Rheinisch-Westfälische Technische Hochschule Aachen
Faculty of Mathematics, Informatics, and Computer Science

Master's Thesis

Denoising Methods for Multi-Dimensional Photoemission Spectroscopy

Submitted in partial fulfillment of the requirements for the degree of
M.Sc. Data Science
with specialization in Physics

Author

Muhammad Zain Sohail
Rheinisch-Westfälische Technische Hochschule Aachen
Christian-Albrechts-Universität zu Kiel
Deutsches Elektronen-Synchrotron DESY

Supervisors

Prof. Dr. Benjamin Berkels
Rheinisch-Westfälische Technische Hochschule Aachen

Prof. Dr. Kai Rossnagel
Christian-Albrechts-Universität zu Kiel
Deutsches Elektronen-Synchrotron DESY

Aachen, November 2024

Abstract

The probabilistic nature of photoemission, combined with the exploration of large multidimensional parameter spaces–including momentum, energy, time and spin polarization–necessitates time-intensive data acquisition to ensure statistical robustness. These measurements are especially important for capturing ultrafast phenomena, where pulsed light sources, such as free-electron lasers (FELs), become indispensable due to their ability to deliver high-brightness, ultrashort X-ray pulses. However, the low repetition rates of current FEL sources significantly extend acquisition times, impeding the real-time decision-making that could otherwise enhance experimental results. Hence, to optimize experimental outcomes for the valuable beamtimes, techniques that can harness the structures and correlations within the multidimensional space are necessary to accelerate data acquisition without compromising data fidelity.

To address these challenges, we present an investigation into advanced denoising methodologies for multidimensional photoemission spectroscopy (MPES) data acquired with time-of-flight momentum microscopes. We focus on two key approaches: (1) employing BM3D with variance stabilization through the Anscombe transform in moderately noisy datasets and (2) leveraging a deep learning-based 3D UNET architecture, based on the Noise2Noise paradigm, excelling in low-count regimes where classical methods fail.

We further establish that the photoemitted electron distributions measured with SASE FELs deviate from traditional Poissonian statistics, instead following negative binomial statistics, an outcome that has implications for denoising strategies in the MPES data.

Our results demonstrate that BM3D delivers robust denoising performance for datasets with moderate average-counts (on the order of 1×10^{-2} counts per voxel). However, in extreme low-count regimes (on the order of 1×10^{-3} counts per voxel), where most conventional denoising techniques fail, the deep learning-based approach achieves exceptional denoising performance. Remarkably, we show that MPES datasets acquired in just 10 minutes using an FEL light source can, when processed with our deep learning model, reveal key features that remain indistinguishable even after hours of conventional measurement.

The findings presented have therefore the potential to streamline data acquisition at both laboratory-scale table-top setups and large-scale facilities such as FEL FLASH. By optimizing acquisition parameters, researchers can conserve valuable beamtime or extend the scope of their studies to broader parameter spaces, results that hold broader implications for related experimental techniques.

Contents

List of Acronyms

AWGN additive white Gaussian noise 12, 13, 16, 17

BM3D block-matching and 3D filtering 2, 12, 14–16, 19, 20, 26, 28–32, 34, 44, 52, 57, 59–62, 81

CNN convolutional neural network 2, 44, 45, 50, 54, 55, 60, 79

DLD delay line detector 2, 3, 7, 9–11, 21, 38–40, 75

ELT extract, load, transform 75, 77
ERM empirical risk minimization 44, 46–48, 51, 52
EUV/XUV extreme-ultraviolet 5, 6, 9, 21, 39

FEL free-electron laser viii, ix, 1–3, 5–7, 20–22, 25, 29, 36–39, 41, 42, 44, 62, 63
FLASH Free-electron LAser in Hamburg 6–8, 20, 22, 77

GMD gas monitor detector 6, 40, 42, 75

HEXTOF high energy X-ray time-of-flight 2, 7–9, 11, 20, 22, 77
HHG high harmonic generation 3, 5–7, 20–22, 38, 39, 42, 62, 63

iid independent and identically distributed 25, 45, 47, 71

MCP microchannel plate 3, 9–11
MM momentum microscope 2, 3, 7, 9, 11, 21
MS-SSIM multi-scale structural similarity index measure 20, 26, 28–34, 55, 61, 62, 74, 77, 78
MSE mean squared error 13, 19, 26, 46–48, 73, 74, 77, 78

NB negative binomial 36–38, 41–44, 62, 72, 73, 80
NLM non-local means 2, 12, 14
NN neural network 50–52, 54, 59–61

PEEM photoemission electron microscopy 7, 8, 11
PES photoemission/photoelectron spectroscopy 1–5, 7, 37–39, 45
  ARPES angle-resolved photoemission spectroscopy 3, 5, 7
  MPES multidimensional photoemission spectroscopy 1, 2, 5, 19, 20, 22, 29, 38, 44, 50, 52, 53, 62, 63
  tr-PES time-resolved photoemission spectroscopy 3, 5, 6, 10
PMF probability mass function 15, 37, 72
PP point process 37
  PPP Poisson point process 2, 36–39
PSD power spectral density 13
PSNR peak signal-to-noise ratio 26, 55, 74, 77, 78

SASE self-amplified spontaneous emission viii, 6, 20, 36–38, 40, 42
SSIM structural similarity index measure 26, 55–57, 74, 77–79

TOF time of flight 3, 7, 9, 11, 21, 22, 38

VST variance stabilization transform 12, 16, 17, 34, 42, 44

List of Symbols

E energy of the emitted electron
E_F fermi level energy

Gd/W(110) Gadolinium on Tungsten(110)
$\mathcal{R}$ generalization error, the expected error on unseen data.
Gr/Ir(111) Graphene on Iridium(111)

$\mathcal{H}$ set of functions accessible to the learner
h hypothesis of a model.

I intensity

k_x surface parallel momentum in the x-direction
k_y surface parallel momentum in the y-direction
k_z surface perpendiclar momentum component

L_2 L2 norm
ℓ loss function, a measure of prediction error.

n_{count} number of observations/electron counts
Ni/W(110) Nickel on Tungsten(110)

λ Poisson noise present in imaging processes.

σ noise level parameter for BM3D denoising.

Δt time interval
t_{tof} detector time-of-flight coordinate, that maps to E
T total observation time/acquisition time
t_{pp} pump-probe time
$\mathcal{L}$ training error/empirical risk.

w number of slices summed along an axis
WSe_2 Tungsten Diselenide
$\mathbf{w}$ weight vector of a learner.

X noisy/corrupted/incomplete image

Y latent clean (true inaccessible) image
$\hat{Y}$ denoised/restored/reconstructed image

Glossary

Noise2Noise A training paradigm where both the input and target datasets are noisy, eliminating the need for clean reference data.

beamline A path leading the photons from the particle accelerator to the experimental endstation.

beamtime The time allocated to an experiment at a synchrotron or free-electron laser (FEL) facility.

gas monitor detector A diagnostic tool to measure the intensity of a FEL pulse in a non-invasive manner. The gas inside the detector is ionized by the FEL pulse, and the resulting current can be used to detect the absolute number of photons with an accuracy of 10%.

generalization The ability of a learning based model to perform well on unseen data.

latent Latent refers to the underlying distribution of the data, which is unobserved.

microbunching Microbunches are produced by the interaction between the oscillating electrons in the undulator and the radiation that they produce (due to the oscillatory acceleration) leads to periodic longitudinal density modulation known as microbunching. The in-phase emitted radiation adds coherently, increasing intensity and enhancing microbunching.

noise The inherent fluctuations in data due to its stochastic nature.

patch A 3D region/subset of a 3D image used in the context of training and validating a deep learning model.

pulse Also known as microbunch. Each *train* (or macrobunch) contains about 500 pulses produced from the self-amplified spontaneous emission (SASE) process (see *self-amplified spontaneous emission*). These are which are used as a secondary index for the data reduction process.

self-amplified spontaneous emission is a process where the electron beam in the accelerator, when passing through an undulator, starts emitting radiation due to acceleration. The interaction between the emitted radiation and the charge distribution leads to microbunching. These microbunches emit radiation coherently, leading to the intense, coherent radition, characteristic of an FEL.

space-charge effect The space-charge effect occurs when intense light pulses generates a large electron density. This leads to the Coulomb repulsion between electrons, causing a distortion in the electric field, leading to a spread in the energy of the electrons

spontaneous emission Spontaneous emission requires no external perturbation, and is explained in quantum electrodynamics by the interaction between an atom and quantized electromagnetic field, where even with no photons in the field, there is a non-zero probability of photon emission from the atom.

training error Also known as *empirical risk*. The error between the predicted output and the target output on the training dataset measured using a loss function.

train Also known as macrobunch. A train represents a group of closely spaced electron bunches produced and accelerated by the FEL (or more generally, any accelerator). Each train is associated with a unique identifier called `trainId`, which is used as the primary index for much of the data reduction process.

1. Introduction

Scientific methodology, especially in empirical fields such as physics, fundamentally relies on observations as the basis to understand the principles of nature. The act of measurement, however, is seldom free from ambiguity. Ingeniously designed experiments, sophisticated detection schemes and controlled environments–such as ultra-low temperatures or precisely engineered detection schemes–all contribute to minimizing external noise and improving the quality of data collected. Nonetheless, as experiments push the boundaries of scale and precision, the limitations imposed by quantum mechanics, such as uncertainty and fluctuations, remain unavoidable [1]–[3].

One important area where such challenges are evident is photoemission/photoelectron spectroscopy (PES). This method is used to study the electronic structure of materials by measuring the energy (and momentum) of emitted electrons from a sample, irradiated by a light source [4], allowing to understand material properties at an atomic level. This makes PES an invaluable tool in modern material science.

Importantly, this entire process of photon absorption and electron emission is inherently probabilistic. When a material is irradiated, each photon has a certain probability of interacting with electrons in the material, and the electrons have a certain probability of being emitted, thus introducing stochastic variability in the measurements [2].

Despite this stochastic nature, experimentalists can rely on the fundamental law in probability theory, the law of large numbers[1], guaranteeing that the observed data converges to the true distribution as the number of observations increases [5]. However, while adding more data reduces *noise*, the improvement occurs at a diminishing rate. This is highlighted by the rate of convergence being proportional to the inverse square root of the number of observations[2], $\mathcal{O}(1/\sqrt{n})$. This implies that in experiments with limited time or resources, it is often impractical to collect enough data to achieve the desired precision.

Multidimensional photoemission spectroscopy (MPES) extends PES by measuring across multiple dimensions, such as momentum, time, spin, probe energy etc., allowing researchers to capture a more comprehensive view of the electronic structure and dynamics of materials. The increase in dimensionality, however, necessitates exponentially more events to fill the sample space. This becomes particularly problematic when event detection is constrained by rare phenomena, specific measurement schemes [6], space-charge effect [7], or sample degradation due to radiation damage and short timescales of transient events, constraining the data acquisition time-window. The result is a low number of acquired electron counts, insufficient to accurately estimate the *latent* distribution.

While increasing the acquisition times would reduce these fluctuations, experiments at large-scale facilities, like free-electron lasers (FELs) or synchrotrons, are often limited by strict *beam-time* allocations. Therefore, techniques that can extract the maximum information from the limited data are essential; techniques that access correlations and structures in the multidimensional space to improve the estimation of the latent distribution.

The present thesis is hence concerned with the estimation of the latent multidimensional images from incomplete observations generated by the MPES experiments, a complex problem at the intersection of experimental physics and data science. The primary focus is on developing methods to enhance the quality of noisy photoemission data; a challenge that holds significance

[1] Refer to Appendix A.1.

[2] $\mathcal{O}(g(n))$, describes an upper bound on the growth rate of a function.

for experimental physicists working with advanced light sources, such as FELs, and for data scientists interested in cutting-edge image restoration techniques. Another focus of this work is to examine the statistical properties of photoemitted electrons; an interesting study in its own right, but one that can also aid in identifying characteristics informing more effective restoration approaches. Additionally, emphasis is placed on explaining the instrumentation, as these details are critical to understand how the data is generated.

In Chapter 2, an overview of the fundamental concepts of PES is provided. The time-resolved variant of PES with a special emphasis placed on the light sources (HHG lasers and FELs), the momentum microscope (MM) setup (HEXTOF), and the segmented delay line detector (DLD) used to obtain data are discussed. This experimental scheme, being at the forefront of experimental physics, allows efficient and versatile characterization of materials. This chapter further addresses how the inherent experimental constraints posed by MPES impact the quality and quantity of data obtained, setting the stage for the image restoration techniques developed later in the thesis.

Following this introduction, in Chapter 3, the thesis moves into an exploration of the image corruption model and reviews classical image denoising techniques such as Wiener filtering and non-local means (NLM), culminating in a focus on the application of block-matching and 3D filtering (BM3D). Additionally, Poisson noise modeling is discussed, commonly assumed in event-counting experiments. The Anscombe transform, and its inversion are then introduced as techniques to stabilize the variance of noisy data, allowing the usage of Gaussian noise models and denoising techniques designed for such models.

The subsequent chapter, Chapter 4, describes the specific datasets used throughout the thesis. We look at the process by which the single-event data is transformed into multidimensional images, how noisy realizations are generated, and the metrics employed to assess image quality. Finally, we evaluate the performance of BM3D with and without the Anscombe transform on these datasets. This evaluation involves optimizing hyperparameters and then examining how denoising effectiveness varies with electron counts, allowing us to understand the denoising effectiveness in low-count scenarios.

Recognizing the limitations of the Poisson noise model and classical denoising techniques, the thesis shifts toward a statistical characterization of photoemission events in Chapter 5, particularly in the context of FEL light sources. The photoelectron emission process is described using the doubly stochastic Poisson point process, a generalization of the Poisson point process (PPP), which accounts for the non-Poissonian counting statistics of FEL light. We then explore how single-event measurements can be utilized to estimate the temporal distribution of photoemission events, providing deeper insight into the underlying data generation process.

With the complexity of high-dimensional datasets and complex counting statistics, we employ a deep learning-based approach in Chapter 6. Specifically, a 3D convolutional neural network (CNN), the UNet3D architecture, is trained to denoise MPES data. The training follows the *Noise2Noise* paradigm, utilizing multiple noisy realizations derived from the single-event data. This novel method offers a promising solution for denoising complex, multidimensional data generated in MPES experiments. Furthermore, considerable effort is made to demystify learning based models. Key concepts in deep learning and neural networks are presented in-depth, providing a clearer understanding of how these tools can be effectively applied in experimental data processing.

In summary, this thesis not only aims to develop denoising techniques and model the data-generating process for MPES data but also to serve as an introduction to both PES and the application of denoising, both classical and learning-based, in this domain. By combining these topics, it is hoped that this work will be a useful resource for both physicists and data scientists interested in these interconnected fields.

2. Photoemission Spectroscopy: Interaction, Light Sources, and Detection

Understanding electron behavior in materials is foundational to modern material science, advancing knowledge of the fundamental properties of matter and simultaneously shaping the development of technologies. The electronic structure and dynamics of materials inform about many of the properties, ranging from conductivity and magnetism to phenomena like superconductivity and topological surface states, making it possible to design new devices and materials with tailored functionalities. These insights allow progress in fields such as semiconductor development, quantum computing, and materials engineering.

We look at an important experimental technique used to study the electronic structure and dynamics: photoemission/photoelectron spectroscopy (PES). The evolution of PES has led to several specialized techniques, each offering unique insights into material properties. From conventional PES measuring density of states to angle-resolved photoemission spectroscopy (ARPES) revealing band structure and many-body effects, to time-resolved variants enabling the study of ultrafast dynamics, these methods have continuously expanded experimental capabilities. The detection of the emitted photoelectrons has similarly evolved, from hemispherical analyzers to the time of flight (TOF)-momentum microscopes (MMs), utilizing microchannel plates (MCPs) and delay line detectors (DLDs). These advanced detection schemes have enabled simultaneous measurement of electron energy, momentum, and temporal information.

As many of the intriguing dynamics such as charge carrier dynamics, phonon interactions [8]–[10], and optically induced phase transitions [11] occur on ultrafast timescales (fs to ps), the light sources must provide sufficient temporal resolution, i.e. sufficiently short pulse durations, to capture these phenomena. High harmonic generation (HHG) and free-electron laser (FEL) sources have become instrumental in this regard, offering the necessary combination of temporal resolution and photon energy tunability. However, these advanced light sources, along with time-resolved photoemission spectroscopy (tr-PES), introduce unique challenges. The high photon densities (flux) of these sources can lead to space-charge effects, where Coulomb repulsion between photoemitted electrons distorts their trajectories and energies, distorting measurements and spectral information.

This chapter provides an overview of PES, building towards the MM instrument–HEXTOF–used to capture the data presented in this thesis[1]. Throughout this discussion, we highlight the specific challenges associated with these advanced measurements, particularly those related to space-charge effects, noise, and data quality in FEL experiments, motivating the need for the denoising approaches that form the core of this thesis.

2.1 Photoemission Process

In the seminal paper by Einstein [12], that laid foundations to Quantum Mechanics, Einstein postulated that light is made of discrete quanta of energy $E = h\nu$ to explain the observations

[1]While there is a dataset we use captured using a different MM instrument, it is used for comparison, so the details are not presented.

by Hertz and J.J. Thompson, explaining the photoelectric effect. The effect can be described as

$$E_e = h\nu - \phi - |E_B| \tag{2.1}$$

where E_e the emitted kinetic energy, h is the Planck's constant, ν the frequency of the incoming photon, ϕ the material-specific work function and E_B the binding energy referenced to the Fermi level E_F.

The equation describes how incident photons on a surface emit photoelectrons, provided the photon energy $h\nu$ exceeds ϕ. It is then apparent that the binding energy of electrons can be found by irradiating light onto the material and measuring the E_e of photoelectrons. PES is exactly such a technique that leverages this principle to probe the electronic structure of materials.

Equation (2.1) also highlights that the kinetic energy of the emitted electrons E_e is dependent on the photon energy but independent of the photon flux (photons per second). However, the flux of photons does affect the number of photoelectrons emitted [13]. This effect and the probabilistic nature of the photoemission process can be described with the quantum theory of light-matter interaction[2].

The whole process of excitation, transport, and emission can be treated as a single coherent process using the formalism of quantum mechanics. This approach incorporates the electronic structure, electron-electron interactions, and the surface barrier in a unified way, and describes the photoemission intensity $I(E, k_x, k_y)$ being proportional to probability.

The process can be described by the Fermi's Golden Rule, which gives the transition rate between two states. In the context of photoemission, the initial state is the electron in the material, and the final state is the electron in the vacuum. The intensity of photoemission is described by the transition rate between the two states, and the energy conservation:

$$I(E, k_x, k_y) \propto |\langle \psi_f | \mathbf{A} \cdot \mathbf{p} | \psi_i \rangle|^2 \delta(E - E_f) \tag{2.2}$$

where ψ_i and ψ_f represent the initial and final electron wavefunctions, $\mathbf{A}\cdot\mathbf{p}$ is the matrix element that couples the photon to the electron, and the δ function ensures energy conservation between the initial and final states.

This highlights the key element that the emission process is a probabilistic event, and hence each event can be described as a random variable. With sufficient observations, the random variables approach their expected value[3], allowing the material's electronic structure to be resolved. Later in Chapter 5, we will discuss the statistical properties of these events in more detail.

