uniform distribution, which was an input to the softmax and CAVE functions. For each entry, we then took the negative log likelihood to see the potential errors for good and poor estimations. We tested this across different scaling factors of the uniform distribution. As the spread of the distribution increased, so too did the variance of the gradients for the softmax function. On the other hand, no matter the scaling factor, CAVE always had the same gradient variance. When the input data spread is small, which is realistically what occurs when neural networks are initialized, the softmax function had an average loss of 4.61 with a variance of 8.08×10^{-6}. CAVE had an average loss of 4.98 with a variance of 9.83×10^{-1}. It's easy to see that the gradient information is much more nuanced with the CAVE activation function than

the softmax. In these low spread regimes, the differential in the maximum and minimum losses is much greater than that of the softmax (and therefore the gradients as well).

6.5. Conclusion

We provided a novel activation function for constraining the output of a neural network. Any user can request a desired minimum, maximum, mean, and variance, and CAVE will be able to provide a differentiable output that satisfies these requirements. It is shown to be fast to converge, despite taking slightly more time per epoch than softmax in a comparison study. Additionally, it is shown to be more accurate as well. Further development and testing of CAVE could include image outputs where a variance is set. In doing so, it may be able to help neural networks mitigate oversmoothing by requiring a variance of the image itself. Of course, other applications exist, and we invite the reader to explore them using CAVE.

References

[1] J. Dik, K. Janssens, G. Van Der Snickt, L. Van Der Loeff, K. Rickers, and M. Cotte, "Visualization of a lost painting by vincent van gogh using synchrotron radiation based x-ray fluorescence elemental mapping," *Analytical chemistry*, vol. 80, no. 16, pp. 6436–6442, 2008.

[2] K. Janssens, G. Vittiglio, I. Deraedt, A. Aerts, B. Vekemans, L. Vincze, F. Wei, I. De Ryck, O. Schalm, F. Adams *et al.*, "Use of microscopic xrf for non-destructive analysis in art and archaeometry," *X-Ray Spectrometry: An International Journal*, vol. 29, no. 1, pp. 73–91, 2000.

[3] C. Calza, M. J. Anjos, M. I. M. Bueno, S. M. de Souza, A. Brancaglion Jr, T. A. Lima, and R. T. Lopes, "Xrf applications in archaeometry: analysis of marajoara pubic covers and pigments from the sarcophagus cartonage of an egyptian mummy," *X-Ray Spectrometry: An International Journal*, vol. 36, no. 5, pp. 348–354, 2007.

[4] E. Pouyet, K. Brummel, S. Webster-Cook, J. Delaney, C. Dejoie, G. Pastorelli, and M. Walton, "New insights into pablo picasso's la miséreuse accroupie (barcelona, 1902) using x-ray fluorescence imaging and reflectance spectroscopies combined with micro-analyses of samples," *SN Applied Sciences*, vol. 2, no. 8, pp. 1–6, 2020.

[5] M. Alfeld, G. Van der Snickt, F. Vanmeert, K. Janssens, J. Dik, K. Appel, L. van der Loeff, M. Chavannes, T. Meedendorp, and E. Hendriks, "Scanning xrf investigation of a flower still life and its underlying composition from the collection of the kröller–müller museum," *Applied Physics A*, vol. 111, no. 1, pp. 165–175, 2013.

[6] S. Saverwyns, C. Currie, and E. Lamas-Delgado, "Macro x-ray fluorescence scanning (ma-xrf) as tool in the authentication of paintings," *Microchemical Journal*, vol. 137, pp. 139–147, 2018.

[7] K. Janssens, G. Van Der Snickt, M. Alfeld, P. Noble, A. van Loon, J. Delaney, D. Conover, J. Zeibel, and J. Dik, "Rembrandt's 'saul and david'(c. 1652): Use of multiple types of smalt evidenced by means of non-destructive imaging," *Microchemical Journal*, vol. 126, pp. 515–523, 2016.

[8] M. Alfeld, K. Janssens, J. Dik, W. de Nolf, and G. van der Snickt, "Optimization of mobile scanning macro-xrf systems for the in situ investigation of historical paintings," *Journal of Analytical Atomic Spectrometry*, vol. 26, no. 5, pp. 899–909, 2011.

[9] M. Alfeld, J. V. Pedroso, M. van Eikema Hommes, G. Van der Snickt, G. Tauber, J. Blaas, M. Haschke, K. Erler, J. Dik, and K. Janssens, "A mobile instrument for in situ scanning macro-xrf investigation of historical paintings," *Journal of Analytical Atomic Spectrometry*, vol. 28, no. 5, pp. 760–767, 2013.

[10] F. P. Romano, C. Caliri, P. Nicotra, S. Di Martino, L. Pappalardo, F. Rizzo, and H. C. Santos, "Real-time elemental imaging of large dimension paintings with a novel mobile macro x-ray fluorescence (ma-xrf) scanning technique," *Journal of Analytical Atomic Spectrometry*, vol. 32, no. 4, pp. 773–781, 2017.

[11] F. Taccetti, L. Castelli, C. Czelusniak, N. Gelli, A. Mazzinghi, L. Palla, C. Ruberto, C. Censori, A. Lo Giudice, A. Re *et al.*, "A multipurpose x-ray fluorescence scanner developed for in situ analysis," *Rendiconti Lincei. Scienze Fisiche e Naturali*, vol. 30, no. 2, pp. 307–322, 2019.

[12] E. Pouyet, N. Barbi, H. Chopp, O. Healy, A. Katsaggelos, S. Moak, R. Mott, M. Vermeulen, and M. Walton, "Development of a highly mobile and versatile large ma-xrf scanner for in situ analyses of painted work of arts," *X-Ray Spectrometry*, vol. 50, no. 4, pp. 263–271, 2021.

[13] K. Janssens, J. Dik, M. Cotte, and J. Susini, "Photon-based techniques for nondestructive subsurface analysis of painted cultural heritage artifacts," *Accounts of chemical research*, vol. 43, no. 6, pp. 814–825, 2010.

[14] P. Brouwer, "Theory of xrf," *Almelo, Netherlands: PANalytical BV*, 2006.

[15] V. Solé, E. Papillon, M. Cotte, P. Walter, and J. Susini, "A multiplatform code for the analysis of energy-dispersive x-ray fluorescence spectra," *Spectrochimica Acta Part B: Atomic Spectroscopy*, vol. 62, no. 1, pp. 63–68, 2007.

[16] M. Alfeld and K. Janssens, "Strategies for processing mega-pixel x-ray fluorescence hyperspectral data: a case study on a version of caravaggio's painting supper at emmaus," *Journal of analytical atomic spectrometry*, vol. 30, no. 3, pp. 777–789, 2015.

[17] R. Browning, "Multi-spectral imaging in materials microanalysis," *Surface and interface analysis*, vol. 20, no. 6, pp. 495–502, 1993.

[18] G. H. Bakır, J. Weston, and B. Schölkopf, "Learning to find pre-images," *Advances in neural information processing systems*, vol. 16, pp. 449–456, 2004.

[19] B. Vekemans, K. Janssens, L. Vincze, A. Aerts, F. Adams, and J. Hertogen, "Automated segmentation of μ-xrf image sets," *X-Ray Spectrometry: An International Journal*, vol. 26, no. 6, pp. 333–346, 1997.

[20] L. Xu, J. Li, Y. Shu, and J. Peng, "Sar image denoising via clustering-based principal component analysis," *IEEE transactions on geoscience and remote sensing*, vol. 52, no. 11, pp. 6858–6869, 2014.

[21] M. Alfeld, M. Wahabzada, C. Bauckhage, K. Kersting, G. Wellenreuther, and G. Falkenberg, "Non-negative factor analysis supporting the interpretation of elemental distribution images acquired by xrf," in *Journal of Physics: Conference Series*, vol. 499, no. 1. IOP Publishing, 2014, p. 012013.

[22] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 11, no. 5, pp. 393–401, 1997.

[23] H. C. Santos, C. Caliri, L. Pappalardo, R. Catalano, A. Orlando, F. Rizzo, and F. P. Romano, "Identification of forgeries in historical enamels by combining the non-destructive scanning xrf imaging and alpha-pixe portable techniques," *Microchemical Journal*, vol. 124, pp. 241–246, 2016.

[24] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer, 2010, vol. 2, no. 1.

[25] R. Mak, M. Lerotic, H. Fleckenstein, S. Vogt, S. M. Wild, S. Leyffer, Y. Sheynkin, and C. Jacobsen, "Non-negative matrix analysis for effective feature extraction in x-ray spectromicroscopy," *Faraday discussions*, vol. 171, pp. 357–371, 2014.

[26] A. De Juan, J. Jaumot, and R. Tauler, "Multivariate curve resolution (mcr). solving the mixture analysis problem," *Analytical Methods*, vol. 6, no. 14, pp. 4964–4976, 2014.

[27] A. Martins, J. Coddington, G. Van der Snickt, B. van Driel, C. McGlinchey, D. Dahlberg, K. Janssens, and J. Dik, "Jackson pollock's number 1a, 1948: A non-invasive study using macro-x-ray fluorescence mapping (ma-xrf) and multivariate curve resolution-alternating least squares (mcr-als) analysis," *Heritage Science*, vol. 4, no. 1, pp. 1–13, 2016.

[28] M. Alfeld, M. Wahabzada, C. Bauckhage, K. Kersting, G. van der Snickt, P. Noble, K. Janssens, G. Wellenreuther, and G. Falkenberg, "Simplex volume maximization (sivm): A matrix factorization algorithm with non-negative constrains and low computing demands for the interpretation of full spectral x-ray fluorescence imaging data," *Microchemical Journal*, vol. 132, pp. 179–184, 2017.

[29] B. Rasti, B. Koirala, P. Scheunders, and P. Ghamisi, "How hyperspectral image unmixing and denoising can boost each other," *Remote Sensing*, vol. 12, no. 11, p. 1728, 2020.

[30] H. Chopp, A. McGeachy, M. Alfeld, O. Cossairt, M. Walton, and A. Katsaggelos, "Denoising fast x-ray fluorescence raster scans of paintings," *arXiv preprint arXiv:2206.01740*, 2022.

[31] J. Yang, Z. Zhang, and Q. Cheng, "Resolution enhancement in micro-xrf using image restoration techniques," *Journal of Analytical Atomic Spectrometry*, vol. 37, no. 4, pp. 750–758, 2022.