In the linear regime, an increase of photon flux results in increased number of photoemitted electrons (or vise versa), but not in E_e. However, at high photon fluxes, non-linear photoemission processes start playing a more significant role. In such a case, an electron is emitted after absorbing more than one photon, leading to a deviation from the single-photon flux relation.

2.2 Spectroscopy Techniques

A variety of photoemission spectroscopy methods can be devised depending on which parameters are varied and what is measured. Naturally, the most basic setup would measure the energy of the emitted electron as described in Equation (2.1), while variations of the technique allow for additional parameters, such as resolving the electron momentum and spin, or dynamics.

[2]Many of the results can be explained by semi-classical theory as well, which treats the light as a classical wave and the electrons as quantum particles.

[3]This was briefly discussed in the introduction. The reader is also referred to Appendix A.1.

One key technique is ARPES, which simultaneously measures the kinetic energy E_e and surface parallel momentum components, k_x and k_y, of the emitted electrons, by analyzing their emission angles. By varying the photon energy, ARPES can also provide information on the surface perpendicular momentum component k_z.

tr-PES takes this process a step further by probing the dynamics of electronic states, allowing to understand transient, non-equibilrium phenomena. A pump-probe scheme is usually employed, where a pump laser drives the system out of equilibrium, and the core- or valence-electron states are probed by a probe laser. By applying a time offset (delay) between the two pulses, the dynamics of the electronic states can be studied as a function of time.

To ensure the statistical significance of time-resolved studies, the repetition rate and flux of the radiation source need to be high enough to capture rare events such as multiphoton processes, to probe excited electronic states with low population densities etc. While HHG and FELs sources are well suited for such experiments, having a high flux can be detrimental for PES due to the phenomenon known as the *space-charge effect* [7].

The space-charge effect occurs when intense light pulses generate a large density electron cloud. This leads to the Coulomb repulsion between electrons, causing a distortion in the electric field, leading to a spread in the energy of the electrons[4]. This effect is more critical in time-resolved studies, as the pump laser creates a pump induced space-charge. To mitigate this, the intensity of the light source must be attenuated, and the most efficient detection schemes must be used in combination with high repetition rate sources.

Measurement of more than three dimensions is generally referred to as multidimensional photoemission spectroscopy (MPES), such as the simultaneous measurement of energy and all three momentum components. This scheme not only provides a more comprehensive view of the electronic structure but can potentially do so in a short amount of time. Shorter acquisition times help to minimize experimental instabilities, such as beam instability and sample degradation. However, the added dimensions also increase the sample space exponentially, and depend on a correspondingly high flux from the light source to obtain statistically significant data within a reasonable timeframe.

2.3 Light Sources

As discussed in the previous section, the light source plays a critical role in both PES and tr-PES. Other than the necessary high fluence, extreme-ultraviolet (EUV/XUV) energy and high repetition rate sources are necessary to probe the material, and acquire large amount of data quickly. tr-PES further requires the source to be temporally coherent, with pulse durations on the order of fs.

2.3.1 High Harmonic Generation

The most ubiquitous of light sources–the laser–revolutionized experimental science as it enabled a vast range of phenomena to be precisely tested and observed, due to its high spatial and temporal coherence properties. A laser produces its light through a process known as stimulated emission, a process in which electrons, excited to higher energy states by an external energy source, emit photons as they return to lower energy states. These emitted photons, in turn, stimulate other electrons to release additional photons, leading to a cascade of coherent light.

To generate ultrafast, EUV/XUV light, necessary to probe electronic states in materials, HHG is a widely used technique. The HHG principle operates by converting the frequency of fs laser pulses into higher harmonic frequencies, resulting in coherent light spanning from

[4]Effectively reducing the energy and momentum resolution.

extreme ultraviolet to soft X-ray ranges. This is achieved through the non-linear interaction of an intense laser pulse with a gaseous medium, in which electrons are released, accelerated, and then collided with their parent atoms, releasing photons at harmonic multiples of the laser's original frequency. This approach enables the generation of ultrashort light pulses with energies typically ranging from $10-100\,\text{eV}$, which is well-suited for tr-PES [14]. Later in this thesis, we analyze WSe_2 experimental data obtained with a HHG source.

2.3.2 Free-Electron Laser

Particle accelerators, initially used for high energy physics experiments, which produce radiation as a byproduct of particles accelerating, found their use in spectroscopy. Soon, dedicated facilities providing extremely bright and tunable source of electromagnetic radiation emerged such as Synchrotrons and FELs. While Synchrotrons are excellent sources of light for a multitude of experiments, they have limitations in terms of the temporal coherence of the light produced. FELs are linear particle accelerators that produce a pulsed light source with a high temporal coherence, and energies to probe core and deep valence electrons. There are two modes under which FELs operate: self-amplified spontaneous emission (SASE) and seeded.

In SASE mode, the amplification process is initiated by the electron shot-noise (*spontaneous emission*) when the electron beam in the accelerator passes through an undulator. Undulators are periodically arranged magnets that cause the electrons to oscillate and emit radiation. The interaction between the emitted radiation and the charge distribution leads to a phenomenon known as *microbunching* [15]. These microbunches emit radiation coherently, leading to the intense, coherent radiation, characteristic of a FEL.

Due to the inherent stochastic nature of the process, the radiation produces intensity fluctuations and hence a lower degree of temporal coherence. Such fluctuations can be seen using a gas monitor detector (GMD) (see glossary *gas monitor detector* for details) as shown in Figure 5.3. The light produced by an FEL is of high peak brightness and can be compressed to fs pulse durations, making it an ideal source for studying ultrafast phenomena.

To improve the temporal coherence, some FELs such as FERMI [16] operate in the seeded mode, where a coherent seed laser is used to initiate the amplification process. This leads to a higher degree of temporal coherence in the light produced.

Deutsches Elektronen-Synchrotron DESY is a national research center in Germany that operates the Free-electron LAser in Hamburg (FLASH) facility [15], [17] that produces ultrashort EUV/XUV and soft X-ray radiation in the wavelength range of $4-50\,\text{nm}$ in the fundamental, and as low as $1.7\,\text{nm}$ in the third harmonic, corresponding to a total photon energy range of $25-830\,\text{eV}$. With an average pulse energy of $1-500\,\mu\text{J}$, and pulse durations $<10\,\text{fs}$, peak powers of $1-5\,\text{GW}$ can be reached. This makes it an ideal source for studying ultrafast processes in materials and molecules. An example scheme of the acceleration modules, *beamline*s and experimental end-stations can be seen in Figure 2.1, where our experiments are performed at the PG2 beamline at FLASH1 (see FLASH1 and 2 in Figure 2.1).

As can be seen on the top left of Figure 2.2, FLASH provides a very high repetition rate of $1\,\text{MHz}$ (or $1\,\mu\text{s}$ gap between *pulses*), but an effective rate of $5\,\text{kHz}$ due to there being 500 pulses in each *train*.

Efficient detection schemes are then necessary to capture all emitted electrons as not only is the repetition rate relatively low, the flux is also attenuated to be below the space-charge limit. Moreover, electron counting detectors typically have a dead-time, meaning that after a measurement, the detector is unable to detect another electron for a certain period of time. This dead-time can be longer than a pulse duration, resulting in the limitation of detecting only a single electron per pulse. Sophisticated detection schemes are then necessary to capture a higher number of emitted electrons, and reduce the acquisition time; the topic of next section.

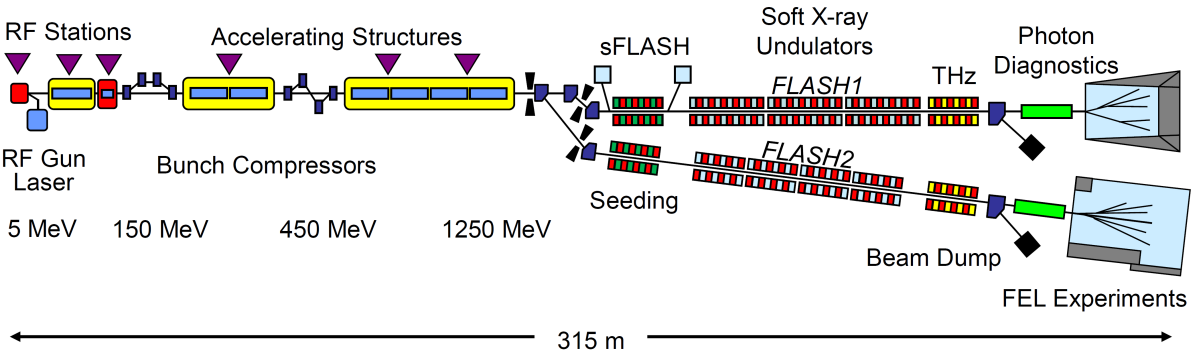


Figure 2.1: *Schematic of FLASH:* showing the accelerator section, and the two beamlines: FLASH1 and FLASH2. Reprinted from [18], under the terms of the Creative Commons Attribution 3.0 License.

2.4 HEXTOF Instrument

For PES, the hemispherical analyzer has long been the preferred instrument, primarily due to its precision in mapping energy-momentum space through an electrostatic lens system and hemispherical energy filter. The analyzer, paired with a 2D detector, can simultaneously measure the energy and azimuthal emission angle, where the emission angle can be mapped to one parallel momentum direction (hence the name ARPES). However, it captures only a narrow energy-momentum window at a time, limiting efficiency in high-throughput applications.

More recently, the MM with a TOF-tube for energy dispersion has been established [20]. This technique provides a full-field view of the photoemitted electron momentum distribution, offering simultaneous access to the entire momentum space without the need for angular scanning. This, combined with an appropriate detector, provides a 3D data set in energy (E) and surface parallel momentum (k_x, k_y), enabling efficient acquisition across the entire Brillouin zone[5]. Due to the larger field of view allowing a higher number of photoelectrons to be emitted within each pulse, the space-charge effects can be more pronounced than in hemispherical analyzers.

The MM exploits the basic concept from optics where the reciprocal of an image corresponds to its Fourier transform. In the context of PES, this means that the reciprocal image of the photoemitted electrons yields, in the back focal plane of a cathode-lens microscope, the projected band structure of the sample under investigation, due to the conservation of parallel momentum in the photoemission process.

The kinetic energies of photoelectrons are resolved through a TOF spectrometer, which consists of a field-free drift tube. In the drift tube, photoelectrons are separated in energy due to their differences in velocity and hit the detector at different times. This setup requires pulsed photon sources, such as the earlier discussed FEL and HHG sources.

MMs also include a photoemission electron microscopy (PEEM) mode, allowing real-space imaging of the sample surface by adjusting the lens system to map the spatial electron distribution instead of momenta. PEEM is useful for verifying the spatial overlap of excitation beams and facilitates precise alignment of photon beam positions and focus on the sample. An example of a PEEM image can be seen in Figure 2.3.

The HEXTOF instrument [19], shown in Figure 2.2, is based on this TOF-MM scheme to perform core- and valence-electron time- and momentum-resolved studies at FLASH, utilizing a specialized 8S-DLD for capturing electron distributions, discussed in the next section. This instrument captured majority of the data presented in this thesis, with the exception of WSe_2 data, captured using another MM instrument in combination with a HHG source [6].

[5]For a detailed comparison between hemispherical analyzer and TOF-MM, the reader is referred to [6].

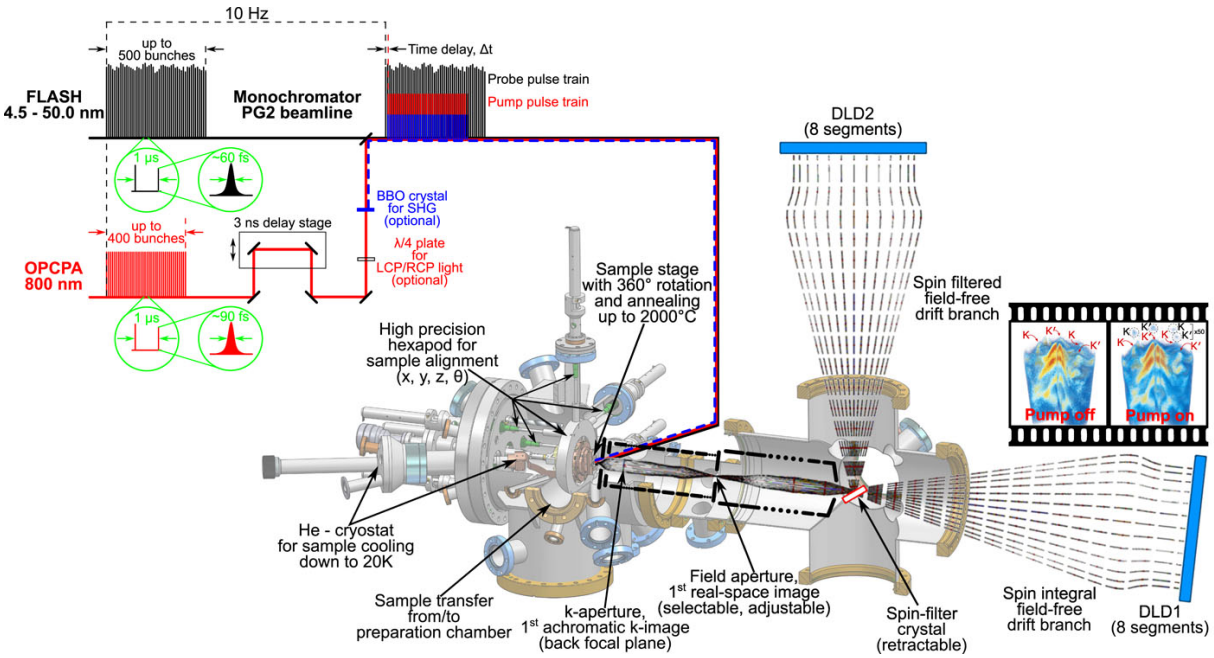


Figure 2.2: *Pulse structure, beamline and HEXTOF:* The simplified overview of the FLASH pulse structure, the PG2 beamline with synchronized pump laser (top left) and the high energy X-ray time-of-flight (HEXTOF) experimental setup (middle) used to perform time-resolved momentum microscopy. Reprinted from [19], under the terms of the Creative Commons Attribution 4.0 International License.

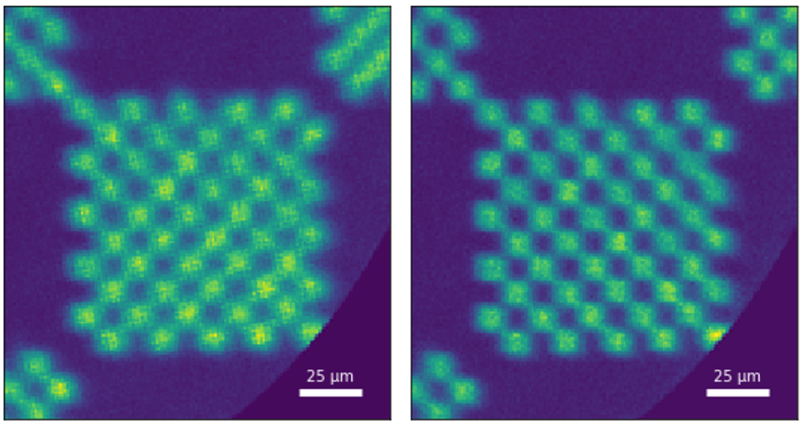


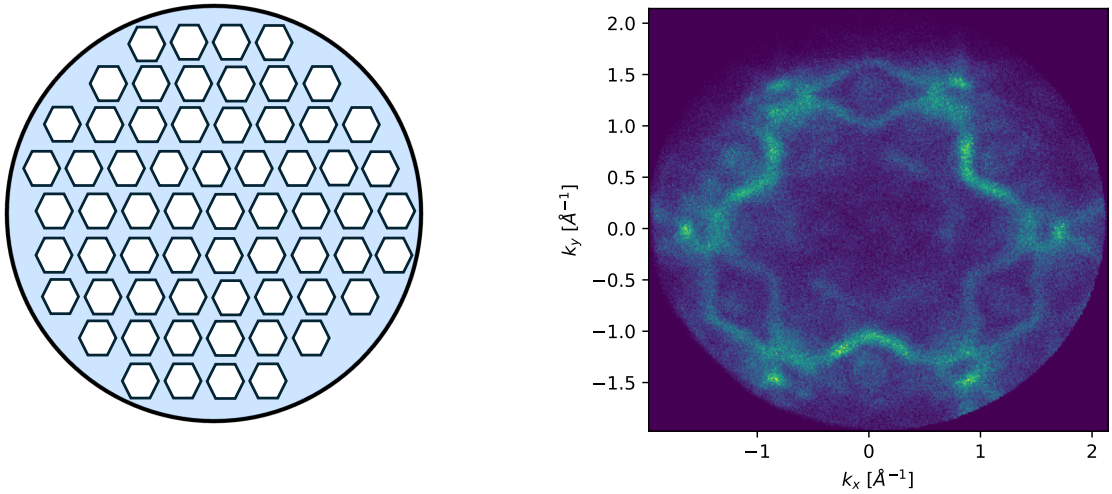
Figure 2.3: *Chessy sample before and after correction:* Merging double-counted events shows better resolved features. The Chessy test sample is employed to determine the spatial overlap and field-of-view alignment in a momentum microscope using the PEEM mode. Courtesy of M. Heber.

One of the standout features of HEXTOF is its multidimensional recording scheme. This approach allows for simultaneous measurement of multiple parameters that can be related to the complete momentum space k_x, k_y, k_z, energy E, and delay time t_{pp}. The k_x and k_y are recorded based on the direction of the emitted electrons, while the k_z is determined by tuning the photon source. The E is determined by the time-of-flight of the electrons, and the t_{pp} is the delay between the pump and probe pulses. More recently, the spin branch has been explored and spin polarization added to the list of parameters that can be measured.

2.5 Delay Line Detector

2.5.1 Microchannel Plate

Individual electrons released by the photoemission process are difficult to directly detect due to their small charge. Therefore, it is necessary to have an amplification process to allow their detection. MCP is one such amplifier, that is sensitive to electrons[6]



(a) Illustrative front view of a circular MCP showing an enlarged hexagonal microchannel structure. MCPs contain millions of such microchannels packed at a microscopic scale, each capable of amplifying electron cascades.

(b) Calibrated k_x–k_y map of Gr/Ir(111) near E_F. This map represents the momentum distribution of photoelectrons detected by the HEXTOF setup, with pixel values mapped to physical momentum axes.

Figure 2.4: *MCP and momentum map:* (a) Illustrative view of the circular MCP, showing its hexagonal microchannel structure, which plays a critical role in amplifying electron signals within the detection system. (b) Circular momentum map of the Gr/Ir(111) dataset in the k_x–k_y plane, derived from measurements using the full detection setup, which includes the MCP as part of the TOF-MM and DLD configuration. Both the MCP structure and momentum map share a circular geometry, reflecting the consistent spatial organization across the detection system.

MCPs are structured dense 2D array of hexagonal microchannels (with µm sizes), as can be seen in Figure 2.4a. With millions of such channels, each charge cloud can be localized. When an electron strikes the surface of the MCP, it generates a cascade of secondary electrons, resulting in a final charge cloud that is orders of magnitude stronger than the initial signal. For additional gain, the MCPs can be stacked in two layers, with V-stack (Chevron) configuration[7]

[6] MCPs are versatile as they are also sensitive to high energy photons in the EUV/XUV regime, and also to charged particles.

[7] Alternate configuration with an even higher gain by stacking together three layers is known as Z-stack.

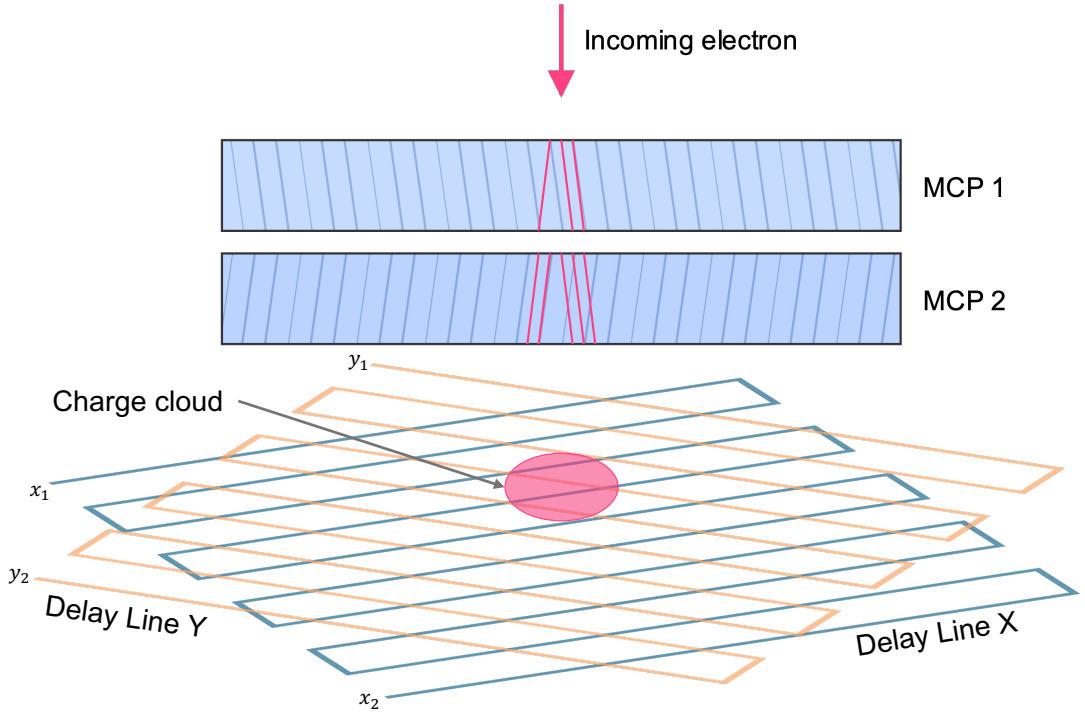


Figure 2.5: *Diagram of DLD and MCP components:* Diagram of an example DLD, with an MCP and a delayline structure. **MCP Structure:** The MCPs are arranged in a V-stack (Chevron) configuration, with microchannels at opposite tilt angles to enhance electron cascade efficiency. Each channel, shown as simplified lines within the semi-transparent blue MCP layers, initiates an electron cascade upon impact. This cascade amplifies the signal across both MCPs, preserving spatial information and forming a final charge cloud. **Delayline Structure:** Below the MCPs, the x and y meandering delay lines are presented, rotated 45 degree relative to one another, illustrating how each delay line (labeled X and Y) captures spatial information through the timing of signals between designated endpoints (x_1, x_2, y_1, y_2).

as can be seen in top of Figure 2.5, where the V shape is visible when viewing MCP 1 and MCP 2 together [21], [22].

The amplified charge cloud is then collected by a readout system, such as phosphor screens paired with a charge-coupled device (CCD), resistive anodes, wedge-and-strip anodes, and pixelated detectors. In the phosphor screen method, phosphor screen that converts the charge cloud into light, which can be detected by, e.g., a CCD camera. The phosphor–CCD scheme provides a high spatial resolution of the electrons to be recorded, but lacks the temporal resolution necessary for tr-PES, i.e., it has a much slower decay time than the dynamics being studied.

Pixelated detectors, which divide the anode into discrete segments, provide an alternative means to capture spatial information with high precision. However, the need to read each pixel individually limits their processing speed, as each pixel must be sequentially analyzed. While this approach supports multiple simultaneous hits across the pixel grid, it remains inadequate for experiments requiring rapid data acquisition and precise temporal resolution. Improvements in the readout speed have been suggested and the prospect of using pixelated detectors in future experiments is being realized [23].

2.5.2 MCP with Delay-line readout

MCPs in combination with a delay-line readout are known as DLDs [24]. DLDs have the unique ability to capture single-event data with high spatiotemporal accuracy. As shown in

Figure 2.5, a delay-line readout includes two orthogonal meandering[8] delay lines (meanders), positioned beneath the MCP, labeled Delay Line X and Y. When an electron strikes the MCP, the resulting amplified charge cloud is transmitted to the delay lines. The charge propagates in both directions along each delay line, with timing recorded at each endpoint (x_1, x_2, y_1, y_2) by a time-to-digital converter (TDC). By calculating timing differences along the x and y axes, the exact position of the electron impact is determined and by using a reference trigger, the arrival time of the electron t_{tof} can also be determined. This allows precise determination of the electron's position in both PEEM and MM modes, providing real and surface parallel momentum information, respectively.

The DLD is also able to record the electron TOF, which is synchronized to the photon pulse source. The TOF can be measured by recording the initiation of the event (when the photon pulse comes) till when they hit the meanders. Since electrons with different energies travel at different velocities, the TOF can be used to determine the kinetic energy of the electrons.

The issue with these detectors is that the meanders experience a dead-time on the ns scale after each event. This allows only some collected electrons to be detected, necessitating a higher acquisition time. An alternative to this is segmenting the detector. One way is to add more 2D meanders and assign them to different sections of the MCP (with overlaps to have no missing location), such as having the meanders in different quadrants of the MCP. This allows for the detection of multiple electrons simultaneously. Another is to layer these meanders vertically behind each other, as the meander experiencing dead-time is mostly invisible to the electrons, and the next layer can detect the electrons. Stacking up to 128 meander layers is currently under development [25].

2.5.3 HEXTOF 8S-DLD

The detector used in the HEXTOF instrument uses a combination of these approaches, with a 2-layer, 4-quadrant structure, known as the 8S-DLD. The 8S-DLD is hence capable of detecting multiple electrons simultaneously, providing a more comprehensive view of the photoelectron distribution. However, the multi-layer detection scheme comes with a caveat: when an electron cloud strikes in locations where two quadrants meet[9], special adjustments need to be made to not detect the event twice. To that end, layers must be appropriately calibrated otherwise the electron counting routines can report multiple events for the same electron, reporting different momentum and energy values for the same electron event.

With a correlation analysis on the photoelectrons generated in pulses, this effect has been observed by Heber [26]. This effect not only wrongly counts electrons but also distorts (blurs) the momentum distribution. Heber further demonstrates that by first finding the multi-counted events and averaging them, the resolution of MM can be improved as can be seen on the right of Figure 2.3, compared to the distorted distribution on the left.

For the interested reader, we also point to the work by Knipfer, Meier, Volk, *et al.* who have proposed a deep learning based method to reconstruct the events accurately [27], significantly improving reconstruction done by prior DLD codes.

[8] This zigzag pattern increases effective length, allowing for finer temporal resolution and hence position.
[9] This is always the case for 2 or more layered schemes.

3. Denoising Preliminaries and BM3D

The goal of image restoration, detached from a specific noise model, is to estimate the latent[1] clean image Y, from an incomplete or corrupted (noisy) image X where

$$Y, X \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d} \tag{3.1}$$

are both d-dimensional discrete real-valued images. The voxel values (intensity or counts) of noisy X and latent clean Y images are denoted as $x[i]$ and $y[i]$, respectively, with

$$x[i], y[i] \quad \text{for} \quad i \in \mathcal{I}. \tag{3.2}$$

where $\mathcal{I}$, the index set in the d-dimensional space, is defined as

$$\mathcal{I} = \{[i_1, i_2, \ldots, i_d]^T \mid i_1 \in [1, n_1], i_2 \in [1, n_2], \ldots, i_d \in [1, n_d]\} \tag{3.3}$$

In the most general form, the observation model can be defined as X being a mapping of Y through a function $\mathcal{F}$, which is generally stochastic. This can be written as:

$$X = \mathcal{F}(Y) \tag{3.4}$$

where $\mathcal{F}$ can vary depending on the noise model (e.g. additive or multiplicative noise, etc.), or other corruptions such as blurring, distortions etc.

One widely used model is additive noise model, where the function $\mathcal{F}(Y)$ simply adds noise N to the clean image Y. In this case, the observation model can be written as:

$$X = Y + N \tag{3.5}$$

where the common assumption for N is additive white Gaussian noise (AWGN), normal distributed $\mathcal{N}$ with mean 0 and variance σ^2

$$n[i] \sim \mathcal{N}(0, \sigma^2) \tag{3.6}$$

with $n[i]$ the noise component at each voxel i.

The inverse problem, of image restoration[2], to estimate Y from X, is an *ill-posed* problem; meaning that there are multiple possible solutions of the estimated image $\hat{Y}$ that are not unique or equal to the true image Y. This is due to the loss of information during the observation process, leading to an under-determined system. Depending on the prior knowledge posed, different estimates of the latent clean image can be recovered.

In the following chapter, we shall look at an AWGN denoising algorithm–the celebrated block-matching and 3D filtering (BM3D), introduced first by Dabov, Foi, Katkovnik, *et al.* in [28], building upon many of the classical denoising techniques such as transform domain denoising, filtering methods (such as the Wiener filter) and non-local means (NLM). We will look at a noise model for Poisson distributed data, discussing the Anscombe variance stabilization transform (VST) for stabilizing its variance, the inversion of the transform, and the BM3D algorithm for Poisson noise.

[1] *Latent* variables are unobserved and can only be inferred from the observed data.

[2] Depending on context, also referred to as reconstruction or denoising.

3.1 Image Denoising in Spatial and Transform Domains

Image restoration techniques have been explored in both the spatial and transform domains [29], [30]. For instance, the Fourier transform is commonly used to map images to the frequency domain. A simple way of denoising in this transform domain is to filter out the high frequency components that typically represent noise elements, followed by an inverse Fourier transform. Other transforms include wavelet transforms, discrete cosine transforms, and curvelet transforms.

In the spatial domain, several linear and nonlinear filtering techniques exist, utilizing different kernels to perform direct manipulation on pixel values. Linear filters, such as the mean filter, compute the weighted average of pixel values in a neighborhood and, in doing so, smooth the image but usually result in edge blurring. Another popular linear filter is the Gaussian filter, which applies a Gaussian kernel to the image to perform smoothing.

3.1.1 Wiener Filter

Wiener filtering is an estimation technique that can be used to estimate the latent clean image Y from the noisy image X [31]. It forms an optimal filter by minimizing the mean squared error (MSE) between the estimated image $\hat{Y}$ and latent clean image Y, written as:

$$h^* = \arg\min_h \mathbb{E}\left[\|Y - \hat{Y}\|^2\right] \tag{3.7}$$

This formulation is a case of risk minimization, a concept discussed in Chapter 6 in the context of learning algorithms. For the case of images with additive noise, as defined in Equation (3.5), the optimal filter h^* can be expressed in the frequency domain[3] (Fourier basis) as H(f)

$$H(f) = \frac{S_Y(f)}{S_Y(f) + S_N(f)} \tag{3.8}$$

scaling each frequency component of the observed signal based on the power spectral densities (PSDs) of Y and N, $S_Y(f)$ and $S_N(f)$, respectively. The most common Wiener filtering technique assumes the PSD of N to be constant, which is the case for AWGN. However, usage of other additive noise priors is as easy as changing the PSD of N. This filter $H(f)$ can be applied to the transformed clean image in the frequency domain, and the filtered image inverted back to the spatial domain to obtain a denoised estimate.

Notice from Equation (3.7) that the Wiener filter requires access to the clean image Y to compute $S_Y(f)$, which is generally not available. Hence, empirical Wiener filters[4], based on noisy X, were proposed by Yaroslavsky [32]. The method follows a moving window approach, where the filtering is estimated from the local statistics of the image. The filter is then applied to the central pixel of the window and inverted to estimate the clean image.

As the Fourier basis operates globally, applying Wiener filtering with this basis can introduce periodic artifacts, as global transforms may overemphasize large-scale image features and fail to capture fine local details. This limitation has motivated the development of local adaptive variants and the exploration of alternative transform domains such as wavelets, which better capture local image characteristics [29].

3.1.2 Non-Local Means

Other than linear filters, non-linear filters such as the median filter, replace each pixel with the median value of its neighbors and are much better at removing salt-and-pepper noise. Other ap-

[3]The filter can not be always be expressed in the transform domain, such as with other priors [31].
[4]This then becomes a case of empirical risk minimization.

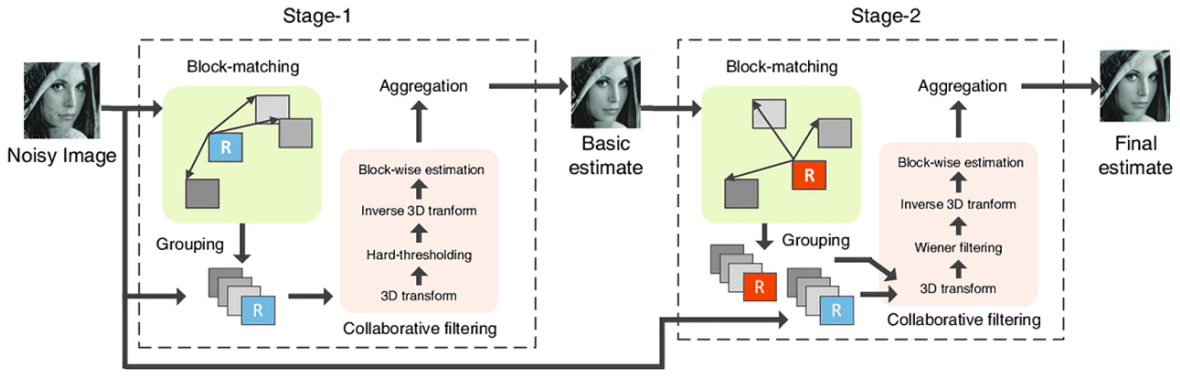


Figure 3.1: *Block diagram showing the 2-stage BM3D algorithm.* The diagram consists of block matching, collaborative filtering, and aggregation blocks, performed twice. The first stage being the basic estimate and the second stage being the final estimate. The algorithm is applied to each reference block in the image, and the final estimate is obtained by aggregating the filtered blocks. Reprinted from [33], under the terms of the Creative Commons Attribution 4.0 International License.

proaches, such as bilateral filtering, combine spatial proximity and intensity similarity, allowing for selective smoothing that preserves sharpness around edges.

The principle behind NLM denoising builds on the redundancy of similar patches in the image and estimates a pixel value by taking a weighted average over other pixels with a similar local structure, regardless of their spatial distance. By utilizing these priors, denoising algorithms can indeed use the intrinsic properties of the images and thereby enhance the visual quality and provide estimates closer to the latent clean image.

3.2 BM3D: Denoising in Sparse Domain

As shown in Algorithm 1 and Figure 3.1, the BM3D scheme works by grouping similar patches in a 2D image and applying a 3D transform[5]. This leads to an enhanced sparse representation of the image which after filtering is transformed back to the spatial domain.

Algorithm 1 BM3D Denoising Algorithm

Require: Noisy image X, noise variance σ^2
Ensure: Denoised image $\hat{Y}$

procedure BM3D(X, σ^2)
  $\hat{Y} \leftarrow X$
  for each reference block B_R in X **do**
    $B_G \leftarrow$ BlockMatching(B_R, X)
    $B_F \leftarrow$ CollaborativeFiltering(B_G, σ^2)
    Aggregate B_F into $\hat{Y}$
  end for
  return $\hat{Y}$
end procedure

In the Grouping step, candidate blocks B_i which are the least dissimilar to an $N_1 \times N_1$ ref-

[5]This algorithm has also been proposed for 3D images, dubbed BM4D [34].

erence block B_R are grouped together using the normalized l^2-distance as dissimilarity measure:

$$d(B_R, B_i) = \frac{\|B_R - B_i\|_2^2}{N_1^2}$$

with group B_G formed by selecting blocks that have distance below the threshold τ:

$$B_G = \{B_i : d(B_R, B_i) \leq \tau\}$$

The Collaborative Filtering[6] shown in Algorithm 2 is then applied to the grouped blocks. This step consists of a 3D transform such as the discrete cosine transform (or the wavelet transform can be used). A filter is applied to the transformed blocks to remove noise, initially by hard thresholding and in the second run by Wiener filtering. The inverse 3D transform is then applied to the filtered blocks, and the filtered blocks are aggregated to form the estimate. The first run is considered the basic estimate, and it is only after Collaborative Wiener filtering that the final estimate is obtained.

Algorithm 2 Collaborative Filtering

Require: Group of similar blocks B_G, noise variance σ^2
Ensure: Filtered block B_F
1: **procedure** CollaborativeFiltering(B_G, σ^2)
2: $B_T \leftarrow$ 3DTransform(B_G)
3: $B_F \leftarrow$ ApplyFiltering(B_T, σ^2)
4: $B_I \leftarrow$ Inverse3DTransform(B_F)
5: Aggregate B_I into B_F
6: **return** B_F
7: **end procedure**

The BM3D algorithm has shown one of the best denoising performances and can only be contested by the recent deep-learning based denoising methods.

3.3 Poisson Noise and Variance Stabilization

In imaging systems, the observed intensity $x[i]$ at each voxel i of a noisy image X is a stochastic mapping of the latent distribution $y[i]$ due to the measurement process. In low light settings, such as photon-limited imaging, $x[i]$ can be modeled as independent Poisson characterized by the true intensity $y[i]$, we aim to estimate[7] [37], [38]

$$x[i] \sim \text{Poi}(y[i])$$

where $\text{Poi}(y[i])$ denotes the Poisson distribution with parameter (and latent intensity) $y[i]$.

To simplify notation moving forward, we denote the observed value as x and the underlying intensity as y. Using this, we can express the probability mass function (PMF) of x given y as:

$$P(x = k|y) = \frac{y^k e^{-y}}{k!}, \quad k \in \mathbb{N}_0 \tag{3.9}$$

Poisson distribution has the key property that the mean and variance are equal, $\mathbb{E}[x] = \text{Var}[x] = y$, with standard deviation $\sqrt{y}$. Poisson noise can then be defined as the deviations from the expected value i.e. difference between the observed intensity x and the true intensity y:

$$n = x - \mathbb{E}[x] = x - y \tag{3.10}$$

[6]Interestingly, collaborative filtering has been the backbone of recommendation systems such as by Netflix and Spotify [35], [36].

[7]This assumption is discussed in more detail in Section 5.1, and is true for a constant intensity light source.