[32] M. Bertero, P. Boccacci, and C. De Mol, *Introduction to inverse problems in imaging*. CRC press, 2021.

[33] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[34] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "Deblurring text images via l0-regularized intensity and gradient prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2901–2908.

[35] Q. Dai, E. Pouyet, O. Cossairt, M. Walton, and A. K. Katsaggelos, "Spatial-spectral representation for x-ray fluorescence image super-resolution," *IEEE Transactions on Computational Imaging*, vol. 3, no. 3, pp. 432–444, 2017.

[36] Q. Dai, H. Chopp, E. Pouyet, O. Cossairt, M. Walton, and A. K. Katsaggelos, "Adaptive image sampling using deep learning and its application on x-ray fluorescence image reconstruction," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2564–2578, 2019.

[37] S. Yan, H. Verinaz-Jadan, J.-J. Huang, N. Daly, C. Higgitt, and P. Dragotti, "Super-resolution for macro x-ray fluorescence data collected from old master paintings," 2022.

[38] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical review E*, vol. 69, no. 6, p. 066138, 2004.

[39] G. Kourousias, F. Billè, R. Borghes, A. Alborini, S. Sala, R. Alberti, and A. Gianoncelli, "Compressive sensing for dynamic xrf scanning," *Scientific reports*, vol. 10, no. 1, pp. 1–8, 2020.

[40] J.-R. Betterton, D. Ratner, S. Webb, and M. Kochenderfer, "Reinforcement learning for adaptive illumination with x-rays," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 328–334.

[41] N. De Keyser, G. Van der Snickt, A. Van Loon, S. Legrand, A. Wallert, and K. Janssens, "Jan davidsz. de heem (1606–1684): a technical examination of fruit and flower still lifes combining ma-xrf scanning, cross-section analysis and technical historical sources," *Heritage Science*, vol. 5, no. 1, pp. 1–13, 2017.

[42] A. Martins, C. Albertson, C. McGlinchey, and J. Dik, "Piet mondrian's broadway boogie woogie: non invasive analysis using macro x-ray fluorescence mapping (ma-xrf) and multivariate curve resolution-alternating least square (mcr-als)," *Heritage Science*, vol. 4, no. 1, pp. 1–16, 2016.

[43] S. Kogou, L. Lee, G. Shahtahmassebi, and H. Liang, "A new approach to the interpretation of xrf spectral imaging data using neural networks," *X-Ray Spectrometry*, vol. 50, no. 4, pp. 310–319, 2021.

[44] Q. Dai, H. Chopp, E. Pouyet, O. Cossairt, M. Walton, and A. K. Katsaggelos, "Adaptive image sampling using deep learning and its application on x-ray fluorescence image reconstruction," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2564–2578, 2020.

[45] T. J. Holmes and Y. H. Liu, "Acceleration of maximum-likelihood image restoration for fluorescence microscopy and other noncoherent imagery," *JOSA A*, vol. 8, no. 6, pp. 893–907, 1991.

[46] F. Luisier, C. Vonesch, T. Blu, and M. Unser, "Fast interscale wavelet denoising of poisson-corrupted images," *Signal processing*, vol. 90, no. 2, pp. 415–427, 2010.

[47] R. G. Gallager, *Stochastic processes: theory for applications*. Cambridge University Press, 2013.

[48] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[49] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the royal statistical society: series B (statistical methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[50] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[52] R. R. Coifman and D. L. Donoho, "Translation-invariant de-noising," in *Wavelets and statistics*. Springer, 1995, pp. 125–150.

[53] M. Alfeld, C. Laurenze-Landsberg, A. Denker, K. Janssens, and P. Noble, "Neutron activation autoradiography and scanning macro-xrf of rembrandt van rijn's susanna and the elders (gemäldegalerie berlin): a comparison of two methods for imaging of historical paintings with elemental contrast," *Applied Physics A*, vol. 119, pp. 795–805, 2015.

[54] H. Chopp, A. McGeachy, M. Alfeld, O. Cossairt, M. Walton, and A. Katsaggelos, "Image processing perspectives of x-ray fluorescence data in cultural heritage sciences," *IEEE BITS the Information Theory Magazine*, 2022.

[55] M. Vermeulen, A. McGeachy, B. Xu, H. Chopp, A. Katsaggelos, R. Meyers, M. Alfeld, and M. Walton, "Xrfast a new software package for processing of ma-xrf datasets using machine learning," *Journal of Analytical Atomic Spectrometry*, vol. 37, no. 10, pp. 2130–2143, 2022.

[56] F. J. Anscombe, "The transformation of poisson, binomial and negative-binomial data," *Biometrika*, vol. 35, no. 3/4, pp. 246–254, 1948.

[57] S. K. Bar-Lev and P. Enis, "On the classical choice of variance stabilizing transformations and an application for a poisson variate," *Biometrika*, vol. 75, no. 4, pp. 803–804, 1988.

[58] B. K. Sinha, J. Hartung, and G. Knapp, *Statistical meta-analysis with applications*. John Wiley & Sons, 2011.

[59] F. Li, C. Tang, H. Li, and L. Ge, "A lls operator based s-i wt de-noising algorithm applied in edxrf," *X-Ray Spectrometry*, vol. 52, no. 1, pp. 13–21, 2023.

[60] B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, "Image denoising review: From classical to state-of-the-art approaches," *Information fusion*, vol. 55, pp. 220–244, 2020.

[61] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep learning on image denoising: An overview," *Neural Networks*, vol. 131, pp. 251–275, 2020.

[62] M. Rahman Chowdhury, J. Zhang, J. Qin, and Y. Lou, "Poisson image denoising based on fractional-order total variation," *Inverse Problems & Imaging*, vol. 14, no. 1, 2020.

[63] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2void-learning denoising from single noisy images," pp. 2129–2137, 2019.

[64] M. Alfeld, C. Laurenze-Landsberg, A. Denker, K. Janssens, and P. Noble, "Scanning macro-xrf and neutron activation autoradiography-complimentary methods for elemental imaging of paintings?" *Berliner Beiträge Zur Archäometrie*, vol. 23, pp. 9–14, 2015.

[65] M. Alfeld, W. De Nolf, S. Cagno, K. Appel, D. P. Siddons, A. Kuczewski, K. Janssens, J. Dik, K. Trentelman, M. Walton *et al.*, "Revealing hidden paint layers in oil paintings by means of scanning macro-xrf: a mock-up study based on rembrandt's "an old man in military costume"," *Journal of Analytical Atomic Spectrometry*, vol. 28, no. 1, pp. 40–51, 2013.

[66] G. Van der Snickt, A. Martins, J. Delaney, K. Janssens, J. Zeibel, M. Duffy, C. McGlinchey, B. Van Driel, and J. Dik, "Exploring a hidden painting below the surface of rené magritte's le portrait," *Applied spectroscopy*, vol. 70, no. 1, pp. 57–67, 2016.

[67] B. J. Xu, Y. Wu, P. Hao, M. Vermeulen, A. McGeachy, K. Smith, K. Eremin, G. Rayner, G. Verri, F. Willomitzer *et al.*, "Can deep learning assist automatic identification of layered pigments from xrf data?" *Journal of Analytical Atomic Spectrometry*, vol. 37, no. 12, pp. 2672–2682, 2022.

[68] C. Jones, N. S. Daly, C. Higgitt, and M. R. Rodrigues, "Neural network-based classification of x-ray fluorescence spectra of artists' pigments: an approach leveraging a synthetic dataset created using the fundamental parameters method," *Heritage Science*, vol. 10, no. 1, pp. 1–14, 2022.

[69] J. D. Martin-Ramos and G. Chiari, "Smart_scan: A method to produce composition maps using any elemental, molecular and image data," *Journal of Cultural Heritage*, vol. 39, pp. 260–269, 2019.

[70] A. Bombini, "ganx–generate artificially new xrf a python library to generate ma-xrf raw data out of rgb images," *arXiv preprint arXiv:2304.14078*, 2023.

[71] S. Yan, H. Verinaz-Jadan, J.-J. Huang, N. Daly, C. Higgitt, and P. L. Dragotti, "Super-resolution for macro x-ray fluorescence data collected from old master paintings," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[72] C. Thurau, K. Kersting, M. Wahabzada, and C. Bauckhage, "Descriptive matrix factorization for sustainability adopting the principle of opposites," *Data Mining and Knowledge Discovery*, vol. 24, pp. 325–354, 2012.

[73] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *Advances in neural information processing systems*, vol. 31, 2018.

[74] S. Gould, B. Fernando, A. Cherian, P. Anderson, R. S. Cruz, and E. Guo, "On differentiating parameterized argmin and argmax problems with application to bi-level optimization," *arXiv preprint arXiv:1607.05447*, 2016.

[75] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 136–145.

[76] P. L. Donti, D. Rolnick, and J. Z. Kolter, "Dc3: A learning method for optimization with hard constraints," *arXiv preprint arXiv:2104.12225*, 2021.

[77] S. Gould, R. Hartley, and D. J. Campbell, "Deep declarative networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[78] J. F. Bard, *Practical bilevel optimization: algorithms and applications*. Springer Science & Business Media, 2013, vol. 30.

[79] S. Dempe and S. Franke, "On the solution of convex bilevel optimization problems," *Computational Optimization and Applications*, vol. 63, no. 3, pp. 685–703, 2016.

[80] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[81] T. Popoviciu, "Sur les équations algébriques ayant toutes leurs racines réelles," *Mathematica*, vol. 9, no. 129-145, p. 20, 1935.

[82] R. Bhatia and C. Davis, "A better bound on the variance," *The american mathematical monthly*, vol. 107, no. 4, pp. 353–357, 2000.

[83] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[84] N. Kaiblinger and B. Spangl, "An inequality for the analysis of variance," *Mathematical Inequalities & Applications*, vol. 23, no. 3, pp. 961–9, 2020.

APPENDIX A

CAVE Derivation

Here we derive the CAVE function in full. For completeness, we additionally provide analysis and evaluations showing that the mean and variance can be found exactly provided the mean and variance are within certain bounds.