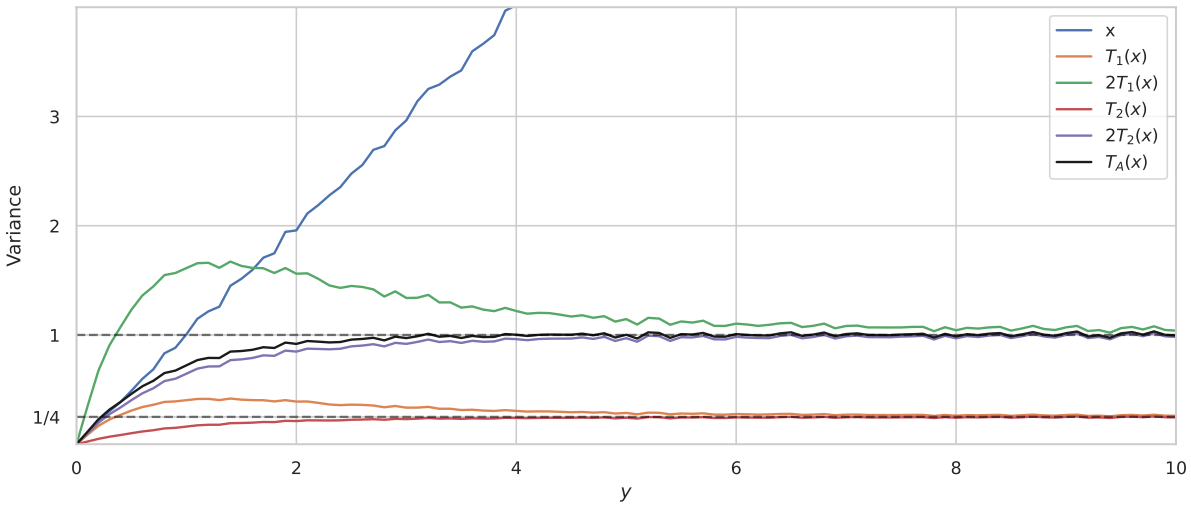


Figure 3.2: *Variance stabilization through various transformations of Poisson-distributed data.* The plot shows the variance of Poisson variables (blue) along with the variances of their transformed values under different variance-stabilizing transformations: the square root transform ($T_1(x)$, scaled $2T_1(x)$), the square root transform with a constant of $\frac{1}{2}$ ($T_2(x)$, scaled $2T_2(x)$), and the Anscombe transform ($T_A(x)$). As the Poisson parameter y increases, the variance of the scaled transforms stabilizes to approximately 1, with the fastest convergence using the Anscombe transform. The scaling by 2 allows the data to more closely approximate a standard normal distribution. Dashed horizontal lines indicate the stabilization levels.

giving us the additive noise model $x = y + n$, with noise being zero-mean since

$$\mathbb{E}[n] = \mathbb{E}[x] - \mathbb{E}[y] = y - y = 0 \tag{3.11}$$

In contrast to the AWGN model, which assumes a variance independent of the true intensity, the Poisson noise model has variance scaling with the true intensity[8] $\mathrm{Var}[n] = y$. This implies that with decreasing intensity, the fluctuations (standard deviation) increase. Consequently, AWGN denoising algorithms, such as BM3D, can not directly be applied to Poisson noise.

3.3.1 Variance Stabilizing Transformations

To address this issue, suitable transformations that stabilize the variance, making it largely independent of the mean (similar to the normal distribution) can be beneficial. Transforms such as these are known as variance stabilization transforms (VSTs). Bartlett [39] first showed such a transform: the square root transformation T_1:

$$T_1 : [0, \infty) \to [0, \infty), \quad x \to \sqrt{x} \tag{3.12}$$

where we look at a Poisson variable

$$x \sim \mathrm{Poi}(y) \tag{3.13}$$

Applying this transform to Poisson data makes the transformed values approximately normally distributed as the Poisson parameter y increases. The variance of Poisson variable x (blue) and transformed variable using T_1 (orange) as a function of y can be seen in Figure 3.2, with the variance of $\sqrt{x}$ stabilizing to $\frac{1}{4}$ with increasing y. Bartlett additionally showed that by adding a constant of $\frac{1}{2}$ to the square root transform, the convergence to normality improves, leading to the transformation:

$$T_2 : [0, \infty) \to [0, \infty), \quad x \to \sqrt{x + \frac{1}{2}} \tag{3.14}$$

[8] $\mathrm{Var}[n] = \mathrm{Var}[x] = y$ since $\mathrm{Var}[y] = 0$ due to y being a fixed parameter (deterministic).

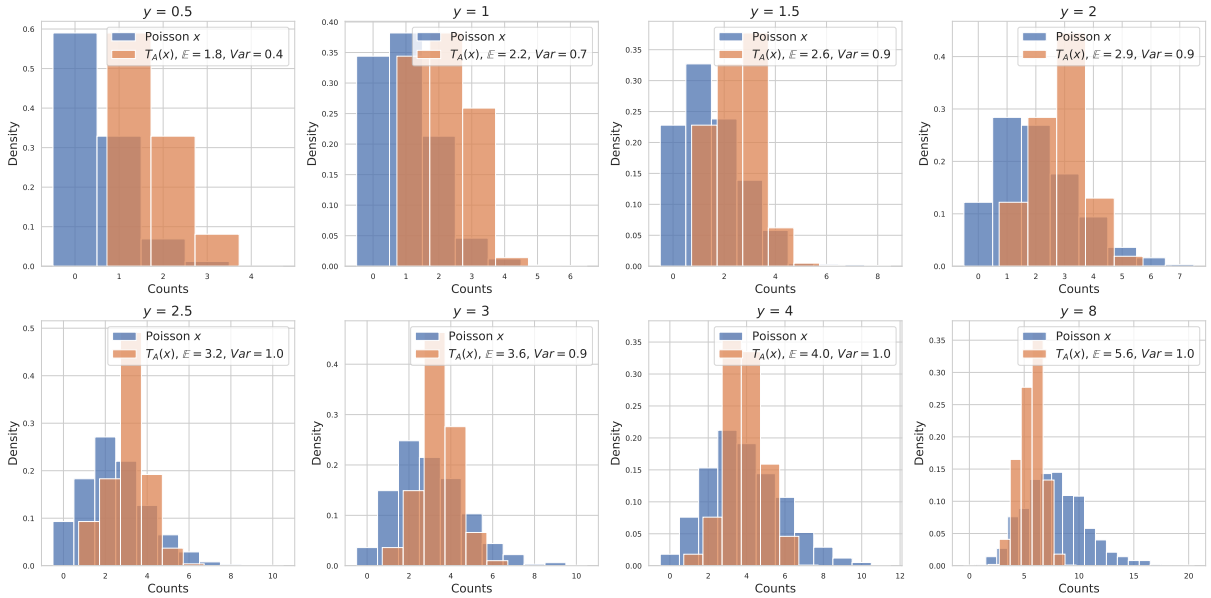


Figure 3.3: *Variance stabilization through the Anscombe transform.* The histogram shows the empirical distribution of Poisson variables with different y values (0.5, 1, 1.5, 2, 2.5, 3, 4 and 8), both before (x) and after applying the Anscombe transform ($T_A(x)$). The expected value $\mathbb{E}[T_A(x)]$ and variance $\mathrm{Var}[T_A(x)]$ of the transformed data are shown in the legend, with the variance stabilizing to 1 as y increases. It can also be seen that expected value presents a bias after the transformation that should be corrected when applying the inverse transform.

T_2 (red) in Figure 3.2 shows this faster convergence (also preventing overshoot) when compared to the T_1 (orange). T_1 (purple) and T_2 (green) are also shown scaled by 2, effectively centering the variance[9] around 1, yielding a transformed variable that approximates a standard normal distribution.

Anscombe [40] further showed that using the constant of $\frac{3}{8}$ is an optimal VST T_A for Poisson distributed data:

$$T_A : [0, \infty) \to [0, \infty), \quad x \to 2\sqrt{x + \frac{3}{8}} \tag{3.15}$$

Comparing scaled T_2 (purple) with T_A (black) in Figure 3.2, it can be seen that the variance of the Anscombe transform stabilizes to 1 faster, while not significantly different from the T_2. Figure 3.3 provides an alternate visualization showing the distributions at different y before and after applying T_A. As y increases, the variance of the transformed data $Var[T_A(x)]$ stabilizes to 1, along with a bias in the expected value $\mathbb{E}[T_A(x)]$.

3.3.2 Inverse Transformations

The Anscombe transform T_A can hence be applied to a noisy pixel x (described as $x \sim \mathrm{Poi}(y)$) to stabilize the variance, making it suitable for denoising using AWGN denoising algorithms. After applying T_A and denoising, we obtain an estimate d that approximates $\mathbb{E}[T_A(x) \mid y]$. The challenge lies in finding an appropriate inverse transformation to recover an estimate of the latent intensity y.

To this end, consider the algebraic inverse of T_A, written as:

$$I_A : [0, \infty) \to [0, \infty), \quad d \to \left(\frac{d}{2}\right)^2 - \frac{3}{8} \tag{3.16}$$

[9]The variance scales by square of the scaling factor, i.e., $\mathrm{Var}[aZ] = a^2\mathrm{Var}[x]$

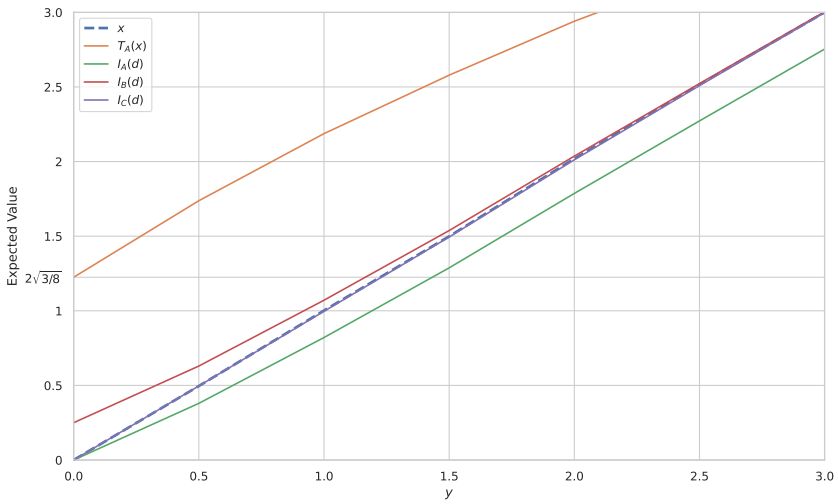


Figure 3.4: *Different inversions of Anscombe transformed data to estimate the true value.* This plot illustrates the estimates of y as a function of the Poisson parameter y. The estimates for y are obtained by inverting the Anscombe transform using three different methods: the algebraic inverse (I_A, green), the asymptotically unbiased inverse (I_B, red), and the exact unbiased inverse (I_C, purple). The expected value of x (dashed blue), $\mathbb{E}[x] = y$ and Anscombe transform $T_A(x)$ (orange) are also shown for reference.

Due to the non-linearity of the Anscombe transform:

$$\mathbb{E}[T_A(x) \mid y] \neq T_A(\mathbb{E}[x \mid y])$$

and therefore

$$I_A(\mathbb{E}[T_A(x) \mid y]) \neq \mathbb{E}[x \mid y] = y$$

meaning that I_A provides a biased estimate of y, as illustrated in Figure 3.4[10]. In this figure, $I_A(d)$ (green line) diverges from the mean $\mathbb{E}[x]$ (dashed blue line) for all values of y.

To reduce this bias, Anscombe [40] proposed an inversion I_B that is asymptotically unbiased:

$$I_B : [0, \infty) \to [0, \infty), \quad d \to \left(\frac{d}{2}\right)^2 - \frac{1}{8} \tag{3.17}$$

The asymptotic unbiasedness of I_B can also be seen in Figure 3.4 (red line), as when $y \geq 3$ (high-counts), the expected value of $I_B(d)$ is close to $y = \mathbb{E}[x]$. However, for low-counts, the bias is even higher than the algebraic inverse.

To address these limitations, Makitalo and Foi introduced an exact unbiased inverse, meaning it produces a mapping I_C that satisfies:

$$I_C : \mathbb{E}[T_A(x) \mid y] \to \mathbb{E}[x \mid y] = y \tag{3.18}$$

Such an exact unbiased inverse I_C can be constructed by computing the expected value of the Anscombe-transformed variable, $\mathbb{E}[T_A(x) \mid y]$, reflecting the expected value of $T_A(x)$, given by

$$\mathbb{E}[T_A(x) \mid y] = \sum_{x=0}^{\infty} T_A(x) \cdot P(x \mid y), \tag{3.19}$$

where $T_A(x)$ is defined in Equation (3.15), and $P(x \mid y)$ in Equation (3.9), so we substitute this into the expectation:

$$\mathbb{E}[T_A(x) \mid y] = \sum_{x=0}^{\infty} 2\sqrt{x + \frac{3}{8}} \cdot y^x e^{-y} \frac{1}{x!}$$

[10]Note that this Figure is similar to Figure 1 in [37], where they present the estimate of y against the transformed and denoised variable d. The author finds that plotting against the Poisson parameter y is simpler to understand.

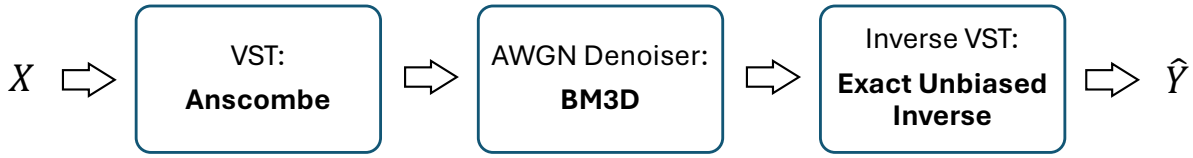


Figure 3.5: *Denoising scheme for Poisson images:* Block diagram showing the denoising scheme for Poisson corrupted images, consisting of the Anscombe transform, BM3D denoising, and the inversion of the Anscombe transform.

This expectation sums over all possible values of x, weighting each transformed value $T_A(x)$ by its probability under the Poisson distribution with mean y. Calculating this expectation accurately allows the exact unbiased inverse I_C to achieve its goal: matching the expected value of the transformed variable $T_A(x)$ back to the original mean, y. I_C can be computed using a numerical optimization algorithm, as detailed in [37].

The authors additionally found that the maximum likelihood inversion[11] coincides with I_C. This method is also shown to perform better than the asymptotically unbiased inverse I_B and the algebraic inverse I_A [37], and is hence the inversion we use in the rest of this work. A closed form approximation of I_C is also available in [41], written as:

$$I_D(d) = \frac{1}{4}d^2 + \frac{\sqrt{\frac{3}{2}}}{4}d^{-1} - \frac{11}{8}d^{-2} + \frac{5\sqrt{\frac{3}{2}}}{8}d^{-3} - \frac{1}{8} \tag{3.20}$$

3.3.3 The Anscombe BM3D Algorithm

Poisson corrupted 2D images can hence be denoised using a three-step scheme shown in Figure 3.5; applying the Anscombe transform to noisy image X, denoising the transformed image using the BM3D algorithm, and inverting the transformation to obtain estimate $\hat{Y}$ of the latent clean image Y. We use the Anscombe transform T_A and the exact unbiased inversion I_C for the pixel-wise transformations in the image. An algorithmic scheme can be seen in Algorithm 3, where we assume that T_A and I_C can take an image as an input.

This method has been shown to perform better than methods based on explicit Poisson noise removal [37]. Therefore, in the next chapter we make use of the scheme described in Figure 3.5 to denoise 2D images from MPES datasets.

Algorithm 3 Algorithm to Denoise Poisson Corrupted Images

Require: Noisy image X
Ensure: Denoised image $\hat{Y}$

procedure AnscombeBM3D(X, σ^2)
    $X_\mathrm{A} \leftarrow T_A(X)$
    $D \leftarrow \mathrm{BM3D}(X_A, \sigma^2)$
    $\hat{Y} \leftarrow I_C(D)$
    return $\hat{Y}$
end procedure

[11] They also find a minimum MSE inversion that coincides with I_C when the denoising is successful i.e. the true mean is recovered.

4. From Raw Data to Denoised Images

In this chapter, we attempt the task of denoising MPES data, beginning with an exploration of the various MPES datasets used in this study, which include Gr/Ir(111), Gd/W(110) and WSe_2, and examinining the characteristics relevant to analyzing the denoising performance.

Given its high electron counts, the Gr/Ir(111) dataset serves as our primary reference, providing a robust target for generating higher quality (compared to other datasets) target images. This dataset forms the basis for training a deep learning model , while Gd/W(110) is used to test the generalization performance (Chapter 6). WSe_2, captured with a different experimental setup, serves as a key comparator to evaluate the generalization of statistical differences due to the light sources (Chapter 5).

An overview of the data reduction and image formation pipeline is provided, illustrating how single-event data from these experiments are transformed into multidimensional images suitable for analysis. We detail how the target and noisy realizations of the data are constructed, generating representations for high-count reference images and low-count noisy images that reflect realistic experimental conditions, respectively.

In order to evaluate the performance of denoising algorithms on MPES data, we present here the criteria and metrics used; in particular, focusing on perceptual metrics like the multi-scale structural similarity index measure (MS-SSIM), as defined in Equation (A.11), which is more robust to the inherent noise in experimental reference images.

The BM3D algorithm is applied with and without the variance stabilizing Anscombe transform, to the Gr/Ir(111) dataset, containing the highest total electron counts (n_{count}). To find the optimal denoising parameters, particularly the choice of the noise level σ value, a hyperparameter search is executed over various n_{count}. The denoising performance is evaluated using this high-count dataset, providing a reliable basis for assessment. Following this, we study the performance of the two methods in enhancing image quality at various count levels.

4.1 Experimental Datasets

Dataset	Photon Energy [eV]	Temperature [K]	n_{count}	T	Light Source
Gd/W(110)	36.3	106	2.21×10^7	4.3 h	SASE FEL
Gr/Ir(111)	141.7	300	1.86×10^8	30.8 h	SASE FEL
WSe_2	21.7	300	1.00×10^9	87.7 s	HHG

Table 4.1: *Summary of properties for the three datasets.* The acquisition time T varies significantly based on the photon source and experimental setup. n_{count} are expressed in scientific notation for consistency.

We investigate two datasets–Gr/Ir(111) [42], and Gd/W(110) [43]–that have been studied using HEXTOF at FLASH. Interpretable results from these experiments require long acquisition times (T), essential for capturing the underlying physical processes. This is primarily due to three reasons:

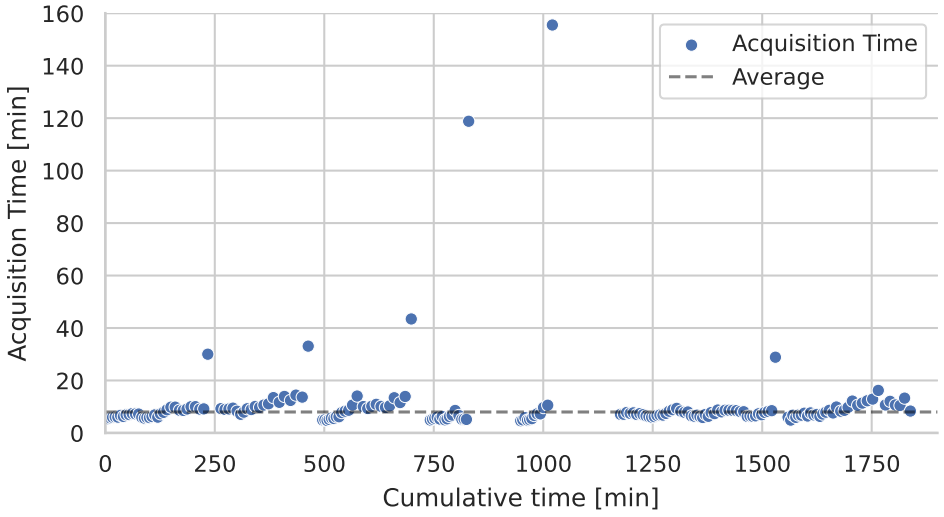


Figure 4.1: *Acquisition time for* 1×10^6 *counts:* Total acquisition time T in minutes for generating 1×10^6 data points. Some outliers are present due to the photon source being off, leading to longer T. The average T without outliers is approximately 8 min (see dashed line).

The intrinsic stochastic nature of the photoemission process requires the collection of a large number of events to approach the true value (Section 2.1). Additionally, the low repetition rate and intentionally reduced flux[1] of the FEL means that the data needs to be collected over a long period of time to get sufficient statistics (Section 2.3.2). For example, as shown in Figure 4.1, acquiring 1×10^6 electron events (n_{count}) requires an acquisition time of $T = 8$ min. Lastly, the multidimensional acquisition scheme to simultaneously resolve k_x, k_y, k_z, E, t_{pp}, and spin polarization etc. necessitates exponentially more data, as the number of dimensions increases, to adequately fill the sample space in higher dimensions (Section 2.2).

For most of this study, we use the Gr/Ir(111) dataset, as it features the longest acquisition time $T = 30.8$ h, and hence the highest number of counts, with $n_{\text{count}} = 1.86 \times 10^8$. The Gd/W(110) dataset has comparatively lower total counts, at n_{count} of 2.21×10^7, and a shorter acquisition time of $T = 4.3$ h. Other experimental details of the datasets are summarized in Table 4.1.

However, n_{count} is not the only factor in determining data quality. Material-specific factors, sample conditions, and instrument settings can influence spectral clarity, meaning that a higher-count dataset does not necessarily guarantee more pronounced spectral features. An example k_x-k_y image for both datasets is shown in Figure 4.2.

For reference, we also look at the WSe_2 dataset [44], measured with a pulsed HHG-based EUV/XUV source using a TOF-MM analyzer[2] and a single-segmented DLD. This setup yields a significantly larger number of counts, recording $n_{\text{count}} = 1 \times 10^9$ within $T = 87.7$ s. In comparison, the Gr/Ir(111) dataset, collected with a FEL source, yields $n_{\text{count}} = 1 \times 10^6$ over $T = 8$ min, thus producing about three orders of magnitude fewer events.

A direct denoising comparison between these datasets, however, is not feasible due to fundamental differences in light sources and detector design. The WSe_2 dataset was acquired with a single-segmented detector setup, whereas the Gr/Ir(111) dataset employed a more complex 8-segment detector (Section 2.5.3) that offers enhanced multi-hit capability through overlapping segments, but impacting count statistics. This already implies that the finding equivalent noise levels between the datasets is a non-trivial task.

[1] To mitigate the *space-charge effect*.
[2] The TOF-MM analyzer being the common aspect between the Gr/Ir(111) etc./ and the WSe_2 dataset.

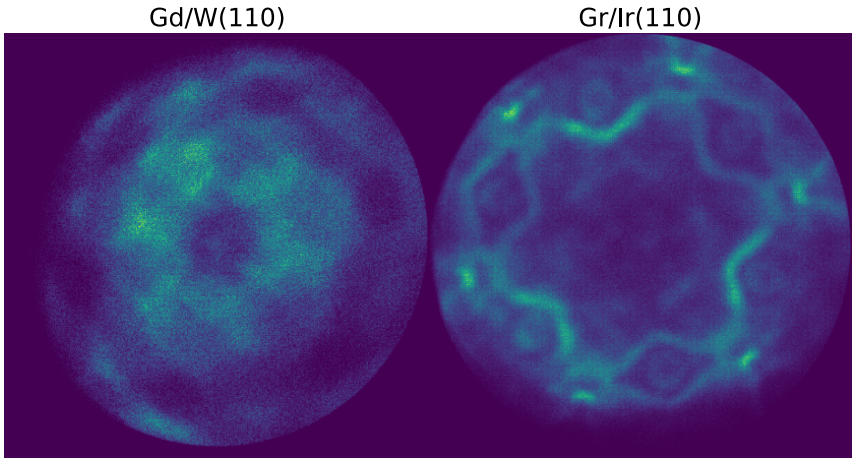


Figure 4.2: *Momentum image:* 2D k_x-k_y slices of the Gr/Ir(111) and Gd/W(110) datasets, slice-summed across E with size $w = 20$. See Section 4.2 for details on how these images are formed.

Additionally, the inherent statistical properties of the light sources (Section 2.3) impact the noise characteristics in the raw data. Consequently, in Chapter 5, the WSe_2 dataset aids in understanding statistical differences between acquisition setups. However, direct denoising comparisons between HHG and FEL sources would not be meaningful due to these varied conditions and detector architectures.

4.2 Data Processing and Image Formation

In an MPES experiment using a TOF-based scheme, the workflow to resolve images typically follows a series of steps, outlined in Figure 4.3. Prior to this, an essential step is data reduction. For the specific case of HEXTOF instrument at FLASH, used in our studies, a detailed description of the reduction process is outlined in Appendix B.1.

After the data reduction, corrections are applied to the measurement axes , e.g., to correct space-charge distortions (see Section 2.2), correct timing jitter between FEL and pump laser etc. The corrected axes are then mapped (calibrated) to the physical axes, after which further correction steps can happen. Once calibrated, the single-event data is binned into a multidimensional volume[3] that represents the full measurement, as illustrated on the right side of Figure 4.3.

4.2.1 Constructing Images from Single-Event Data

In Equation (3.1), we defined a d-dimensional latent clean image Y (and noisy image X). For most of the thesis, we focus on images with $d \in 2, 3$, as most denoising algorithms are designed to work with 2D or 3D images. Moreover, since even 3D images are commonly visualized as a series of 2D images, it is an intuitive place to begin.

Let us hence look at how such a 3D image is constructed from the single-event data. As shown in Figure 4.4, the example single-event data is stored in a table format, with each row representing a single electron-event. Of physical interest are x, y, t_{tof} and t_{delay} that map to the physical axes k_x, k_y, E and t_{pp}, respectively, where the other columns are example diagnostic and timing quantities, although they represent only a subset of the measurable quantities and diagnostics available in the experiment.

An example 2D image can be formed by selecting t_{tof} and t_{delay} columns and binning across these dimensions, with the image showing the dynamical energy response, or with x and y

[3]We make use of the Single Event DataFrame (SED) library https://github.com/OpenCOMPES/sed.

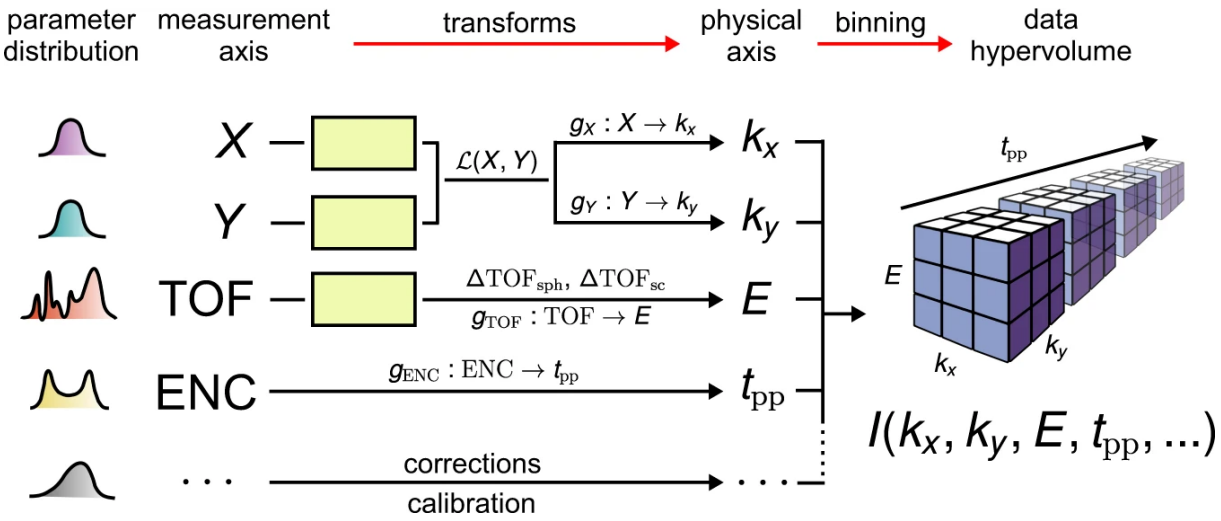
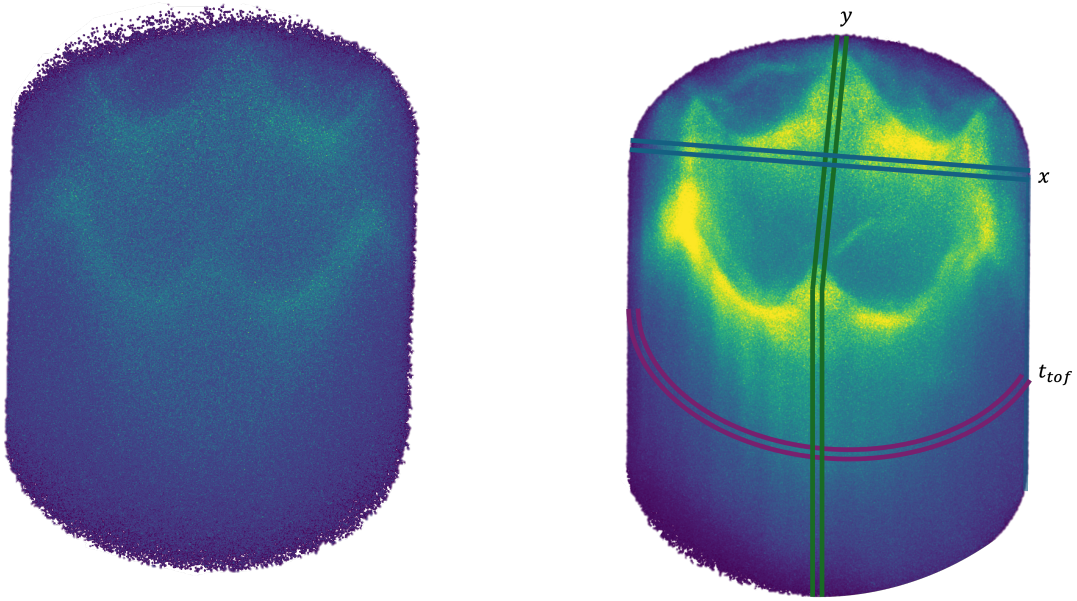


Figure 4.3: *MPES experiment workflow:* Typical workflow that comprises transformations to the measured data and binning to form the multidimensional image. Reprinted from [45], under the terms of the Creative Commons Attribution 4.0 International License.

			x	y	t_{tof}	t_{delay}	sectorID	GMD	timeStamp
trainId	pulseId	electronId							
1648730078	**7**	**0**	715.0	560.0	5249	1462.304199	1	0.633716	1.679634e+09
		1	710.0	564.0	5249	1462.304199	0	0.633716	1.679634e+09
	38	**0**	684.0	599.0	4804	1462.304199	1	0.603878	1.679634e+09
	69	**0**	543.0	536.0	4687	1462.304199	4	0.874713	1.679634e+09
		1	544.0	536.0	4686	1462.304199	1	0.874713	1.679634e+09
...	**...**	**...**	...	...	...	...	...	...	...
1648885745	**485**	**1**	684.0	571.0	4317	1463.997437	0	0.771876	1.679650e+09
		2	665.0	577.0	4696	1463.997437	1	0.771876	1.679650e+09
		3	661.0	581.0	4696	1463.997437	4	0.771876	1.679650e+09
	486	**0**	544.0	578.0	4846	1463.997437	5	0.735766	1.679650e+09
		1	543.0	582.0	4847	1463.997437	4	0.735766	1.679650e+09

Figure 4.4: *Single-event data table:* displaying the first and last five rows of a recorded dataset. The table includes the detector dimensions x, y, and t_{tof}, which represent the spatial and time-of-flight coordinates for each detected electron (resolved at the electronId level). It also shows (unadjusted with respect to t_0) pump-probe delay stage values t_{delay} (resolved per train), the sectorID (a unique identifier for each detector segment), the *gas monitor detector* (GMD) readings (averaged per pulse), and (UNIX style) timeStamps (accurate to each train). Data is hierarchically indexed by trainId (*train*), pulseId (*pulse*), and electronId (event number within each pulse). Non-resolved values, such as those linked only to trains or pulses, are forward-filled for consistency across rows, allowing easy alignment with per-event data. To optimize binning and computations, certain columns can be selectively dropped to reduce dimensionality, such as when focusing on specific features like static spectra (e.g., k_x–k_y–E) or dynamic response data (e.g., energy E vs. delay t_{delay}). Additionally, rows can be filtered to include only selected ranges, such as a particular energy interval, before the binning step.



(a) $n_{\text{count}} = 8 \times 10^6$ corresponding to $T = 8\,\text{min}$. The fine features of the data are not visible due to the low n_{count}.

(b) $n_{\text{count}} = 1.86 \times 10^8$ corresponding to $T = 30\,\text{hour}$. 2D images can be formed by slicing (or summing slices) along any of the axes, as depicted by the three lines.

Figure 4.5: *3D imaging of GrIr dataset:* (left) n_{count} of 8×10^6 and (right) 1.86×10^8. The image is constructed by binning over the three detector axes x, y, t_{tof}, corresponding to the physical axes k_x, k_y, E. 2D images can be formed by slicing (or summing slices) along any of the axes, as depicted by the three lines (blue, green, purple) showing arbitrary summing sizes.

columns, forming a momentum image. By selecting three columns such as x, y and t_{tof}, we form a 3D image as shown in Figure 4.5.

Since the detector can capture a broad dynamic range of quantities, spanning extensive energy and momentum ranges, filtering[4] is crucial to generate meaningful images. For instance, the detector records a wide range of energies, but not all are relevant to the analysis; typically, we focus on energies close to the Fermi level.

Binning these three filtered axes forms a 3D x–y–t_{tof} image. Figure 4.5 shows such images from the Gr/Ir(111) dataset, with filtered energy values to be near the Fermi level E_F. Since the t_{delay} axis is dropped, the dynamics are summed and this therefore, this image forms only an approximation of an image from a static momentum and energy resolved study. Nonetheless, since the dynamic effects are orders of magnitude smaller, this approximation suffices.

As physical interpretation is the goal, the general scheme is to calibrate the axes to the physical quantities first, and then bin the data to form the multidimensional image. However, in this study, we do not calibrate the data and use the detector quantization as the binning resolution, as calibrating can be non-linear in dimensions such as energy. Nonetheless, we will generally refer to the physical axes as they are conventionally used in the domain, even though we use the detector axes for binning.

Where the spatial dimensions (x and y) are linearly mapped to the physical axes (k_x, k_y), the E axis has a non-linear (quadratic) scaling with t_{tof}, making the t_{tof} bins non-uniform in size when mapped to the E axis. The non-linear scaling can potentially change the noise-profile of the data, which has potential to impact denoising performance. It is yet to be investigated if this impact has a positive or negative output, and is a potential avenue for future work.

In the aforementioned 3D images, spatial dimensions (x and y directions) are binned with a

[4]The range can also be reduced post binning.

resolution of 480 px, corresponding to the native detector quantization. The t_{tof} is also binned with 480 steps to match the spatial resolution[5].

It is often useful to bin the data at a coarser granularity than the detector's native resolution, mainly to address low-count statistics, producing a representation that has lower resolution but higher count statistics[6]. Coarse binning can be applied to a single dimension or across multiple dimensions. For instance, Figure 4.2 showed 2D k_x–k_y images formed by summing 20 slices along the E axis of the 3D image. This is effectively the same as coarsely binning the E axis, and selecting a single slice from this coarser 3D image. Throughout this text, we use the native detector resolution as the baseline, and for coarser analyses, we specify the slice-summing size, w, to define the summing along a certain dimension.

While the full dataset contains 1.86×10^8 electron counts, the selected energy region used to generate the image in Figure 4.5b contains 1.15×10^8 counts. Note that as discussed in Section 2.5.3, the majority of events are counted multiple times, with the most common case being that each event is recorded twice. This suggests that the effective number of counts is approximately half of 1.15×10^8, i.e. 5.75×10^7. For consistency, n_{count} is always referring to the total dataset count (1.86×10^8 in this case), even if the actual unique counts may be lower due to multiple counting of events, or due to filtering of the energy range. This reason and a difference in detector dimensions is key reason for the difficulty in comparison of Gr/Ir(111) dataset with WSe_2, which is recorded with another light source–detector setup.

4.2.2 Generation of Noisy Realizations

In typical detection setups, generating noisy realizations X of clean images Y is limited by fixed integration windows. This means that the noise levels can only be varied in discrete steps, as multiples of the integration time, or by simulating noise post-capture. In contrast, the single-event stream enables us to obtain real and adjustable noisy realizations by selecting subsets of the varying electron counts n_{count} from the full dataset, without relying on simulated noise. This flexibility, by controlling the noise levels, has a major advantage in training and evaluating denoising algorithms on a large scale of noise levels.

The noisy image X can be generated by taking a n_{count} subset of the complete dataset and then binned to form independent 3D images. For example, taking non-overlapping subsets of n_{count} being 1×10^6 from the Gr/Ir(111) dataset, we can generate 186 noisy realizations. The images can then be assumed to be formed with an independent and identically distributed (iid) stochastic process, as the underlying data generation process is assumed iid (Section 5.1). This assumption is discussed in more detail in Chapter 6 as it has important implications for training machine learning models. It should be noted that even for non-overlapping subsets, this assumption is only valid for short time scales, as the FEL light source is generally not stable over extended acquisition periods, and other factors such as sample degradation, temperature deviations, etc., could break this assumption.

To evaluate denoising performance across different noise levels, we generate a series of noisy realizations from the Gr/Ir(111) dataset, each with a distinct total count, n_{count}. The total counts and average counts per voxel and acquisition time T for each n_{count} is shown in Table 4.2. These realizations allow systematic assessment of the denoising algorithm performance under varying noise levels. Lower count realizations ($n_{\text{count}} < 1 \times 10^7$), corresponding to shorter T, are of most interest to denoise. Successful denoising of such data (such as image seen in Figure 4.5a) could significantly improve experimental efficiency, allowing the investigators to steer the experiment in the right direction, crucial in time-limited *beamtime*s.

To balance coverage across a wide range of n_{count} from $1 \times 10^6 - 1.86 \times 10^8$ while limiting

[5]This is done to easily form equal-resolution 2D cuts across any dimension.