A.1. CAVE Function & Constraints

CAVE functions guarantee that the following conditions are met so long as the solution space is not empty:

$$\min_i \left(\text{CAVE}\left(\mathbf{X};\, \Theta\right)_i\right) \geq l \tag{A.1}$$

$$\max_i \left(\text{CAVE}\left(\mathbf{X};\, \Theta\right)_i\right) \leq h \tag{A.2}$$

$$\mathbb{E}\left[\text{CAVE}\left(\mathbf{X};\, \Theta\right)\right] = \mu \tag{A.3}$$

$$\text{var}\left(\text{CAVE}\left(\mathbf{X};\, \Theta\right)\right) = \nu \tag{A.4}$$

where l is the lower bound, h is the upper bound, μ is the mean, and ν is the variance of the output, all chosen by the user. Set $\Theta = \{l, h, \mu, \nu\}$ contains the specified output constraints. Input data $\mathbf{X} \in \mathbb{R}^N$ is a real-valued N-dimensional vector (or any arbitrary shape).

In order to enforce the constraints, CAVE proposes a linear transform *prior* to a function f. Function f should have the following properties, the reason for which is discussed in Section A.3:

(1) $f : \mathbb{R} \to \mathbb{R}$, and operates element-wise for multidimensional inputs

(2) f is range limited to $[f_{min}, f_{max}]$ for some $-\infty < f_{min} < f_{max} < \infty$

(3) f is smooth

(4) f is strictly increasing.

CAVE finds a linear transform such that when pre-applied to the input prior to a nonlinear activation, the target mean and variance are matched. Mathematically, the full CAVE function is defined as

$$\text{CAVE}\left(\mathbf{X};\, \Theta\right) = (h - l)\, \frac{f\left(a\, \mathbf{X} + b\right) - f_{min}}{f_{max} - f_{min}} + l \tag{A.5}$$

where variables $a \in \mathbb{R}$ and $b \in \mathbb{R}$ constitute the linear pre-transform. The fractional component reduces the range to $[0, 1]$. This is then linearly transformed to reach the target range.

We can quickly verify that the minimum and maximum Constraints (A.1) and (A.2) are met because of the finite range of f:

$$\min_i \left(\text{CAVE}\left(\mathbf{X};\, \Theta\right)_i\right) \geq l \tag{A.6}$$

$$\max_i \left(\text{CAVE}\left(\mathbf{X};\, \Theta\right)_i\right) \leq h. \tag{A.7}$$

The output mean and variance, however, are dependent on the value of a and b. We need to solve for a and b such that Constraints (A.3) and (A.4) are met. The mean and

variance of the cave function are

$$\mathbb{E}\left[\text{CAVE}\left(\mathbf{X};\,\Theta\right)\right] = (h-l)\,\frac{\mathbb{E}\left[f\left(a\,\mathbf{X}+b\right)\right]-f_{min}}{f_{max}-f_{min}}+l \tag{A.8}$$

$$\text{var}\left(\text{CAVE}\left(\mathbf{X};\,\Theta\right)\right) = \left(\frac{h-l}{f_{max}-f_{min}}\right)^{2}\text{var}\left(f\left(a\,\mathbf{X}+b\right)\right), \tag{A.9}$$

which are dependent on variables a and b. From Constraints (A.3) and (A.4), we want

$$(h-l)\,\frac{\mathbb{E}\left[f\left(a\,\mathbf{X}+b\right)\right]-f_{min}}{f_{max}-f_{min}}+l=\mu \tag{A.10}$$

$$\left(\frac{h-l}{f_{max}-f_{min}}\right)^{2}\text{var}\left(f\left(a\,\mathbf{X}+b\right)\right)=\nu, \tag{A.11}$$

so we massage Eqs. (A.10) and (A.11) to isolate a and b as far as possible:

$$\mathbb{E}\left[f\left(a\,\mathbf{X}+b\right)\right]=\mu' \tag{A.12}$$

$$\text{var}\left(f\left(a\,\mathbf{X}+b\right)\right)=\nu' \tag{A.13}$$

$$\mu' = \frac{f_{max}-f_{min}}{h-l}\left(\mu-l\right)+f_{min} \tag{A.14}$$

$$\nu' = \left(\frac{f_{max}-f_{min}}{h-l}\right)^{2}\nu. \tag{A.15}$$

Variables μ' and ν' are the CAVE-adjusted mean and variance.

While we have two equations and two unknowns, we cannot analytically solve for a and b since function f must strictly be nonlinear. Thus, we turn to numerical approaches for the solution.

A.2. CAVE Optimization

To approximate the solution, we need to develop an optimization framework. The goal of the optimization is to find an a and b such that Constraints (A.3) and (A.4) are met. We do this by defining a loss function that we minimize through gradient descent and Newton's method.

Using gradient descent and Newton's method requires further constraints to the function f. Primarily, since Newton's method is a second order optimization technique, we require that f be a smooth, twice differentiable function.

A.2.1. CAVE Loss Function

In order to find a and b, we introduce a loss function to minimize the error of the mean and variance between the target and output. We define the minimization as

$$a^*, b^* = \arg\min_{a,\,b} \; \mathcal{C}\left(a, b, \mathbf{X}; \mu', \nu'\right) \tag{A.16}$$

where

$$\mathcal{C}\left(a, b, \mathbf{X}; \mu', \nu'\right) = \mathcal{C}_\mu\left(a, b, \mathbf{X}; \mu'\right) + \mathcal{C}_\nu\left(a, b, \mathbf{X}; \nu'\right) \tag{A.17}$$

$$\mathcal{C}_\mu\left(a, b, \mathbf{X}; \mu'\right) = \| E_\mu\left(a, b, \mathbf{X}; \mu'\right) \|_2^2 \tag{A.18}$$

$$\mathcal{C}_\nu\left(a, b, \mathbf{X}; \nu'\right) = \| E_\nu\left(a, b, \mathbf{X}; \nu'\right) \|_2^2 \tag{A.19}$$

$$E_\mu\left(a, b, \mathbf{X}; \mu'\right) = \mathbb{E}\left[f\left(a\,\mathbf{X} + b\right)\right] - \mu' \tag{A.20}$$

$$E_\nu\left(a, b, \mathbf{X}; \nu'\right) = \mathrm{var}\left(f\left(a\,\mathbf{X} + b\right)\right) - \nu'. \tag{A.21}$$

Solutions a^* and b^* are the values that minimize $\mathcal{C}$ and produce a function that satisfy Conditions (A.1)–(A.4).

A.2.2. CAVE Minimization

Since there are only two variables, we can use gradient descent in conjunction with Newton's method of optimization to find a^* and b^*. It is well known that the size of the Hessian is on the order of $\mathcal{O}(n^2)$ for n variables, which often limits its practicality in large scale machine learning and optimization tasks. However, optimizing $n = 2$ variables is second in memory simplicity behind single variable optimization.

Gradient descent is first used to approach the minimum. Newton's method moves the solution towards a minimum if the loss manifold is concave up at the initial a and b, but towards a maximum if the manifold is concave down. It is essential that the initialization be close to the solution, which is why gradient descent is used first. For two variable optimization of weights a and b, gradient descent in its most basic form is defined as

$$\begin{bmatrix} a \\ b \end{bmatrix}_{k+1} = \begin{bmatrix} a \\ b \end{bmatrix}_{k} - \eta \, \mathcal{S}_G \left(\begin{bmatrix} a \\ b \end{bmatrix}_{k} \right) \tag{A.22}$$

$$\mathcal{S}_G \left(\begin{bmatrix} a \\ b \end{bmatrix}_{k} \right) = \nabla_{a,b} \, \mathcal{C} \left(\begin{bmatrix} a \\ b \end{bmatrix}_{k} \right) \tag{A.23}$$

where k is the iteration index, $\mathcal{S}_G(\cdot)$ is the gradient descent step function, η is the learning rate, and $\nabla_{a,b}$ is the gradient operator with respect to a and b. After taking some number of steps, we approach the minimum where the loss manifold is concave up.

Newton's method is then utilized to finish the optimization. The iterative method is defined as

$$\begin{bmatrix} a \\ b \end{bmatrix}_{k+1} = \begin{bmatrix} a \\ b \end{bmatrix}_{k} - \eta \, \mathcal{S}_N \left(\begin{bmatrix} a \\ b \end{bmatrix}_{k} \right) \tag{A.24}$$

$$\mathcal{S}_N \left(\begin{bmatrix} a \\ b \end{bmatrix}_{k} \right) = \left(\nabla^2_{a,b} \mathcal{C} \left(\begin{bmatrix} a \\ b \end{bmatrix}_{k} \right) \right)^{-1} \left(\nabla_{a,b} \mathcal{C} \left(\begin{bmatrix} a \\ b \end{bmatrix}_{k} \right) \right) \tag{A.25}$$

where $\mathcal{S}_N(\cdot)$ is the Newton's method step function, and $\nabla^2_{a,b}$ is the Hessian operator with respect to a and b. We need to find $\mathcal{S}_G$ and $\mathcal{S}_N$ in terms of function f, the furthest decomposition.

In the coming subsections, we use a shorthand notation whereby we drop the arguments to the loss and error functions, which all return scalar values.

A.2.3. Gradient and Hessian of CAVE Loss

We can solve for the gradient analytically by

$$\nabla_{a,b} \mathcal{C} = \begin{bmatrix} \frac{\partial \mathcal{C}}{\partial a} \\ \frac{\partial \mathcal{C}}{\partial b} \end{bmatrix} \tag{A.26}$$

$$= \begin{bmatrix} \frac{\partial \mathcal{C}_\mu}{\partial a} + \frac{\partial \mathcal{C}_\nu}{\partial a} \\ \frac{\partial \mathcal{C}_\mu}{\partial b} + \frac{\partial \mathcal{C}_\nu}{\partial b} \end{bmatrix} \tag{A.27}$$

where

$$\frac{\partial \mathcal{C}_\mu}{\partial a} = 2 E_\mu \frac{\partial E_\mu}{\partial a} \tag{A.28}$$

$$\frac{\partial \mathcal{C}_\nu}{\partial a} = 2E_\nu \frac{\partial E_\nu}{\partial a} \tag{A.29}$$

$$\frac{\partial \mathcal{C}_\mu}{\partial b} = 2E_\mu \frac{\partial E_\mu}{\partial b} \tag{A.30}$$

$$\frac{\partial \mathcal{C}_\nu}{\partial b} = 2E_\nu \frac{\partial E_\nu}{\partial b}. \tag{A.31}$$

The Hessian is found by taking the second derivative of $\mathcal{C}$:

$$\nabla^2_{a,b}\mathcal{C} = \begin{bmatrix} \frac{\partial^2 \mathcal{C}}{\partial a^2} & \frac{\partial^2 \mathcal{C}}{\partial a \partial b} \\ \frac{\partial^2 \mathcal{C}}{\partial a \partial b} & \frac{\partial^2 \mathcal{C}}{\partial b^2} \end{bmatrix} \tag{A.32}$$

$$= \begin{bmatrix} \frac{\partial^2 \mathcal{C}_\mu}{\partial a^2} + \frac{\partial^2 \mathcal{C}_\nu}{\partial a^2} & \frac{\partial^2 \mathcal{C}_\mu}{\partial a \partial b} + \frac{\partial^2 \mathcal{C}_\nu}{\partial a \partial b} \\ \frac{\partial^2 \mathcal{C}_\mu}{\partial a \partial b} + \frac{\partial^2 \mathcal{C}_\nu}{\partial a \partial b} & \frac{\partial^2 \mathcal{C}_\mu}{\partial b^2} + \frac{\partial^2 \mathcal{C}_\nu}{\partial b^2} \end{bmatrix} \tag{A.33}$$

where

$$\frac{\partial^2 \mathcal{C}_\mu}{\partial a^2} = 2\left(\left(\frac{\partial E_\mu}{\partial a}\right)^2 + E_\mu \frac{\partial^2 E_\mu}{\partial a^2}\right) \tag{A.34}$$

$$\frac{\partial^2 \mathcal{C}_\nu}{\partial a^2} = 2\left(\left(\frac{\partial E_\nu}{\partial a}\right)^2 + E_\nu \frac{\partial^2 E_\nu}{\partial a^2}\right) \tag{A.35}$$

$$\frac{\partial^2 \mathcal{C}_\mu}{\partial a \partial b} = 2\left(\frac{\partial E_\mu}{\partial a}\frac{\partial E_\mu}{\partial b} + E_\mu \frac{\partial^2 E_\mu}{\partial a \partial b}\right) \tag{A.36}$$

$$\frac{\partial^2 \mathcal{C}_\nu}{\partial a \partial b} = 2\left(\frac{\partial E_\nu}{\partial a}\frac{\partial E_\nu}{\partial b} + E_\nu \frac{\partial^2 E_\nu}{\partial a \partial b}\right) \tag{A.37}$$

$$\frac{\partial^2 \mathcal{C}_\mu}{\partial b^2} = 2\left(\left(\frac{\partial E_\mu}{\partial b}\right)^2 + E_\mu \frac{\partial^2 E_\mu}{\partial b^2}\right) \tag{A.38}$$

$$\frac{\partial^2 \mathcal{C}_\nu}{\partial b^2} = 2\left(\left(\frac{\partial E_\nu}{\partial b}\right)^2 + E_\nu \frac{\partial^2 E_\nu}{\partial b^2}\right). \tag{A.39}$$

The losses are now all defined in terms of the mean error function E_μ and variance error function E_ν.

A.2.4. Gradient and Hessian of CAVE Errors

We use an identical shorthand here for function $f(\cdot)$. Explicitly, this shorthand is

$$f \leftarrow f(a\,\mathbf{X} + b) \tag{A.40}$$

so f and its derivatives with respect to a and b represent a multidimensional output of the same size as $\mathbf{X}$. All multiplications in this appendix we also assume are point-wise Hadamard products.

The mean error function and its derivatives are found as

$$E_\mu = \mathbb{E}[f] - \mu' \tag{A.41}$$

$$\frac{\partial E_\mu}{\partial a} = \mathbb{E}\left[\frac{\partial f}{\partial a}\right] \tag{A.42}$$

$$\frac{\partial E_\mu}{\partial b} = \mathbb{E}\left[\frac{\partial f}{\partial b}\right] \tag{A.43}$$

$$\frac{\partial^2 E_\mu}{\partial a^2} = \mathbb{E}\left[\frac{\partial^2 f}{\partial a^2}\right] \tag{A.44}$$

$$\frac{\partial^2 E_\mu}{\partial a \partial b} = \mathbb{E}\left[\frac{\partial^2 f}{\partial a \partial b}\right] \tag{A.45}$$

$$\frac{\partial^2 E_\mu}{\partial b^2} = \mathbb{E}\left[\frac{\partial^2 f}{\partial b^2}\right]. \tag{A.46}$$

The variance error function and its derivatives are also found as

$$E_\nu = \mathbb{E}\left[f^2\right] - \mathbb{E}^2[f] - \nu' \tag{A.47}$$

$$\frac{\partial E_\nu}{\partial a} = 2\left(\mathbb{E}\left[f\frac{\partial f}{\partial a}\right] - \mathbb{E}\left[f\right]\frac{\partial E_\mu}{\partial a}\right) \tag{A.48}$$

$$\frac{\partial E_\nu}{\partial b} = 2\left(\mathbb{E}\left[f\frac{\partial f}{\partial b}\right] - \mathbb{E}\left[f\right]\frac{\partial E_\mu}{\partial b}\right) \tag{A.49}$$

$$\frac{\partial^2 E_\nu}{\partial a^2} = 2\left(\mathbb{E}\left[\left(\frac{\partial f}{\partial a}\right)^2 + f\frac{\partial^2 f}{\partial a^2}\right] - \left(\frac{\partial E_\mu}{\partial a}\right)^2 - \mathbb{E}\left[f\right]\frac{\partial^2 E_\mu}{\partial a^2}\right) \tag{A.50}$$

$$\frac{\partial^2 E_\nu}{\partial a \partial b} = 2\left(\mathbb{E}\left[\frac{\partial f}{\partial a}\frac{\partial f}{\partial b} + f\frac{\partial^2 f}{\partial a \partial b}\right] - \frac{\partial E_\mu}{\partial a}\frac{\partial E_\mu}{\partial b} - \mathbb{E}\left[f\right]\frac{\partial^2 E_\mu}{\partial a \partial b}\right) \tag{A.51}$$

$$\frac{\partial^2 E_\nu}{\partial b^2} = 2\left(\mathbb{E}\left[\left(\frac{\partial f}{\partial b}\right)^2 + f\frac{\partial^2 f}{\partial b^2}\right] - \left(\frac{\partial E_\mu}{\partial b}\right)^2 - \mathbb{E}\left[f\right]\frac{\partial^2 E_\mu}{\partial b^2}\right). \tag{A.52}$$

The error functions are now all defined in terms of the range-limited function f.

A.2.5. Gradient and Hessian of CAVE Base Function

Lastly, we have the derivatives of the base function f:

$$\frac{\partial f}{\partial a} = \frac{\partial}{\partial a}\left[f\left(a\,\mathbf{X} + b\right)\right] = f'\left(a\,\mathbf{X} + b\right)\mathbf{X} \tag{A.53}$$

$$\frac{\partial f}{\partial b} = \frac{\partial}{\partial b}\left[f\left(a\,\mathbf{X} + b\right)\right] = f'\left(a\,\mathbf{X} + b\right) \tag{A.54}$$

$$\frac{\partial^2 f}{\partial a^2} = \frac{\partial^2}{\partial a^2}\left[f\left(a\,\mathbf{X} + b\right)\right] = f''\left(a\,\mathbf{X} + b\right)\mathbf{X}^2 \tag{A.55}$$

$$\frac{\partial^2 f}{\partial a \partial b} = \frac{\partial^2}{\partial a \partial b}\left[f\left(a\,\mathbf{X} + b\right)\right] = f''\left(a\,\mathbf{X} + b\right)\mathbf{X} \tag{A.56}$$

$$\frac{\partial^2 f}{\partial b^2} = \frac{\partial^2}{\partial b^2}\left[f\left(a\,\mathbf{X} + b\right)\right] = f''\left(a\,\mathbf{X} + b\right) \tag{A.57}$$

where f' and f'' denote the first and second derivatives of the base function. Recall that function f operates element-wise, and so too do its derivatives. The base function f is chosen by the user, so we conclude the derivation here. All the user would need to do is provide the function f as well as its first and second derivatives to use CAVE.

A.2.6. Single Variable Optimization Modes

We provided a solution to minimizing the loss of the mean and variance for CAVE functions. While the derivation was for joint minimization, one does not always need to specify both the mean and variance (assuming the minimum and maximum values are specified). In these cases, we only specify the mean or the variance, but not both. If only the mean is specified, the overall loss function is $\mathcal{C}_\mu(a, b, \mathbf{X}; \mu')$; if only the variance is specified, the overall loss function is $\mathcal{C}_\nu(a, b, \mathbf{X}; \nu')$.

A.3. CAVE Feasibility Space

We have a framework for minimizing the mean and variance losses via Eq. (A.16), but there is no guarantee that the minimum loss is necessarily 0. For example, one can specify $l = 0$, $h = 1$, and $\mu = 2$, but it is clearly impossible that $\mu > h$. The optimization can still be carried out to reach a minimum, but the minimum will certainly be positive. Bounds on the mean and variance need to be established to prevent cases with positive minima.

Since the loss is a sum of quadratics as shown in Eqs. (A.17)–(A.19), we have that

$$\min_{a,\,b} \mathcal{C}(a, b, \mathbf{X}; \mu', \nu') \geq 0. \tag{A.58}$$

When the optimal minimum is strictly positive, it indicates that the CAVE constraints are *not* met despite optimizing the loss. We need to find a feasible region of μ and ν where

$$\min_{a,\,b} \mathcal{C}(a, b, \mathbf{X}; \mu', \nu') = 0, \tag{A.59}$$

noting the strict equality.

We will discuss three different constraints for the three different CAVE optimization modes: (1) mean optimization only, (2) variance optimization only, and (3) mean and variance joint optimization.

There is also a question concerning Eqs. (A.12)–(A.15): given any mean and variance satisfying the bounds constraints, does there exist an a and b that satisfies CAVE Constraints (A.1)–(A.4) for *any* f, $\mathbf{X}$, and valid μ, and ν? Certainly there are some functions f that are not suitable for the CAVE framework. Take for example a Heaviside function

$$u(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{A.60}$$

which is clearly a range-bounded function. The CAVE framework would not work in part because only a set of discrete means and variances can be matched. For $N = 10$, there can only be means of $\mu = 0.1k$ for $k \in \{0, 1, \ldots, 10\}$. A specified mean of $\mu = 0.01$ for example is infeasible.