[6]Where this could indeed change the noise statistics, this is a fair compromise for low-count data.

n_{count}	Average Counts Per Voxel	T	Number of Subsets
1×10^6	5.98×10^{-3}	0.13	186
2×10^6	1.22×10^{-2}	0.28	93
4×10^6	2.44×10^{-2}	0.57	46
8×10^6	4.89×10^{-2}	1.23	23
1.6×10^7	9.77×10^{-2}	2.62	11
3.2×10^7	2.00×10^{-1}	5.32	5
4.8×10^7	2.90×10^{-1}	8.14	3
9.6×10^7	5.84×10^{-1}	16.07	1
1.86×10^8	11.3×10^{-1}	30.78	1

Table 4.2: Summary of noisy realizations generated by varying number of electron counts n_{count} from the Gr/Ir(111) dataset. The acquisition time T is proportional to n_{count}, and 1.86×10^8 is used as the reference dataset.

the number of generated realizations, we sampled n_{count} as successive multiples of prior values, allowing more focus on lower counts. This same approach is applied to generate noisy realizations in other datasets, ensuring comparability across a range of noise levels and acquisition times.

4.3 Evaluation Criteria

As described in the last section, a 3D image is constructed by binning over the three physical axes k_x, k_y, E from different datasets. By slicing along these axes, different 2D images can be generated for analysis. We evaluate the BM3D algorithm, with and without the Anscombe transform[7], using the Gr/Ir(111) dataset, which has the highest average counts of all datasets. Despite this, the average counts per voxel is low (refer to last row of Table 4.2), making a true, noise-free image unavailable. This limitation complicates the task of evaluating denoising performance, as standard metrics rely on comparison with such a noise-free reference.

Image quality assessment (IQA) is a field dedicated to measuring the objective and perceptual quality of an image. Objective metrics such as peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), MSE and MS-SSIM require a reference image[8]—the latent clean image—to compare the denoised image against. No-reference metrics also exist, such as natural image quality evaluator (NIQE). However, these metrics are designed for real-world images. Subjective metrics such as the mean opinion score (MOS) can also be used, but these require evaluations by a group of experts, which can be impractical [46], [47].

Given the low n_{count} in the datasets of interest, a possible method to create a higher quality reference image is to sum across neighboring slices. Figure 4.6 illustrates such a case, where noise is progressively reduced by summing across increasing slices (see Figure 4.5 for a 3D depiction of slice axis), at the cost of feature blurring. Even with a large slice-summing, an ideal, noise-free reference image is not obtainable.

To address this, we assess metrics that are more resilient to noisy reference images. In Appendix B.2, we compare the performance of different metrics (PSNR, SSIM, MSE and MS-SSIM) for evaluating the denoising performance. Our findings suggest that the MS-SSIM metric is particularly well-suited for evaluating the denoising performance of images, when comparing against a noisy reference image. The MS-SSIM metric, conceived by Wang, Simoncelli, and

[7]This was discussed in detail in Chapter 3.
[8]The metric definitions can be accessed in Appendix A.5.

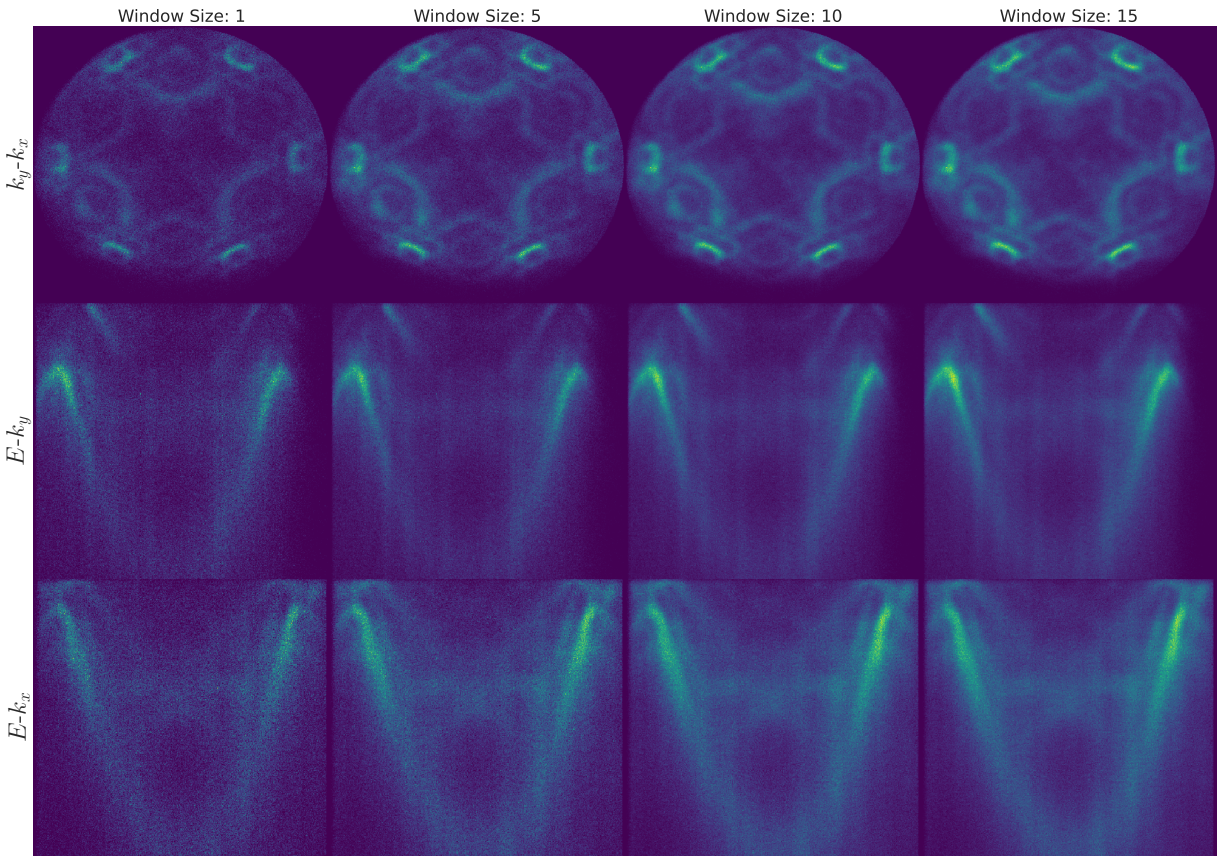


Figure 4.6: *Noise reduction via slice-summing:* E, k_y, and k_x slices at arbitrary positions of the Gr/Ir(111) 3D dataset, showing the effect of summing with different slice sizes w. The leftmost column shows a single slice ($w = 1$) with significant noise, while subsequent columns show slice-summed images with $w = 5,\ 10$ and 15. Increasing w progressively reduces noise at the cost of feature broadening (also referred to as blurring). This trade-off highlights the difficulty in obtaining a true, noise-free reference image even through averaging techniques.

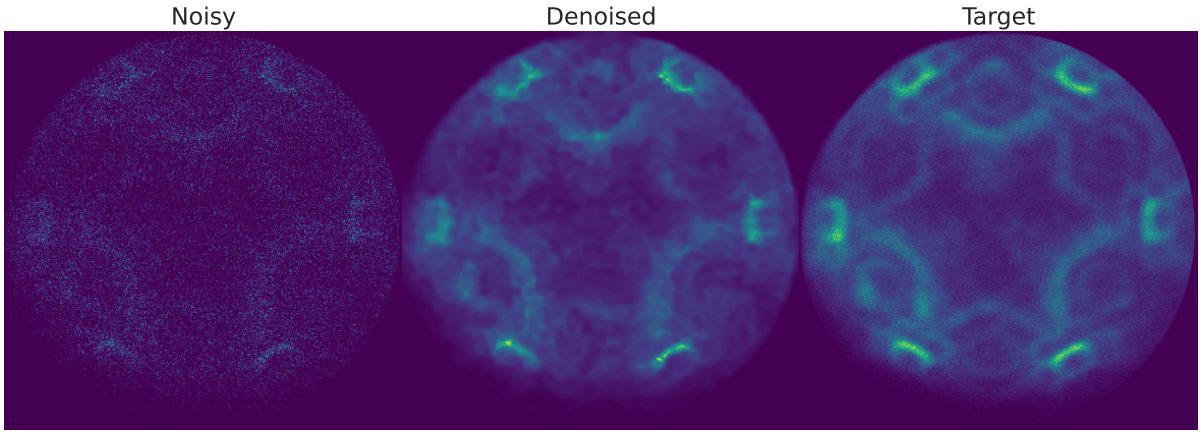


Figure 4.7: *Example noisy, denoised, and target images for* n_{count} *of* 1.6×10^7*:* The noisy and target images are formed by summing slices with size $w = 5$ and $w = 15$ along the E dimension, respectively.

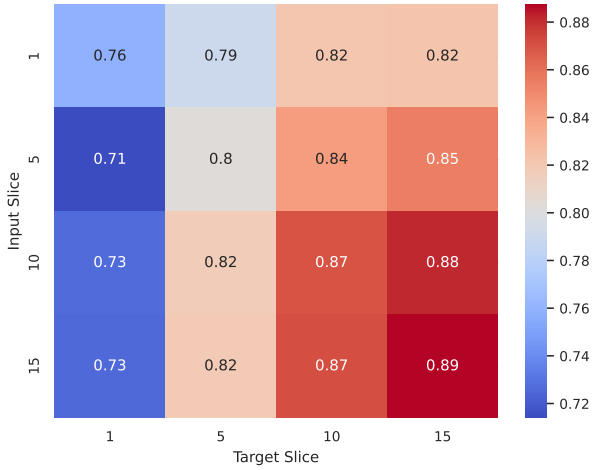


Figure 4.8: *MSSSIM heatmap for denoising:* Heatmap matrix showing the MS-SSIM values with different numbers of summed slices (w) for input and target images. The MS-SSIM (higher is better) is computed for the denoised images (of n_{count} 1.6×10^7) using the Anscombe-BM3D scheme. The matrix shows that using a larger w for the target image leads to better comparison of denoising.

Bovik [48], extends SSIM by incorporating multiple scales. Furthermore, some randomly chosen estimated images are analyzed by an expert to provide a qualitative assessment of the denoising performance.

Given that the ideal denoising aims to produce distortion-free images, one free from artifacts and removing all unwanted signal, it makes sense to target structural similarity rather than relative intensity values against the reference. Hence, throughout this study, the images are normalized to the [0, 1] range. This normalization ensures that comparisons focus on relative differences in image structures and features, rather than on absolute intensity values.

4.4 Denoising MPES data with BM3D

Let us start with denoising the noisy realization with $n_{\text{count}} = 1.6 \times 10^7$ ($T = 2\,\text{h}$) of the k_y-k_x images shown in Figure 4.6. This particular slice serves as a good reference due to its clear features. We use the Anscombe–BM3D–Inverse Anscombe scheme described in Figure 3.5.

We compare the effect of varying the slice-summing size w on the reported MS-SSIM score. Denoising the image with n_{count} of 1.6×10^7, we compute the metric with $w = 1,\ 5,\ 10$ and 15

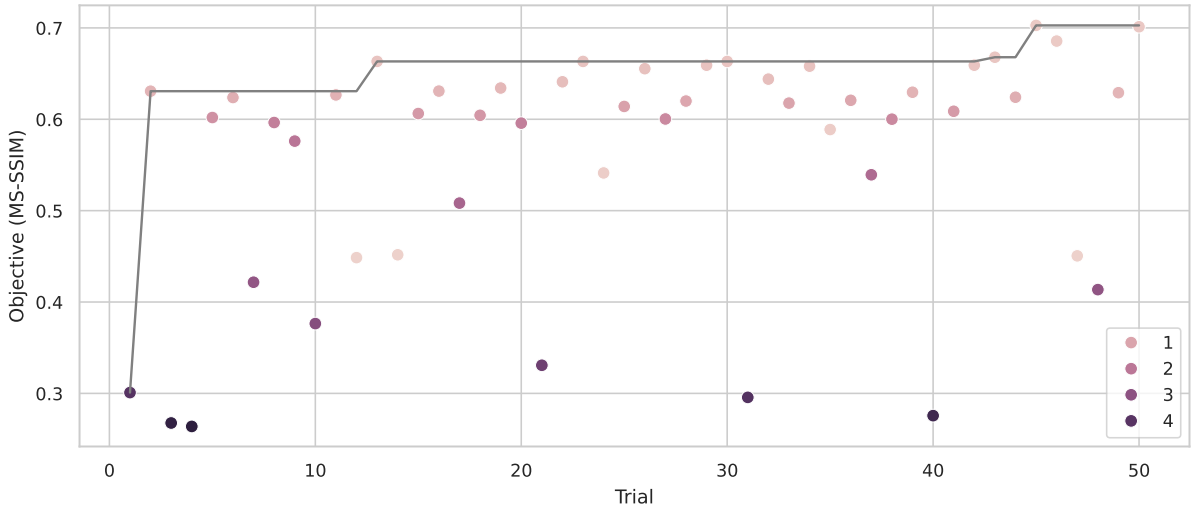


Figure 4.9: *Example optimization study:* Study to find the optimal BM3D hyperparameter σ for Anscombe-BM3D scheme. The plot shows how the optimization progresses over trials using the MS-SSIM score, which serves as the objective in the hyperparameter optimization. The cumulative maximum MS-SSIM score achieved up to each trial is shown as a gray line. The color of the scatter points represents the σ parameter for each trial, with larger and darker points indicating higher values of σ, showing that most time is spent near the optimal value.

(slice-summing along the E dimension) for all combinations of noisy and target (n_{count} of 1.86×10^8) images. An example noisy ($w = 5$), denoised and target ($w = 15$) set is shown in Figure 4.7.

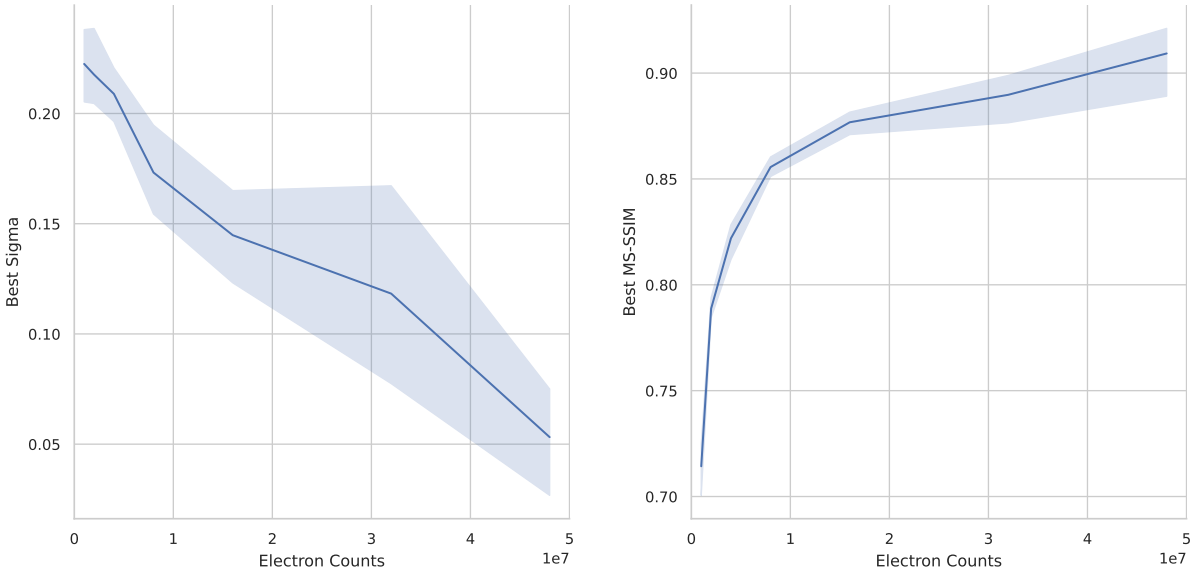
The matrix in Figure 4.8 shows that a larger w yields a better comparison for denoising. This is possibly because a larger w reduces the noise in the target image, leading to a more accurate representation of a clean target image. However, it should also be addressed that the denoising quality is not uniform across all w values, as the denoising algorithm has a parameter that, we will see in Section 4.4.1, scales with the noise level. The results presented in Figure B.2 with (a) $w = 1$ and (b) $w = 10$ support the idea that a larger w is beneficial for denoising, as the reported metric values improve significantly with the increased w. For a reasonable assessment, where the features are still sharp (not blurred due to slice-summing), we hence employ $w = 10$ for the analysis of the BM3D algorithm.

4.4.1 Finding the Optimal Sigma

Till now, we only focused on a single electron count (n_{count}) and a fixed denoising parameter σ. However, considering that the noise decreases with increased electron counts (see Appendix A.1), we would expect the required level of denoising to decrease accordingly.

One way to find the optimal denoising strength would be to estimate the noise level and use that estimate as the σ for denoising. This approach requires prior knowledge or assumptions about the noise distribution, which we have assumed to be Poissonian in Section 3.3. Later, in Section 5.2.2, we see that this assumption is not valid for MPES data acquired with an FEL light source.

A different approach would be to perform an optimization to find the optimal σ, denoted σ_{oo}, such that an objective function, such as the MS-SSIM metric, is maximized. It is important to note that optimizing for MS-SSIM and then using MS-SSIM as the quality measure can introduce potential bias, as the metric is both the optimization objective and the evaluation criterion. Howeverm due to the constraints of having a noisy reference image, this is the best approach available. For user-defined parameters in an algorithm, this type of optimization is known as a hyperparameter search [49].



(a) Optimal sigma value σ_{oo} as a function n_{count}, showing how the optimal denoising decreases with increasing electron count.

(b) MS-SSIM score as a function n_{count}, indicating that the denoising performance does not scale linearly with n_{count}, even when using optimal parameters.

Figure 4.10: *Denoising performance vs. electron counts:* The plots show the relationship between electron counts n_{count} and BM3D-Anscombe denoising performance measured using MS-SSIM score. (Left) The optimal value of the denoising parameter, σ_{oo}, and (right) the corresponding MS-SSIM score, which serves as the optimization objective. These results were obtained from 50 trials for each noisy image (slice-summed with size $w = 10$ across E), with the optimization focused on maximizing the MS-SSIM score by adjusting σ. The bands show the 95% confidence interval, computed over 5 images for each count.

Finding the optimal parameter through an exhaustive grid search is the simplest method, but it is computationally expensive. We perform a hyperparameter search with `optuna`[9] [50], which uses Bayesian optimization to find the optimal hyperparameters.

The search is conducted for Algorithm 3 on a small set of 5 identical-featured images. These images are independent noisy realizations extracted from the complete dataset (see Section 4.2.2), and the same slice (or slice-sums) is used from each independent image to ensure consistency. These images feature characteristics similar to those shown in Figure 4.7, with slices summed using $w = 10$.

The search is conducted over 50 trials for each n_{count} and each image. The optimization maximizes MS-SSIM score by adjusting the denoising parameter σ, which is constrained between $0-5$ to prevent the disappearance of features, as higher values can lead to significant loss of detail, and ≥ 0 as negative values for σ are invalid.

The optimization process for a single image (from n_{count} of 2×10^6) is illustrated in Figure 4.9. This figure shows the progression of the optimization process across trials. The Bayesian optimization efficiently explores the parameter space, with most trials focusing near the optimal σ value, to maximize the MS-SSIM score.