We need to address certain properties of f needed to make the CAVE framework feasible for any data $\mathbf{X}$ and all mean μ and variance ν provided they both comply with their constraints in Section A.3. We'll revisit the constraints on function f introduced without much explanation in Section A.1:

(1) $f : \mathbb{R} \to \mathbb{R}$, and operates element-wise for multidimensional inputs,

(2) f is range limited to $[f_{min}, f_{max}]$ for some $-\infty < f_{min} < f_{max} < \infty$,

(3) f is smooth,

(4) f is strictly increasing.

As will be shown, these properties are needed to prove that there does exist an a and b that satisfies CAVE Constraints (A.1)–(A.4) for *any* f and $\mathbf{X}$ as well as any valid μ and ν values.

A.3.1. CAVE Mean Constraints

In order to establish the bounds for the mean μ, we first examine the lower bound. The minimization Eq. (A.8) over $\mathbf{X}$ to get the lower bound for the mean, μ_{min}:

$$\mu_{min} = \min_{\mathbf{X}} \ (h - l) \ \frac{\mathbb{E}\left[f\left(a\,\mathbf{X} + b\right)\right] - f_{min}}{f_{max} - f_{min}} + l. \tag{A.61}$$

Since we are in a minimization environment, we simplify by removing the summands without $\mathbf{X}$, as well as the coefficient of the expectation since it is strictly a positive value:

$$\mu'_{min} = \min_{\mathbf{X}} \ \mathbb{E}\left[f\left(a\,\mathbf{X} + b\right)\right]. \tag{A.62}$$

This is the same μ' as in Eq. (A.14). To minimize the expectation, we assume $f\left(a\,\mathbf{X}_i + b\right) = f_{min}$ for all i. Thus,

$$\mu'_{min} = f_{min}. \tag{A.63}$$

A similar argument follows to find the upper bound of the mean, μ_{max}, where we instead perform a maximization in Eq. (A.61):

$$\mu'_{max} = f_{max}. \tag{A.64}$$

The feasible range of μ' is then

$$f_{min} \leq \mu' \leq f_{max}. \tag{A.65}$$

Substituting μ'_{min} and μ'_{max} separately into Eq. (A.61), we conclude that the specified mean must be in the range

$$l \leq \mu \leq h \tag{A.66}$$

in order for the CAVE optimization to have a minimum of 0.

A.3.2. CAVE Mean Feasibility

To show that any mean under Constraint (A.66) can be found, we will fix $a = 1$ and only provide a linear shift. We want to show that any b can be found to satisfy Eq. (A.10) for any mean μ under Constraint (A.66).

First, we note that f must be strictly increasing, and its output range is $[f_{min}, f_{max}]$. This implies that

$$\lim_{b \to -\infty} f(x + b) = f_{min} \tag{A.67}$$

$$\lim_{b \to \infty} f(x + b) = f_{max} \tag{A.68}$$

for any $x \in \mathbb{R}$. We can apply these limits to Eq. (A.8):

$$(h - l) \frac{\mathbb{E}\left[\lim_{b \to -\infty} f(\mathbf{X} + b)\right] - f_{min}}{f_{max} - f_{min}} + l = l \tag{A.69}$$

$$(h-l)\ \frac{\mathbb{E}\left[\lim_{b\to\infty} f\left(\mathbf{X}+b\right)\right]-f_{min}}{f_{max}-f_{min}}+l=h. \tag{A.70}$$

Since f is defined as being smooth, and the expectation operator is smooth about b, we conclude by the intermediate value theorem that there exists an a and b such that Eq. (A.10) is satisfied for any mean μ under Constraint (A.66).

A.3.3. CAVE Variance Constraints

Here, we establish an upper bound on the variance ν_{max} such that the minimum of the CAVE loss function is 0. The variance is always nonnegative, so the variance lower bound is simply

$$\nu_{min}=0. \tag{A.71}$$

For the maximum variance, we maximize Eq. (A.9):

$$\nu_{max}=\max_{\mathbf{X}}\ \left(\frac{h-l}{f_{max}-f_{min}}\right)^2 \mathrm{var}\left(f\left(a\,\mathbf{X}+b\right)\right). \tag{A.72}$$

We can remove the coefficient of the variance term since it is nonnegative, so we have

$$\nu'_{max}=\max_{\mathbf{X}}\ \mathrm{var}\left(f\left(a\,\mathbf{X}+b\right)\right). \tag{A.73}$$

It's been shown through Popoviciu's inequality of variance [81] that the maximum variance of a bounded random variable is

$$\nu'_{max}=\frac{\left(f_{max}-f_{min}\right)^2}{4}. \tag{A.74}$$

For a sample population, the maximum variance depends on the parity of N [84]:

$$\nu'_{max} = \begin{cases} \frac{(f_{max}-f_{min})^2}{4}, & N \text{ even} \\ \frac{\left(N^2-1\right)(f_{max}-f_{min})^2}{4N^2}, & N \text{ odd} \end{cases}. \tag{A.75}$$

After substituting ν'_{max} into Eq. (A.72), we conclude that the specified variance ν must be in the range

$$0 \leq \nu \leq \begin{cases} \frac{(h-l)^2}{4}, & N \text{ even} \\ \frac{\left(N^2-1\right)(h-l)^2}{4N^2}, & N \text{ odd} \end{cases} \tag{A.76}$$

for the CAVE loss to have a minimum of 0.

A.3.4. CAVE Variance Feasibility

To show that any variance under Constraint (A.76) can be found, we first use a slightly different notation than that of (A.9):

$$\left(\frac{h-l}{f_{max}-f_{min}}\right)^2 \text{var}\left(f\left(a\left(\mathbf{X}-b\right)\right)\right) = \nu. \tag{A.77}$$

We then find b such that

$$\|\{i \mid \mathbf{X}_i < b\}\| = \left\lfloor \frac{N}{2} \right\rfloor \tag{A.78}$$

$$b \notin \mathbf{X}. \tag{A.79}$$

The value of b can be anywhere between the two middle values of $\mathbf{X}$. Half of the values of $\mathbf{X} - b$ are negative, and the other half are positive. Since b is not in $\mathbf{X}$, there are no zero values of $\mathbf{X} - b$.

We now show that any a can be found to satisfy Eq. (A.11) for any variance ν under Constraint (A.76). The lower bound of the variance is $\nu = 0$, which is easily achieved by choosing $a = 0$. This gives

$$\left(\frac{h-l}{f_{max} - f_{min}}\right)^2 \text{var}\left(f\left(\mathbf{0}\right)\right) = 0 \tag{A.80}$$

where $\mathbf{0} \in \{0\}^N$. All the values of $f\left(\mathbf{0}\right)$ are the same, thus the variance is 0.

For the upper bound, we examine what happens as $a \to \infty$:

$$\lim_{a\to\infty} f\left(a\left(\mathbf{X} - b\right)\right) \in \{f_{min}, f_{max}\}^N. \tag{A.81}$$

As $a \to \infty$, the values where $\mathbf{X}_i - b < 0$ approach f_{min} while the values where $\mathbf{X}_i - b > 0$ approach f_{max}. Recall that half of the values of $\mathbf{X}$ are greater than b, and the other half are less than b. For an even N,

$$\text{var}\left(\lim_{a\to\infty} f\left(a\left(\mathbf{X} - b\right)\right)\right) = \frac{\frac{N}{2} f_{min}^2 + \frac{N}{2} f_{max}^2}{N} - \left(\frac{\frac{N}{2} f_{min} + \frac{N}{2} f_{max}}{N}\right)^2 \tag{A.82}$$

which simplifies to

$$\text{var}\left(\lim_{a\to\infty} f\left(a\left(\mathbf{X} - b\right)\right)\right) = \frac{\left(f_{max} - f_{min}\right)^2}{4}. \tag{A.83}$$

For an odd N,

$$\text{(A.84)} \qquad \operatorname{var}\left(\lim_{a\to\infty} f\left(a\left(\mathbf{X}-b\right)\right)\right) = \frac{\frac{N-1}{2}f_{min}^2 + \frac{N+1}{2}f_{max}^2}{N} - \left(\frac{\frac{N-1}{2}f_{min} + \frac{N+1}{2}f_{max}}{N}\right)^2$$

which simplifies to

$$\text{(A.85)} \qquad \operatorname{var}\left(\lim_{a\to\infty} f\left(a\left(\mathbf{X}-b\right)\right)\right) = \frac{\left(N^2-1\right)\left(f_{max}-f_{min}\right)^2}{4N^2}.$$

We can then conclude that

$$\text{(A.86)} \qquad \left(\frac{h-l}{f_{max}-f_{min}}\right)^2 \operatorname{var}\left(\lim_{a\to\infty} f\left(a\left(\mathbf{X}-b\right)\right)\right) = \begin{cases} \frac{(h-l)^2}{4}, & N \text{ even} \\ \frac{\left(N^2-1\right)(h-l)^2}{4N^2}, & N \text{ odd} \end{cases}$$

which are the upper bounds of Constraint (A.76).

Since f is defined as smooth, and the variance operator is itself smooth about a, we can conclude via the intermediate value theorem that there exists an a and b such that Eq. (A.11) is satisfied for any variance ν under Constraint (A.76).

A.3.5. CAVE Joint Mean & Variance Constraints

While Constraints (A.66) and (A.76) can be applied when optimizing either the mean or the variance, joint optimization of the mean and variance introduces further constraints on the allowed mean and variance. The Bhatia-Davis inequality [82] provides a tighter constraint on the variance of a bounded random variable than Popoviciu's inequality of variance. When the mean is given, the variance has an upper bound of

$$\text{(A.87)} \qquad \nu \leq \left(h-\mu\right)\left(\mu-l\right).$$

In a finite sample population, this only works in certain cases.

As a simple example, let $N = 2$, $l = 0$, $h = 1$, and $\mu = 0.1$. To maximize the variance, we maximize the distance from $\mathbf{X}_1$ and $\mathbf{X}_2$, so $\mathbf{X}_1 = 0$ and $\mathbf{X}_2 = 0.2$ lies within the bounds and has a mean $\mu = 0.1$. The maximum variance here would be 0.01, but the upper bound from Constraint (A.87) would be 0.09. Clearly, a tighter bound is possible for sample populations instead of random variables.

We prove a new upper bound by first showing the cases when Constraint (A.87) holds true. Using a similar argument to that of Section A.3.4, we first use the slightly different notation than that of Eq. (A.9):

$$\left(\frac{h-l}{f_{max}-f_{min}}\right)^2 \operatorname{var}\left(f\left(a\left(\mathbf{X}-b\right)\right)\right) = \nu. \tag{A.88}$$

We then find b_k such that

$$\|\{i \mid \mathbf{X}_i < b_k\}\| = k \tag{A.89}$$

for some $k \in \mathbb{Z}$ where $0 \leq k \leq N$. Here, Eq. (A.81) holds true as well: as $a \to \infty$, $f\left(a\left(\mathbf{X}-b_k\right)\right) = \{f_{min}, f_{max}\}^N$. In fact, k of the values of $\lim_{a\to\infty} f\left(a\left(\mathbf{X}-b_k\right)\right)$ are f_{min}, and the other $N-k$ values are f_{max}. This gives an average of

$$\mathbb{E}\left[\lim_{a\to\infty} f\left(a\left(\mathbf{X}-b_k\right)\right)\right] = \mu'_k \tag{A.90}$$

$$\mu'_k = \frac{k f_{min} + (N-k) f_{max}}{N}. \tag{A.91}$$

The non-adjusted mean follows the transform of Eq. (A.14), which yields

$$\mu_k = \frac{kl + (N-k)\,h}{N}, \tag{A.92}$$

where each μ_k is within the bounds of Constraint (A.66). The variance is then

$$\operatorname{var}\left(\lim_{a\to\infty} f\left(a\left(\mathbf{X} - b_k\right)\right)\right) = \nu'_k \tag{A.93}$$

$$\nu'_k = \frac{k\,(N-k)\,(f_{max} - f_{min})^2}{N^2}. \tag{A.94}$$

Notably, we can massage the equation to show that

$$\nu'_k = \left(f_{max} - \frac{kf_{min} + (N-k)\,f_{max}}{N}\right)\left(\frac{kf_{min} + (N-k)\,f_{max}}{N} - f_{min}\right) \tag{A.95}$$

$$= (f_{max} - \mu'_k)\,(\mu'_k - f_{min})\,. \tag{A.96}$$

By substituting Eq. (A.14) in and using the relation of Eq. (A.15), we get the non-adjusted variance ν_k as

$$\nu_k = (h - \mu_k)\,(\mu_k - l)\,, \tag{A.97}$$

which is the upper bound of Constraint (A.87). Therefore,

$$\left(\frac{h-l}{f_{max} - f_{min}}\right)^2 \operatorname{var}\left(\lim_{a\to\infty} f\left(a\left(\mathbf{X} - b_k\right)\right)\right) = (h - \mu_k)\,(\mu_k - l) \tag{A.98}$$

is a maximum variance for μ_k.

For other means $\mu \neq \mu_k$, we show that the maximum variance is achieved when all values of $\mathbf{X}_i \in \{f_{min}, f_{max}\}^N$ for all i except for $i = j$ for some j; the value $\mathbf{X}_j$ is bounded

by $l < \mathbf{X}_j < h$. This is shown via a relaxation of Eq. (A.81) where instead of using the limit $a \to \infty$, we instead use the limit $a \to M$ for some sufficiently large $M < \infty$:

$$\lim_{a \to M} f\left(a\left(\mathbf{X}_i - b_k\right)\right) \in \begin{cases} \left\{f_{min} + \delta_i, f_{max} - \epsilon_i\right\}^N, & i \neq j \\ \left(f_{min}, f_{max}\right) & i = j \end{cases} \tag{A.99}$$

for some small $0 < \delta_i, \epsilon_i$. This relaxation is important in order to be able to control the location of $\mathbf{X}_j$. We can then find a b_k such that $M\left(\mathbf{X}_j - b_k\right) = c$ for some finite $0 < c \ll \infty$ when $\mathbf{X}_j > b_k$ or some $-\infty \ll c < 0$ when $\mathbf{X}_j < b_k$. The value of $f\left(c\right) = d$ is then

$$f_{min} < d < f_{max}. \tag{A.100}$$

We first find the mean of this set of $a \to M$ and $b = b_k$:

$$\mathbb{E}\left[\lim_{a \to M} f\left(a\left(\mathbf{X}_i - b_k\right)\right)\right] = \mu_k' \tag{A.101}$$

$$\mu_k' \approx \frac{k f_{min} + \left(N - k - 1\right) f_{max} + d}{N}. \tag{A.102}$$

From Constraint (A.100), we can establish bounds on μ_k' as

$$\frac{\left(k+1\right) f_{min} + \left(N - k - 1\right) f_{max} + d}{N} < \mu_k' < \frac{k f_{min} + \left(N - k\right) f_{max} + d}{N}. \tag{A.103}$$

Note that as $0 \leq k \leq N - 1$ varies, we encompass the remaining values of the mean.

We then examine the variance when $a \to M$ and $b = b_k$:

$$\text{var}\left(\lim_{a \to M} f\left(a\left(\mathbf{X}_i - b_k\right)\right)\right) = \nu_k' \tag{A.104}$$

where

$$\nu'_k \approx \frac{k f_{min}^2 + (N-k-1) f_{max}^2 + d^2}{N} - \left(\frac{k f_{min} + (N-k-1) f_{max} + d}{N} \right)^2, \tag{A.105}$$

which cannot meaningfully be simplified further. We claim that this is the maximum possible variance when $\mu'_k \neq \frac{k f_{min} + (N-k) f_{max}}{N}$. The non-adjusted variance is then

$$\nu_k \approx \left(\frac{h-l}{f_{max} - f_{min}} \right)^2 \nu'_k. \tag{A.106}$$

To prove that only one value should deviate from the extrema to get the maximum variance for means $\mu'_k \neq \frac{k f_{min} + (N-k) f_{max}}{N}$, let's design a maximization problem. We want to maximize the variance of $f\left(a\left(\mathbf{X} - b\right)\right)$ given the mean of the data by changing the number of non-extrema values in $\mathbf{X}$. From the definition of the mean here, we formulate the mean as

$$\mu'_m = \frac{k f_{min} + (N-k-m) f_{max} + \sum_{i=1}^{m} c_i}{N} \tag{A.107}$$

where m is the number of non-extrema values in $\mathbf{X}$, and $f_{min} < c_i < f_{max}$ for all i. The range of means covered by μ'_m is

$$\frac{(k+m) f_{min} + (N-k-m) f_{max}}{N} < \mu'_m < \frac{k f_{min} + (N-k) f_{max}}{N}. \tag{A.108}$$

For any $m \geq 1$, as k varies $0 \leq k \leq N - m$, it can be shown that it covers the range of possible mean values $f_{min} < \mu'_m < f_{max}$.

Without loss of generality, we perform a linear shift of $f(a\,\mathbf{X}+b)$ by $-f_{min}$. The new range is

$$0 \leq f(a\,\mathbf{X}+b) - f_{min} \leq f_{max} - f_{min}. \tag{A.109}$$

We can do this since the variance is shift invariant. The variance of the transformed variable is

$$\nu'_k = \frac{(N-k-m)\,f_{max}^2 + \sum_{i=1}^{m} c_i^2}{N} - \left(\frac{(N-k-m)\,f_{max} + \sum_{i=1}^{m} c_i}{N} \right)^2. \tag{A.110}$$

If we maximize ν'_k over m, we can greatly simplify it by (1) removing terms independent of m, and (2) solving for and substituting $\sum_{i=1}^{m} c_i$ in Eq. (A.107):

$$\arg\max_m \; \nu'_k = -(f_{max} - f_{min})^2\, m + \sum_{i=1}^{m} c_i^2 \tag{A.111}$$

for $0 < c_i < f_{max} - f_{min}$. Since $0 < c_i < f_{max} - f_{min}$, we can maximize the variance when $m = 0$, the case we covered previously with a discrete set of means. We can see that as m increases, we deviate further away from the maximal variance of 0, however we can cover the remainder of the range of means. Thus, we conclude that $m = 1$, and we found the maximum variance in Eq. (A.106) when $\mu \neq \frac{kl+(N-k)h}{N}$ for $0 \leq k \leq N$.

A.3.6. CAVE Mean & Variance Feasibility

We have shown in Section A.3.2 that we can achieve a mean $l \leq \mu \leq h$. We also know through the intermediate value theorem that $f(c)$ can take on any value $[f_{min}, f_{max}]$ for

some c. By setting $a = 0$, we quickly show that it is possible to achieve zero variance for any mean $l \leq \mu \leq h$:

$$\left(\frac{h-l}{f_{max}-f_{min}}\right)^2 \operatorname{var}\left(f\left(\mathbf{b}\right)\right) = 0 \tag{A.112}$$

for $\mathbf{b} \in \{b\}^N$. All the values of $f\left(\mathbf{b}\right)$ are the same, so the variance is 0. We are also able to find b such that $f\left(b\right) = \mu'$ from Eq. (A.14) with bounds from Constraint (A.65). Again, because the values of $\mathbf{b}$ are identical, we quickly verify that

$$\left(h-l\right)\frac{\mathbb{E}\left[f\left(\mathbf{b}\right)\right] - f_{min}}{f_{max}-f_{min}} + l = \mu. \tag{A.113}$$

Additionally, we showed in Section A.3.5 that there exist values of a and b that can approximately achieve the maximal variance given a mean. Since we found bounds for zero variance and maximal variance for all μ, and the variance operator is smooth for the non-extreme value c, we can conclude again via the intermediate value theorem that we can find any mean and variance within the mean bounds of Constraint (A.66) and variance bounds of Constraint (A.106).

APPENDIX B

The Gradient of CAVE Functions

As is outlined in Appendix A, there are many calculations that need to be carried out even for a single gradient descent or Newton's method step. This is not an issue in a mathematical sense, but becomes an issue for implementation. PyTorch [80] for example stores the input of operations in order to track the gradients. This can lead to GPU memory issues if the process is not simplified to store fewer inputs.

In this appendix, we find the gradient of one update step of gradient descent $\mathcal{S}_G$ and Newton's method $\mathcal{S}_N$ not with respect to a or b, but to the input $\mathbf{X}$. Variable $\mathbf{X}$ contains the information on the weights outside of the CAVE environment. Specifically, we derive

$$\nabla_{\mathbf{X}} \mathcal{S}_G = \nabla_{\mathbf{X}} \left(\nabla_{a,b} \mathcal{C} \right) \tag{B.1}$$

$$\nabla_{\mathbf{X}} \mathcal{S}_N = \nabla_{\mathbf{X}} \left(\left(\nabla^2_{a,b} \mathcal{C} \right)^{-1} \left(\nabla_{a,b} \mathcal{C} \right) \right) \tag{B.2}$$

where $\nabla_{\mathbf{X}}$ is the gradient operator with respect to all the values in $\mathbf{X}$.