The results in Figure 4.10 corroborate the hypothesis that the optimal σ for denoising decreases with increasing electron counts, a linearly decreasing trend.

While using MS-SSIM as the objective function for optimization is good at showing the denoising performance improvement, it leads to more cautious results (low σ_{oo} values) as those fare better against the target, which has the relevant features but is also noisy. To counter that,

[9] https://optuna.org/.

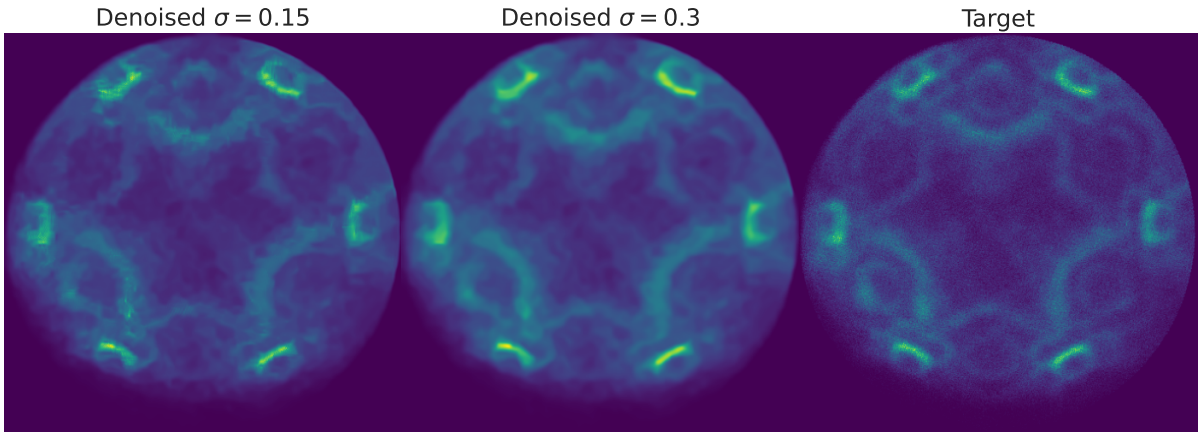


Figure 4.11: *Optimal denoising parameters:* (Left) Denoised image using the optimal $\sigma_{oo} = 0.15$ (optimal value found for 1.6×10^7 counts from the hyperparameter search), (middle) denoised image with $\sigma_o = 0.3$ (adjusted optimal value), and (right) the target image.

we scale the optimum σ_{oo} values by a factor of 2 to get a more aggressive denoising and denote that as σ_o, using this for denoising. A comparison of the denoised image using the adjusted optimal σ_o and the unadjusted optimal σ_{oo} is shown in Figure 4.11.

The denoising performance for $n_{count} < 4 \times 10^6$ are poor, both with and without the optimal value σ_o, even though the MS-SSIM reports high values. Figure 4.12a and Figure 4.12b show the denoised images for 2×10^6 and 4×10^6 counts. Since the images are summed over 10 slices, the average count per pixel increases proportionally to approximately 0.14 and 0.27, respectively, with the average target image count being 12.9. The slight deviation from a factor of 10 is due to uneven count distributions across the 3D volume. It can be seen that the perceptual quality of the denoised images is poor, with the features not being well-preserved. This highlights that BM3D is not well suited for denoising such low-count images.

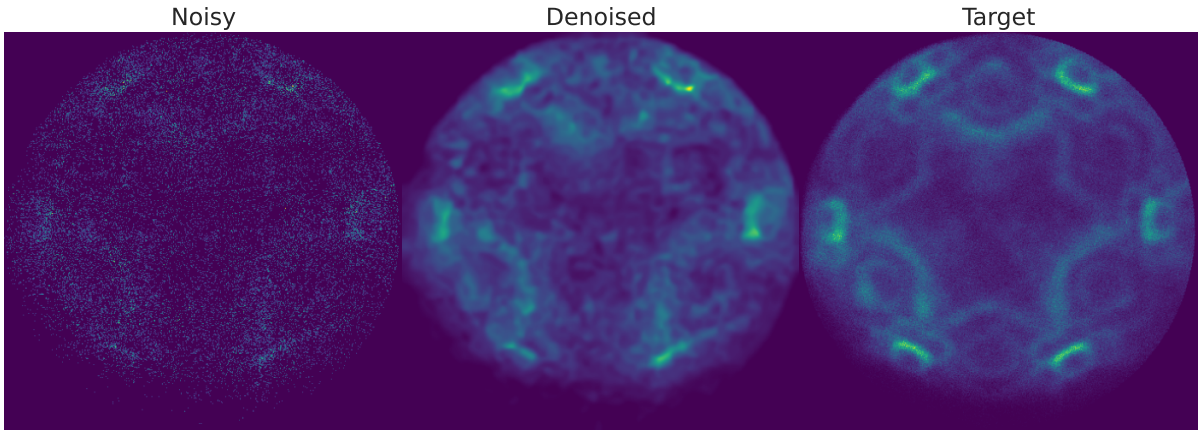
4.4.2 Comparing BM3D and Anscombe-BM3D

We conduct a visual evaluation of the denoising performance with Algorithm 1 (BM3D without Anscombe) and Algorithm 3 (BM3D with Anscombe), using the optimal σ_o values determined through the hyperparameter search from previous section. The results are presented in Figure 4.13, showcasing datasets with 8×10^6 and 4.8×10^7 n_{count} under different slice-summing conditions.

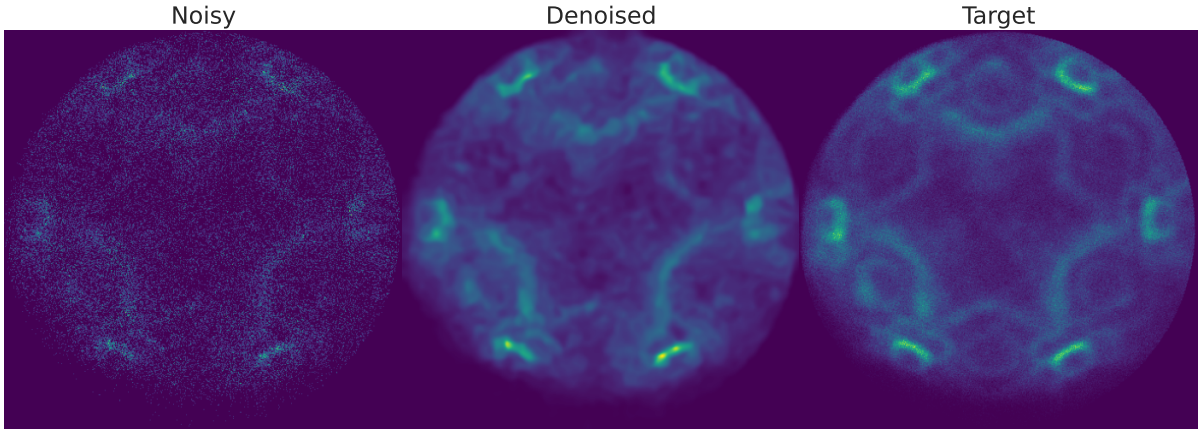
Figure 4.13a and Figure 4.13c show examples from these datasets, both slice-summed ($w = 10$). Perceptually, both schemes with their optimal σ_o values look similar. The MS-SSIM score also agrees with this statement, with a 0.01 improvement for the BM3D scheme.

Taking a difference, we also find that the values are near zero, indicating that the denoising performance is similar. Importantly, since σ_o has been only found for Anscombe-BM3D scheme, it is likely that the optimal for BM3D is different. This has to be further explored.

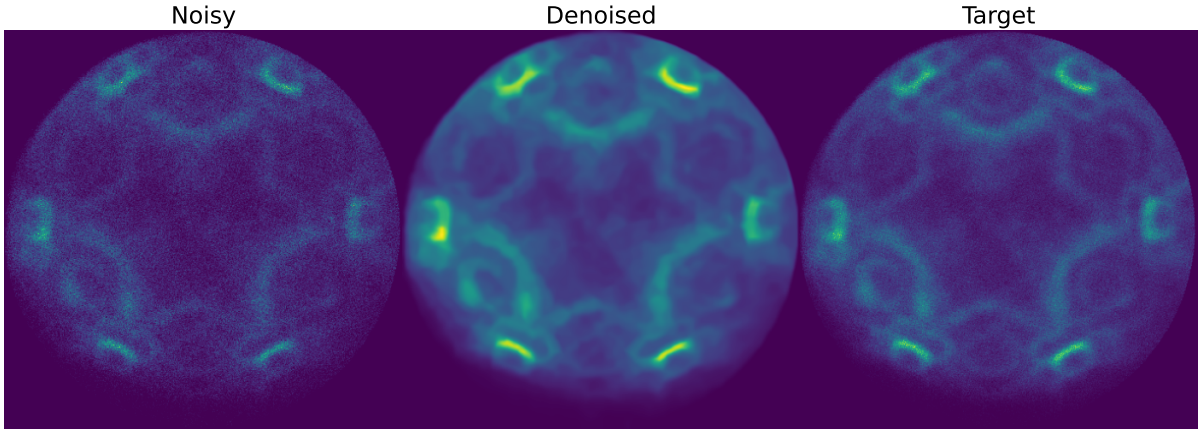
It is also found that slice-summing in Figure 4.13a and using a single slice in Figure 4.13b leads to the same average count per pixel. Applying the σ_o found for 8×10^6 for both figures, it can be seen that the denoising performance is similar. Trivial, but hence one can see the slice-summing as some sort of improving statistics/acqusition time. The reported MS-SSIM differ, but that is because they have only approximately same counts per pixel, with 0.56 and 0.61 respectively.



(a) $n_{\text{count}} = 2 \times 10^6$. The denoising performance is quite poor, even with the adjusted optimal $\sigma_{\text{o}} = 0.41$.

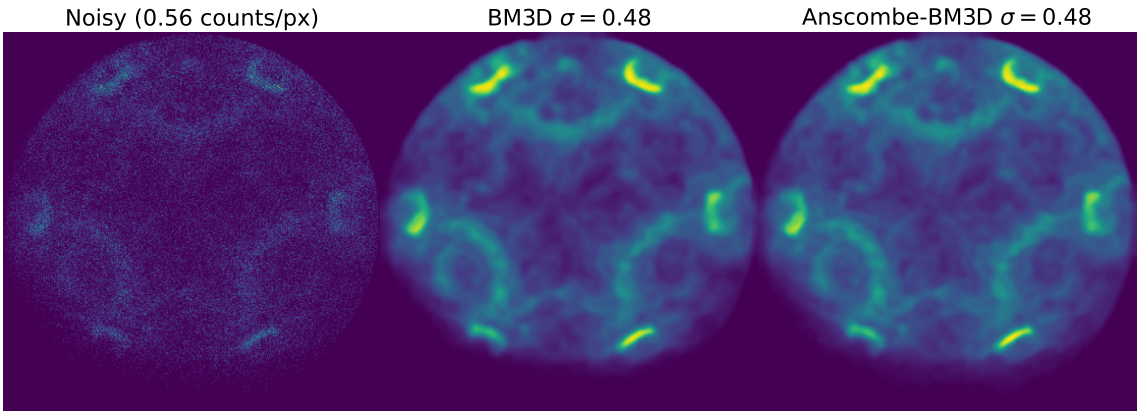


(b) $n_{\text{count}} = 4 \times 10^6$. The denoising performance leaves room for improvement, using the adjusted optimal $\sigma_{\text{o}} = 0.42$.

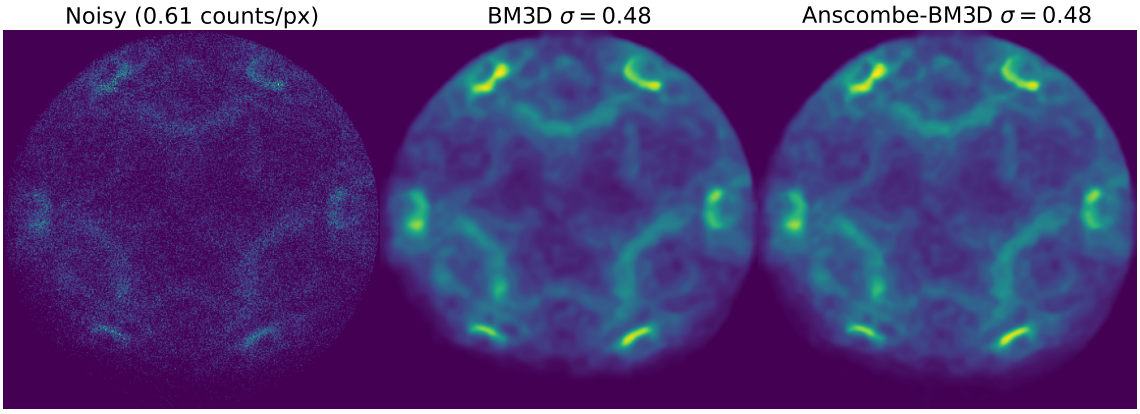


(c) $n_{\text{count}} = 4.8 \times 10^7$. Using the adjusted optimal $\sigma_{\text{o}} = 0.29$. At this n_{count}, MS-SSIM reports lower values for the denoised images compared to the noisy images, even though the denoised image features are well-preserved.

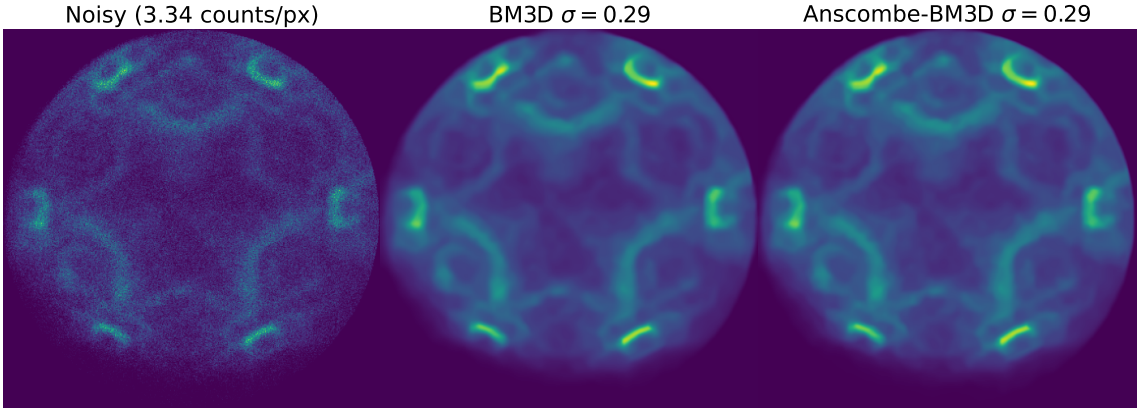
Figure 4.12: *Comparison across counts and slice-summing:* Comparison of noisy, denoised, and target images for $n_{\text{count}} = 2 \times 10^6$, 4×10^6 and 4.8×10^7, where the noisy and target images were formed using $w = 10$ over E to form k_x-k_y images. The Anscombe-BM3D algorithm was used for denoising.



(a) n_{count} 8×10^6 slice-summed with $w = 10$. Noisy image formed by slice-summing with size , having an average count of 0.56 per pixel. With optimal denoising parameter $\sigma_o = 0.48$ for this count. MS-SSIM: Noisy: 0.66, BM3D: 0.81, Anscombe-BM3D: 0.81.



(b) n_{count} 4.8×10^7 single slice $w = 1$. Average count of 0.61/px. With optimal denoising parameter $\sigma_o = 0.48$ of the 8×10^6 dataset. **MS-SSIM**: Noisy: 0.69, BM3D: 0.83, Anscombe-BM3D: 0.83.



(c) n_{count} 4.8×10^7 slice-summed with $w = 10$. Average count of 3.34/px. With optimal denoising parameter $\sigma_o = 0.29$. MS-SSIM: Noisy: 0.88, Anscombe-BM3D: 0.88, BM3D: 0.89.

Figure 4.13: *Comparison of BM3D with and without Anscombe at two different* n_{count}. (a) Dataset with 8×10^6 counts and slice-summing $w = 10$. (b) Dataset with 4.8×10^7 counts and single slice $w = 1$. Both noisy images have similar average counts per pixel with comparable metrics. (c) Dataset with 4.8×10^7 counts and slice-summing $w = 10$. The target image is formed by slice-summing with size $w = 10$.

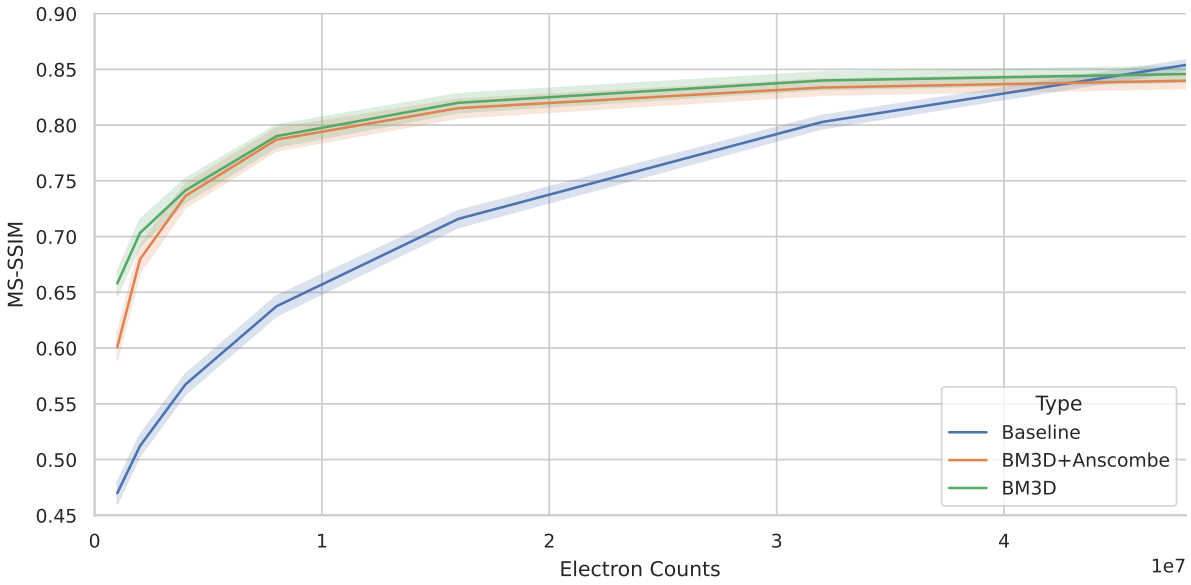


Figure 4.14: *Denoising performance of BM3D:* Denoising performance of the BM3D algorithm, with and without the Anscombe transformation. The optimal σ_o values were determined through a hyperparameter search conducted using the Anscombe-BM3D method and were applied to BM3D as well. The images are slice-summed with $w = 10$ slices for both the noisy and target images. The baseline metric is computed using the noisy image as input.

4.4.3 Varying Total Counts

The denoising performance is now evaluated for varying n_{count} using the MS-SSIM metric. A total of 1848 images were extracted from two separate noisy datasets for each n_{count} of 1×10^6, 2×10^6, 4×10^6, 8×10^6, 1.6×10^7, 3.2×10^7 and 4.8×10^7 (264 per n_{count}). These are slice-summed with $w = 10$ for both the noisy and target images.