We find the derivative of the step functions since it maximizes the number of operations. Only the input $\mathbf{X}$ to each gradient or Newton step needs to be saved, thus drastically reducing the memory allocated for backpropagation. We choose one step since it is impractical to unravel multiple steps of either gradient descent or Newton's method.

We continue the derivation from the end of Appendix A using the same shorthand notation. Additionally, we will only explicitly solve for $\mathcal{S}_N$ in terms of the base function f. Step function $\mathcal{S}_G$ is found via the solution of $\mathcal{S}_N$.

B.1. Gradient of Newton's Method w.r.t. $\mathbf{X}_i$

We can expand the gradient step by

$$\left(\nabla^2_{a,b}\mathcal{C}\right)^{-1}\nabla_{a,b}\mathcal{C} = \frac{1}{D}\begin{bmatrix} N_a \\ N_b \end{bmatrix} \tag{B.3}$$

where

$$N_a = \frac{\partial\mathcal{C}}{\partial a}\frac{\partial^2\mathcal{C}}{\partial b^2} - \frac{\partial\mathcal{C}}{\partial b}\frac{\partial^2\mathcal{C}}{\partial a\partial b} \tag{B.4}$$

$$N_b = \frac{\partial\mathcal{C}}{\partial b}\frac{\partial^2\mathcal{C}}{\partial a^2} - \frac{\partial\mathcal{C}}{\partial a}\frac{\partial^2\mathcal{C}}{\partial a\partial b} \tag{B.5}$$

$$D = \frac{\partial^2\mathcal{C}}{\partial a^2}\frac{\partial^2\mathcal{C}}{\partial b^2} - \left(\frac{\partial^2\mathcal{C}}{\partial a\partial b}\right)^2. \tag{B.6}$$

All of these values were found in Appendix A, however we need to find the derivative of each entry with respect to each $\mathbf{X}_i$:

$$\frac{\partial}{\partial\mathbf{X}_i}\left[\frac{N_a}{D}\right] = \frac{D\frac{\partial N_a}{\partial\mathbf{X}_i} - N_a\frac{\partial D}{\partial\mathbf{X}_i}}{D^2} \tag{B.7}$$

$$\frac{\partial}{\partial\mathbf{X}_i}\left[\frac{N_b}{D}\right] = \frac{D\frac{\partial N_b}{\partial\mathbf{X}_i} - N_b\frac{\partial D}{\partial\mathbf{X}_i}}{D^2} \tag{B.8}$$

where

$$\frac{\partial N_a}{\partial\mathbf{X}_i} = \frac{\partial^2\mathcal{C}}{\partial a\partial\mathbf{X}_i}\cdot\frac{\partial^2\mathcal{C}}{\partial b^2} + \frac{\partial\mathcal{C}}{\partial a}\cdot\frac{\partial^3\mathcal{C}}{\partial b^2\partial\mathbf{X}_i} - \frac{\partial^2\mathcal{C}}{\partial b\partial\mathbf{X}_i}\cdot\frac{\partial^2\mathcal{C}}{\partial a\partial b} - \frac{\partial\mathcal{C}}{\partial b}\cdot\frac{\partial^3\mathcal{C}}{\partial a\partial b\partial\mathbf{X}_i} \tag{B.9}$$

$$\frac{\partial N_b}{\partial \mathbf{X}_i} = \frac{\partial^2 \mathcal{C}}{\partial b \partial \mathbf{X}_i} \cdot \frac{\partial^2 \mathcal{C}}{\partial a^2} + \frac{\partial \mathcal{C}}{\partial b} \cdot \frac{\partial^3 \mathcal{C}}{\partial a^2 \partial \mathbf{X}_i} - \frac{\partial^2 \mathcal{C}}{\partial a \partial \mathbf{X}_i} \cdot \frac{\partial^2 \mathcal{C}}{\partial a \partial b} - \frac{\partial \mathcal{C}}{\partial a} \cdot \frac{\partial^3 \mathcal{C}}{\partial a \partial b \partial \mathbf{X}_i} \tag{B.10}$$

$$\frac{\partial D}{\partial \mathbf{X}_i} = \frac{\partial^3 \mathcal{C}}{\partial a^2 \partial \mathbf{X}_i} \cdot \frac{\partial^2 \mathcal{C}}{\partial b^2} + \frac{\partial^2 \mathcal{C}}{\partial a^2} \cdot \frac{\partial^3 \mathcal{C}}{\partial b^2 \partial \mathbf{X}_i} - 2\frac{\partial^2 \mathcal{C}}{\partial a \partial b} \cdot \frac{\partial^3 \mathcal{C}}{\partial a \partial b \partial \mathbf{X}_i}. \tag{B.11}$$

Now, we need to find the new derivatives with respect to $\mathbf{X}_i$. These derivatives of $\mathcal{L}$ are decomposed into their $\mathcal{C}_\mu$ and $\mathcal{C}_\nu$ counterparts:

$$\frac{\partial^2 \mathcal{C}}{\partial a \partial \mathbf{X}_i} = \frac{\partial^2 \mathcal{C}_\mu}{\partial a \partial \mathbf{X}_i} + \frac{\partial^2 \mathcal{C}_\nu}{\partial a \partial \mathbf{X}_i} \tag{B.12}$$

$$\frac{\partial^2 \mathcal{C}}{\partial b \partial \mathbf{X}_i} = \frac{\partial^2 \mathcal{C}_\mu}{\partial b \partial \mathbf{X}_i} + \frac{\partial^2 \mathcal{C}_\nu}{\partial b \partial \mathbf{X}_i} \tag{B.13}$$

$$\frac{\partial^3 \mathcal{C}}{\partial a^2 \partial \mathbf{X}_i} = \frac{\partial^3 \mathcal{C}_\mu}{\partial a^2 \partial \mathbf{X}_i} + \frac{\partial^3 \mathcal{C}_\nu}{\partial a^2 \partial \mathbf{X}_i} \tag{B.14}$$

$$\frac{\partial^3 \mathcal{C}}{\partial a \partial b \partial \mathbf{X}_i} = \frac{\partial^3 \mathcal{C}_\mu}{\partial a \partial b \partial \mathbf{X}_i} + \frac{\partial^3 \mathcal{C}_\nu}{\partial a \partial b \partial \mathbf{X}_i} \tag{B.15}$$

$$\frac{\partial^3 \mathcal{C}}{\partial b^2 \partial \mathbf{X}_i} = \frac{\partial^3 \mathcal{C}_\mu}{\partial b^2 \partial \mathbf{X}_i} + \frac{\partial^3 \mathcal{C}_\nu}{\partial b^2 \partial \mathbf{X}_i}. \tag{B.16}$$

B.1.1. Gradient of Losses w.r.t. $\mathbf{X}_i$

Using previous definitions, we can find the above loss derivatives. The second order derivatives are

$$\frac{\partial^2 \mathcal{C}_\mu}{\partial a \partial \mathbf{X}_i} = 2\left(\frac{\partial E_\mu}{\partial \mathbf{X}_i} \cdot \frac{\partial E_\mu}{\partial a} + E_\mu \cdot \frac{\partial^2 E_\mu}{\partial a \partial \mathbf{X}_i}\right) \tag{B.17}$$

$$\frac{\partial^2 \mathcal{C}_\nu}{\partial a \partial \mathbf{X}_i} = 2\left(\frac{\partial E_\nu}{\partial \mathbf{X}_i} \cdot \frac{\partial E_\nu}{\partial a} + E_\nu \cdot \frac{\partial^2 E_\nu}{\partial a \partial \mathbf{X}_i}\right) \tag{B.18}$$

$$\frac{\partial^2 \mathcal{C}_\mu}{\partial b \partial \mathbf{X}_i} = 2\left(\frac{\partial E_\mu}{\partial \mathbf{X}_i} \cdot \frac{\partial E_\mu}{\partial b} + E_\mu \cdot \frac{\partial^2 E_\mu}{\partial b \partial \mathbf{X}_i}\right) \tag{B.19}$$

$$\frac{\partial^2 \mathcal{C}_\nu}{\partial b \partial \mathbf{X}_i} = 2\left(\frac{\partial E_\nu}{\partial \mathbf{X}_i} \cdot \frac{\partial E_\nu}{\partial b} + E_\nu \cdot \frac{\partial^2 E_\nu}{\partial b \partial \mathbf{X}_i}\right) \tag{B.20}$$

and the third derivatives are

$$\frac{\partial^3 \mathcal{C}_\mu}{\partial a^2 \partial \mathbf{X}_i} = 2\left(2\frac{\partial E_\mu}{\partial a}\cdot\frac{\partial^2 E_\mu}{\partial a \partial \mathbf{X}_i} + \frac{\partial E_\mu}{\partial \mathbf{X}_i}\cdot\frac{\partial^2 E_\mu}{\partial a^2} + E_\mu \cdot \frac{\partial^3 E_\mu}{\partial a^2 \partial \mathbf{X}_i}\right) \tag{B.21}$$

$$\frac{\partial^3 \mathcal{C}_\nu}{\partial a^2 \partial \mathbf{X}_i} = 2\left(2\frac{\partial E_\nu}{\partial a}\cdot\frac{\partial^2 E_\nu}{\partial a \partial \mathbf{X}_i} + \frac{\partial E_\nu}{\partial \mathbf{X}_i}\cdot\frac{\partial^2 E_\nu}{\partial a^2} + E_\nu \cdot \frac{\partial^3 E_\nu}{\partial a^2 \partial \mathbf{X}_i}\right) \tag{B.22}$$

$$\frac{\partial^3 \mathcal{C}_\mu}{\partial a \partial b \partial \mathbf{X}_i} = 2\left(\frac{\partial^2 E_\mu}{\partial a \partial \mathbf{X}_i}\cdot\frac{\partial E_\mu}{\partial b} + \frac{\partial E_\mu}{\partial a}\cdot\frac{\partial^2 E_\mu}{\partial b \partial \mathbf{X}_i} + \frac{\partial E_\mu}{\partial \mathbf{X}_i}\cdot\frac{\partial^2 E_\mu}{\partial a \partial b} + E_\mu \cdot \frac{\partial^3 E_\mu}{\partial a \partial b \partial \mathbf{X}_i}\right) \tag{B.23}$$

$$\frac{\partial^3 \mathcal{C}_\nu}{\partial a \partial b \partial \mathbf{X}_i} = 2\left(\frac{\partial^2 E_\nu}{\partial a \partial \mathbf{X}_i}\cdot\frac{\partial E_\nu}{\partial b} + \frac{\partial E_\nu}{\partial a}\cdot\frac{\partial^2 E_\nu}{\partial b \partial \mathbf{X}_i} + \frac{\partial E_\nu}{\partial \mathbf{X}_i}\cdot\frac{\partial^2 E_\nu}{\partial a \partial b} + E_\nu \cdot \frac{\partial^3 E_\nu}{\partial a \partial b \partial \mathbf{X}_i}\right) \tag{B.24}$$

$$\frac{\partial^3 \mathcal{C}_\mu}{\partial b^2 \partial \mathbf{X}_i} = 2\left(2\frac{\partial E_\mu}{\partial b}\cdot\frac{\partial^2 E_\mu}{\partial b \partial \mathbf{X}_i} + \frac{\partial E_\mu}{\partial \mathbf{X}_i}\cdot\frac{\partial^2 E_\mu}{\partial b^2} + E_\mu \cdot \frac{\partial^3 E_\mu}{\partial b^2 \partial \mathbf{X}_i}\right) \tag{B.25}$$

$$\frac{\partial^3 \mathcal{C}_\nu}{\partial b^2 \partial \mathbf{X}_i} = 2\left(2\frac{\partial E_\nu}{\partial b}\cdot\frac{\partial^2 E_\nu}{\partial b \partial \mathbf{X}_i} + \frac{\partial E_\nu}{\partial \mathbf{X}_i}\cdot\frac{\partial^2 E_\nu}{\partial b^2} + E_\nu \cdot \frac{\partial^3 E_\nu}{\partial b^2 \partial \mathbf{X}_i}\right). \tag{B.26}$$

B.1.2. Gradient of Errors w.r.t. $\mathbf{X}_i$

Next, we find the derivatives of the error functions. The mean error functions are

$$\frac{\partial E_\mu}{\partial \mathbf{X}_i} = \frac{1}{N}\frac{\partial f}{\partial \mathbf{X}_i} \tag{B.27}$$

$$\frac{\partial^2 E_\mu}{\partial a \partial \mathbf{X}_i} = \frac{1}{N}\frac{\partial^2 f}{\partial a \partial \mathbf{X}_i} \tag{B.28}$$

$$\frac{\partial^2 E_\mu}{\partial b \partial \mathbf{X}_i} = \frac{1}{N}\frac{\partial^2 f}{\partial b \partial \mathbf{X}_i} \tag{B.29}$$

$$\frac{\partial^3 E_\mu}{\partial a^2 \partial \mathbf{X}_i} = \frac{1}{N}\frac{\partial^3 f}{\partial a^2 \partial \mathbf{X}_i} \tag{B.30}$$

$$\frac{\partial^3 E_\mu}{\partial a \partial b \partial \mathbf{X}_i} = \frac{1}{N}\frac{\partial^3 f}{\partial a \partial b \partial \mathbf{X}_i} \tag{B.31}$$

$$\frac{\partial^3 E_\mu}{\partial b^2 \partial \mathbf{X}_i} = \frac{1}{N}\frac{\partial^3 f}{\partial b^2 \partial \mathbf{X}_i} \tag{B.32}$$

and the variance error functions are

$$\frac{\partial E_\nu}{\partial \mathbf{X}_i} = \frac{2}{N}\left(\frac{\partial f}{\partial \mathbf{X}_i} - \mathbb{E}\left[f\right]\frac{\partial f}{\partial \mathbf{X}_i}\right) \tag{B.33}$$

$$\frac{\partial^2 E_\nu}{\partial a \partial \mathbf{X}_i} = \frac{2}{N}\left(\frac{\partial f}{\partial a}\frac{\partial f}{\partial \mathbf{X}_i} + f\frac{\partial^2 f}{\partial a \partial \mathbf{X}_i} - \frac{\partial E_\mu}{\partial a}\frac{\partial f}{\partial \mathbf{X}_i} - \mathbb{E}\left[f\right]\frac{\partial^2 f}{\partial a \partial \mathbf{X}_i}\right) \tag{B.34}$$

$$\frac{\partial^2 E_\nu}{\partial b \partial \mathbf{X}_i} = \frac{2}{N}\left(\frac{\partial f}{\partial b}\frac{\partial f}{\partial \mathbf{X}_i} + f\frac{\partial^2 f}{\partial b \partial \mathbf{X}_i} - \frac{\partial E_\mu}{\partial b}\frac{\partial f}{\partial \mathbf{X}_i} - \mathbb{E}\left[f\right]\frac{\partial^2 f}{\partial b \partial \mathbf{X}_i}\right) \tag{B.35}$$

$$\begin{aligned}\frac{\partial^3 E_\nu}{\partial a^2 \partial \mathbf{X}_i} = \frac{2}{N}\Bigg(&2\frac{\partial f}{\partial a}\frac{\partial^2 f}{\partial a \partial \mathbf{X}_i} + \frac{\partial f}{\partial \mathbf{X}_i}\frac{\partial^2 f}{\partial a^2} + f\frac{\partial^3 f}{\partial a^2 \partial \mathbf{X}_i} - \cdots \\ &2\frac{\partial E_\mu}{\partial a}\frac{\partial^2 f}{\partial a \partial \mathbf{X}_i} - \frac{\partial^2 E_\mu}{\partial a^2}\frac{\partial f}{\partial \mathbf{X}_i} - \mathbb{E}\left[f\right]\frac{\partial^3 f}{\partial a^2 \partial \mathbf{X}_i}\Bigg)\end{aligned} \tag{B.36}$$

$$\begin{aligned}\frac{\partial^3 E_\nu}{\partial a \partial b \partial \mathbf{X}_i} = \frac{2}{N}\Bigg(&\frac{\partial^2 f}{\partial a \partial \mathbf{X}_i}\frac{\partial f}{\partial b} + \frac{\partial^2 f}{\partial b \partial \mathbf{X}_i}\frac{\partial f}{\partial a} + \frac{\partial f}{\partial \mathbf{X}_i}\frac{\partial^2 f}{\partial a \partial b} + \frac{\partial^3 f}{\partial a \partial b \partial \mathbf{X}_i}f - \cdots \\ &\frac{\partial E_\mu}{\partial a}\frac{\partial^2 f}{\partial b \partial \mathbf{X}_i} - \frac{\partial E_\mu}{\partial b}\frac{\partial^2 f}{\partial a \partial \mathbf{X}_i} - \frac{\partial^2 E_\mu}{\partial a \partial b}\frac{\partial f}{\partial \mathbf{X}_i} - \mathbb{E}\left[f\right]\frac{\partial^3 f}{\partial a \partial b \partial \mathbf{X}_i}\Bigg)\end{aligned} \tag{B.37}$$

$$\begin{aligned}\frac{\partial^3 E_\nu}{\partial b^2 \partial \mathbf{X}_i} = \frac{2}{N}\Bigg(&2\frac{\partial f}{\partial b}\frac{\partial^2 f}{\partial b \partial \mathbf{X}_i} + \frac{\partial f}{\partial \mathbf{X}_i}\frac{\partial^2 f}{\partial b^2} + f\frac{\partial^3 f}{\partial b^2 \partial \mathbf{X}_i} - \cdots \\ &2\frac{\partial E_\mu}{\partial b}\frac{\partial^2 f}{\partial b \partial \mathbf{X}_i} - \frac{\partial^2 E_\mu}{\partial b^2}\frac{\partial f}{\partial \mathbf{X}_i} - \mathbb{E}\left[f\right]\frac{\partial^3 f}{\partial b^2 \partial \mathbf{X}_i}\Bigg).\end{aligned} \tag{B.38}$$

B.1.3. Gradient of CAVE Base Functions w.r.t. $\mathbf{X}_i$

Lastly, we can find the additional function derivatives as

$$\frac{\partial f}{\partial \mathbf{X}_i} = \frac{\partial}{\partial \mathbf{X}_i}\left[f\left(a\,\mathbf{X}_i + b\right)\right] = f'\left(a\,\mathbf{X}_i + b\right)a \tag{B.39}$$

$$\frac{\partial^2 f}{\partial a \partial \mathbf{X}_i} = \frac{\partial^2}{\partial a \partial \mathbf{X}_i}\left[f\left(a\,\mathbf{X}_i + b\right)\right] = f''\left(a\,\mathbf{X}_i + b\right)a\,\mathbf{X}_i + f'\left(a\,\mathbf{X}_i + b\right) \tag{B.40}$$

$$\frac{\partial^2 f}{\partial b \partial \mathbf{X}_i} = \frac{\partial^2}{\partial b \partial \mathbf{X}_i}\left[f\left(a\,\mathbf{X}_i + b\right)\right] = f''\left(a\,\mathbf{X}_i + b\right)a \tag{B.41}$$

$$\frac{\partial^3 f}{\partial a^2 \partial \mathbf{X}_i} = \frac{\partial^3}{\partial a^2 \partial \mathbf{X}_i}\left[f\left(a\,\mathbf{X}_i + b\right)\right] = f'''\left(a\,\mathbf{X}_i + b\right)a\,\mathbf{X}_i^2 + 2f''\left(a\,\mathbf{X}_i + b\right)\mathbf{X}_i \tag{B.42}$$

$$\frac{\partial^3 f}{\partial a \partial b \partial \mathbf{X}_i} = \frac{\partial^3}{\partial a^2 \partial \mathbf{X}_i} \left[f \left(a\, \mathbf{X}_i + b \right) \right] = f''' \left(a\, \mathbf{X}_i + b \right) a\, \mathbf{X}_i + f'' \left(a\, \mathbf{X}_i + b \right) \tag{B.43}$$

$$\frac{\partial^3 f}{\partial b^2 \partial \mathbf{X}_i} = \frac{\partial^3}{\partial a^2 \partial \mathbf{X}_i} \left[f \left(a\, \mathbf{X}_i + b \right) \right] = f''' \left(a\, \mathbf{X}_i + b \right) a \tag{B.44}$$

where f''' is the third derivative of f. All of these equations constitute the gradient of Newton's method. In addition to the function f and its first and second derivatives, one would also implement the third derivative for a memory-optimized CAVE function. We conclude the derivation.