As before, the MS-SSIM metric is used, with the baseline computed using the noisy images. Averaging over the large amount of images at varied counts gives us a robust estimate of the denoising performance. Using statistical bootstrapping (see Appendix A.4.1), where the estimate is computed over multiple resamples of the data, the 95% confidence interval for the MS-SSIM metric is also computed..

As shown in Figure 4.14, there is a noticeable improvement in image quality with counts up to $n_{\text{count}} = 4 \times 10^7$, with MS-SSIM values increasing from 0.6 to 0.83. Beyond this count, the metric starts reporting lower values for denoised results, whereas visual inspection reveals that the denoised images contain similar information as the target, smoother with preserved features. We can see this in Figure 4.12c, where the denoised image has a lower MS-SSIM value compared to the noisy image, but the features are well-preserved. This necessitates the need for higher quality target images for better evaluation. In particular, this also shows that the MS-SSIM has be interpreted carefully, but a better metric is not available.

The most notable finding is that the application of the VST leads to a slightly worse performance (Figure 4.14 orange vs. green lines), despite the expectation that it would enhance denoising performance for Poissonian noise. This has already been seen perceptually in Figure 4.13. In previous work by Makitalo and Foi, the authors demonstrated that applying the Anscombe transform indeed improves denoising performance. This strongly suggests that the noise statistics in the image deviate from a Poisson distribution.

If the Poisson noise is not voxel-wise independent, the slice-summing used to form the images might have altered the noise statistics (the sum of two independent Poisson random variables

is Poisson). The other obvious reason is that the electron count statistics forming the images are not Poissonian. This would result in the transforms and inverse not being optimal for the data. We shall see that the count statistics indeed deviate from Poissonian statistics in the next chapter.

5. Characterizing Photon and Photoelectron Statistics

To attempt at reconstructing the latent signal from incomplete observations, it is generally of interest to understand the underlying statistics, as this informs both noise characteristics and the likelihood of different observations. In many imaging and detection systems, especially those involving photons and photoelectron emissions, the observed data is inherently stochastic. Classical and quantum optics provide a comprehensive theoretical foundation to explain photon statistics. For instance, it is well understood that photons from a coherent light source follow a Poisson distribution (an outcome Section 3.3 was based on), often referred to as shot-noise. Whereas, for chaotic (bunched) light, the variance exceeds that compared to their mean, dubbed super-Poissonian. And finally for a squeezed light source, the distribution is sub-Poissonian [13, Chapter 5]. The same principle can be extended to photoelectron emission. When photons interact with a material, the resulting photoelectron emission follows the same statistical behavior as the incident photons [51], [52].

In this chapter, we begin by formally defining a general model for describing photoelectron event data using a Poisson point process (PPP), that captures the Poisson-distributed counting statistics for coherent light sources and the negative binomial (NB) distribution for SASE FELs. These model are then tested for counting statistics of photoemitted electrons on data from a pulsed coherent light source and a SASE FEL.

5.1 Modelling Photoelectron Statistics

Let us start by developing an intuition for why photons exhibit stochastic behavior. Photons are the quantized form of electromagnetic light and can be thought of as discrete energy packets. The energy of a photon is given by $E = h\nu$, where h is the Planck constant and ν is the frequency of the light. Due to the discrete nature of photons, it can be shown that the number of photons in a short time interval Δt is not constant. These fluctuations are known as photon shot noise.

For a light source with a constant flux[1] ϕ such as a single-mode laser, the average number of photons in a beam segment of length is given by L, $\lambda = \phi\frac{L}{c}$, where c is the speed of light. If we subdivide this L into many small intervals of size L/N, where N is large enough so that there is low probability of a photon being in an interval, eventually there will be divisions with no photons, divisions with only single photon, and negligible divisions with multiple photons. For all possible orderings, the probability of finding n subdivisions with a single photon and $(N-n)$ with no photons can be modeled by the Binomial distribution as follows, with $p = \frac{\lambda}{N}$ being the probability of a photon being in a segment:

$$P(n) = \binom{N}{n} p^n (1-p)^{N-n} \tag{5.1}$$

Using the Poisson Limit Theorem [5], it can be shown that as $N \to \infty$, and the probability $p \to 0$ such that $Np = \lambda$ remains constant, there is a convergence in distribution to the Poisson distribution (Appendix A.3.1). Hence, the Poisson distribution is a suitable model for counting statistics of photons from a coherent light source.

[1] Flux is the average number of photons passing through the cross-section of a beam per unit time.

5.1.1 Photoelectron Counting

Mandel [51], [52] and others have shown that the transition probability of an electron from its ground state to an unbounded state at the surface of a photodetector is directly proportional to both the duration of the time interval Δt and the instantaneous light intensity $I(t)$. This suggests that the rate density of photoelectrons is proportional to the light intensity $\lambda(t) \propto \int_A I(t)$, where A is the detector area.

The aforementioned can be mathematically formalized through one-dimensional stochastic process known as a point process (PP)[2]. A PP is used to model occurrences of stochastic events that happen in some space. If we look at the time domain, the PP allows us to the model the occurrence of events in time. The process generates a series of time points $\{t_1, t_2, \ldots, t_n\}$ where the events occur, within a time interval $[0, T]$.

A PP often used to model events is the Poisson point process (PPP), describable by the rate density function $\Lambda(t)$ (also known as the intensity function). The key property of such a process is that the events are statistically independent:

$$\Lambda(t_1, t_2, \ldots, t_n) = \prod_{i=1}^{n} \Lambda(t_i) \tag{5.2}$$

If the rate density function is constant, $\Lambda(t) = \Lambda$, the process is known as a homogeneous (or stationary) PPP, such as the case with a coherent laser light source [54]. If the rate function is time-dependent and deterministic $\Lambda(t)$, it is known as the inhomogeneous PPP. Such a situation could occur when the intensity of light is modulated, as is the case with a pulsed laser source. In a situation such as where the $\Lambda(t)$ itself is a stochastic variable, the process is known as the doubly stochastic PPP, or Cox process.

Integrating light intensity $I(t)$ over the time interval Δt can be treated as a random variable with probability density $P(W)$, with W as:

$$W = \int_{\Delta t}^{t+\Delta t} I(t')dt \tag{5.3}$$

The probability of detecting n photoelectrons in that time interval is then given by the Poisson transform relation [55] (a realization of the doubly stochastic PPP):

$$P(n, t, \Delta t) = \int_0^\infty \frac{\alpha W^n}{n!} e^{-\alpha W} P(W)\, dW \tag{5.4}$$

With a constant intensity light source, W becomes deterministic and Equation (5.4) simplifies to a Poisson distribution:

$$P(n, t, \Delta t) = \frac{W^n e^{-W}}{n!} \tag{5.5}$$

Due to the SASE process of an FEL light, discussed briefly in Section 2.3, the light intensity fluctuates stochastically. This leads to the photoelectron counting statistics being over dispersed and deviating from the Poisson distribution. Saldin, Schneidmiller, and Yurkov have shown in [56] that the photoelectron counting statistics from a SASE FEL light source can be modeled by a NB distribution (PMF and other details defined in Appendix A.3.1), a realization of the Cox process. And due to the light and photoelectron statistics relation, the photoelectron counting statistics also follow the same distribution.

The important thing to note here is that above discussion is about the statistics of photoelectrons from photon detection. Whereas, our experiment is of PES, where we are then interested in the (detected) emitted electrons (electron count distributions).

[2]For a more generalized description of PP, the reader is referred to [53].

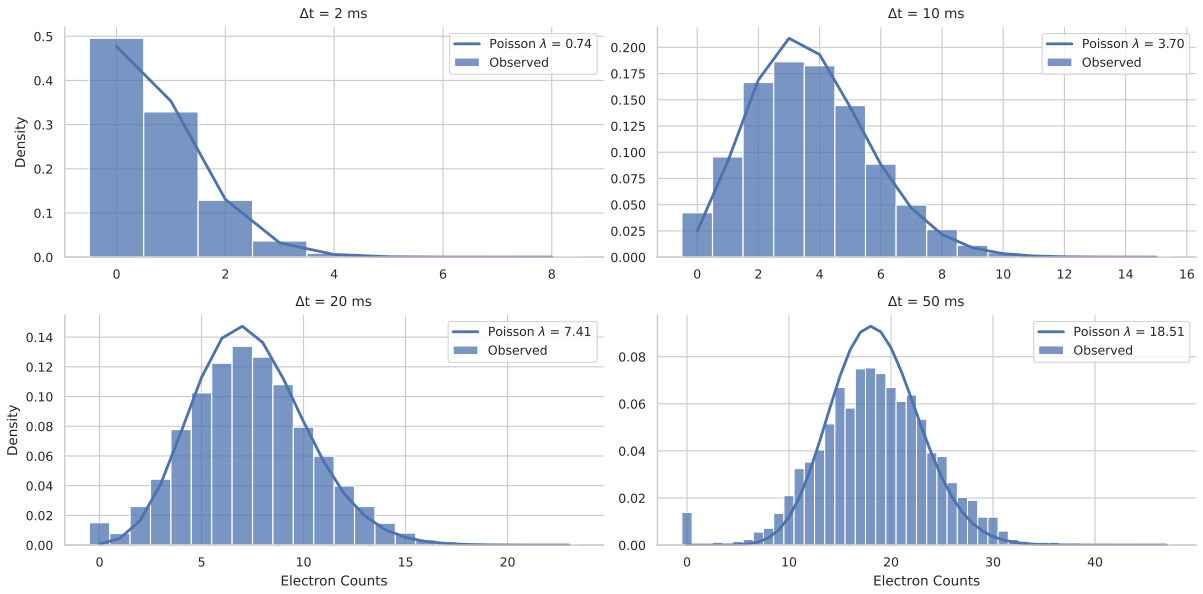


Figure 5.1: *Distribution of photoelectron counts* at time intervals $\Delta t = 2\,\text{ms}$, 10 ms, 20 ms and 50 ms for a selected volumetric subset of the full WSe_2 dataset. Poisson statistics are observed at smaller time intervals, but as the time window increases ($\Delta t = 50\,\text{ms}$), the data starts to deviate from the Poisson distribution, as spatial correlations become apparent. The total counts in this selected region are $n_{\text{count}} = 4.7 \times 10^4$ with total observation time $T = 126\,\text{s}$.

5.2 Analyzing Counting Statistics

There are two simple ways to analyze if a process can be modeled by the PPP[3]. *The interval statistics*, where the time intervals between events Δt should follow an exponential distribution for a PPP. The other is *counting statistics*, where the number of events (n_{count}) within a fixed time interval Δt should follow a distribution based on Equation (5.4).

We consider the latter case, for two reasons. First, from the discussion in Section 5.1.1, we have well-established hypotheses between the expected distributions for different light sources. Second, the timing information from the light source is coarser than the electron detection time. While the DLD detector measures the TOF of each electron, allowing us to get individual timing, the detector's dead-time could influence the time-interval statistics. Therefore, we analyze the *counting statistics*.

We test the counting statistics observed for PES at two different light sources: a pulsed HHG laser (WSe_2 dataset), and a SASE FEL light source (Gr/Ir(111) dataset). Based on theoretical considerations, we hypothesize that the counting statistics of the photoelectrons from the laser source would align a Poisson distribution, and from the SASE FEL source, with a NB distribution.

While the PPP should exhibit the same counting statistics for any time window Δt, practical challenges such as the long-term measurement drifts in intensity, can make that hard to realize. To account for this, we will analyze several time intervals to determine if the statistics remain valid based on the light source used. Given that the goal of MPES is to map band structures whose feature sizes are generally much larger than a single voxel, the data inherently exhibits spatial correlations[4]. An ideal experimental setup to measure counting statistics would instead

[3]Specialized statistical tests do exist to directly test for a homogeneous PPP, hypothesizing complete spatial randomness. Ripley's K-function can be used as a goodness of fit test [53, Section 2.6.4]. However, these do not test for the doubly stochastic PPP.

[4]Spatial here means in the image space, which also includes the time delay axis.

minimize spatial correlations and allow for long acquisition times to obtain reliable statistical estimates across different time windows.

In our analysis, we consider the detector-defined 3D subsets of image volume, which are often sparsely populated. While the most precise assessment of counting statistics would involve analyzing a single voxel, the sparsity of the data necessitates examining small subsets of the 3D volume instead. We select sufficiently small Δt to ensure that spatial correlations have minimal impact on the observed statistics, and also look at the cases where it is high.

5.2.1 HHG Light Source

Let us look at the dataset using HHG laser first: WSe_2. Figure 5.1 shows the count distribution at $\Delta t = 2\,\text{ms}$, 10 ms, 20 ms and 50 ms, with an observation time $T = 126\,\text{s}$, and one specific volumetric subset. The count distributions for smaller time intervals ($\Delta t = 2\,\text{ms}$, 10 ms and 20 ms) follow Poisson statistics, as expected from an uncorrelated photoemission process where spatial and temporal fluctuations are minimal. However, for longer time intervals ($\Delta t = 50\,\text{ms}$), the distribution starts to deviate from the Poisson distribution. This deviation suggests that spatial correlations, due to the material properties of the sample, become significant enough to impact the distribution. While the impact of pulse light source should be minimal since the time intervals we look at are much longer than the pulse durations, the intensity drifts could also cause the deviation. Figure B.3 shows count statistics from a different volumetric subset, forming the same conclusions.

From the above, we can conclude that the photoemitted electrons statistics are a realization of a PPP. Heimerl, Mikhaylov, Meier, *et al.* [57] have also recently showed that the emitted electrons show a Poisson distribution with a (coherent) pulsed laser light source.

To determine the homogeneity of the process, if we look at $\Delta t = 50\,\text{ms}$ in both Figure 5.1 and Figure B.3, the zero counts start forming a bimodal characteristic. This could be attributed to the pulse structure of the laser light source, where the absence of photons between pulses leads to a high occurrence of zero counts. Hence, due to this inhomogeneity in intensity, the statistics might be better modeled with an inhomogeneous PPP. For a pulsed light source with pulse interval τ_{pulse}, the counting statistics should still be Poisson for $\Delta t >> \tau_{\text{pulse}}$, and otherwise due to regular spacing when no events occur, it is better modeled by an inhomogeneous PPP.

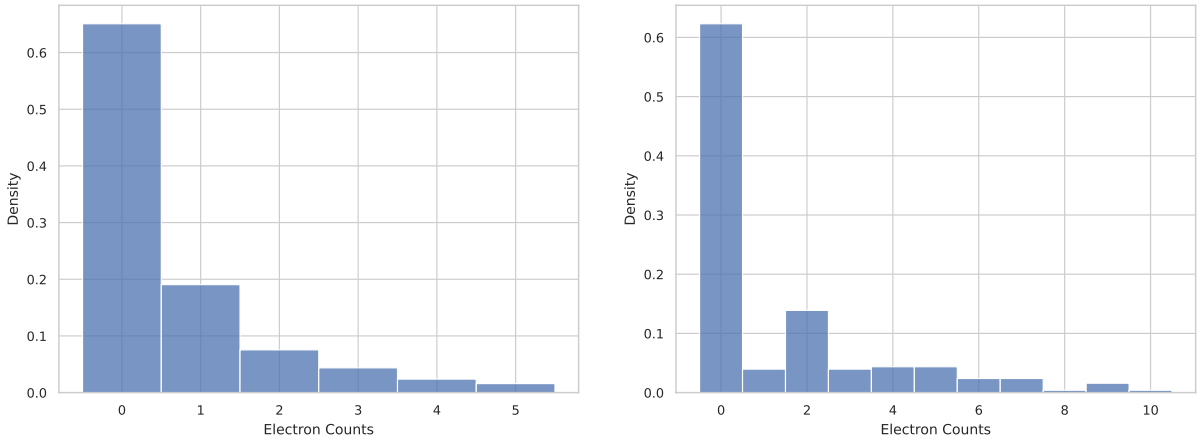
It must be noted that for HHG sources, Gorlach, Neufeld, Rivera, *et al.* [58] have shown that the highly non-linear HHG process can significantly affect the photon statistics. They have shown that depending on the generated harmonic, the photon statistics vary, with some harmonics able to exhibit squeezed light properties, and others over-dispersion. The seventh harmonic (21.7 eV) up-converted to EUV/XUV via HHG was used for the PES of WSe_2 [6]. Hence, in view of the above-mentioned work, further studies are required to understand the photon and the corresponding photoemitted statistics.

5.2.2 SASE FEL Light Source

Analyzing this data requires a few additional considerations. In Section 2.5, we already saw that the 8S DLD shows repeated counts for a single electron, due to the segmented structure. From preliminary analysis of counting statistics shown in Figure 5.2, it can be seen that this has a significant impact on the counting statistics, where the 2-event occurrence is unnaturally high compared to the others (Figure 5.2b). Therefore, this analysis will only consider a single layer (from the 2-layered delay-line structure), and ideally a volumetric subset from a single segment[5].

The second consideration is the long-term intensity drifts in the FEL light source. Figure 5.3

[5]Due to the overlap between segments.



(a) Single layer of the DLD.

(b) Both layers of the DLD.

Figure 5.2: *Counting statistics comparison for single layer or both layers of the DLD.* It is clear that the incorrect counting influences the statistics significantly.

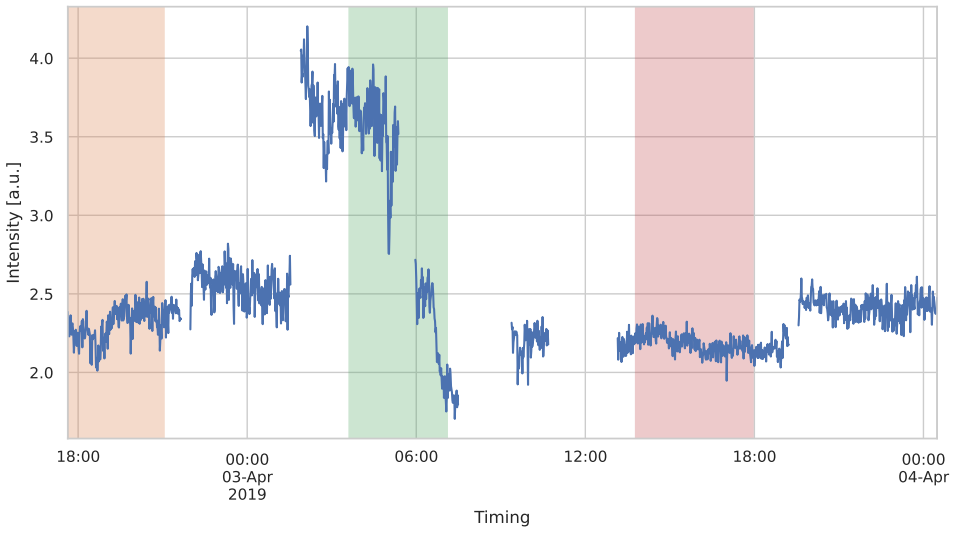


Figure 5.3: *Beam intensity measurements using the GMD* over the time period $T = 30\,\text{hour}$, with data window-averaged at 1 min intervals. The fluctuations in intensity can be observed, an intrinsic property of the SASE process. Other notable observations are the long-term drifts in intensity, and the beam interruptions (no recorded values). The orange, green and red spans indicate the time periods used for other analyses.

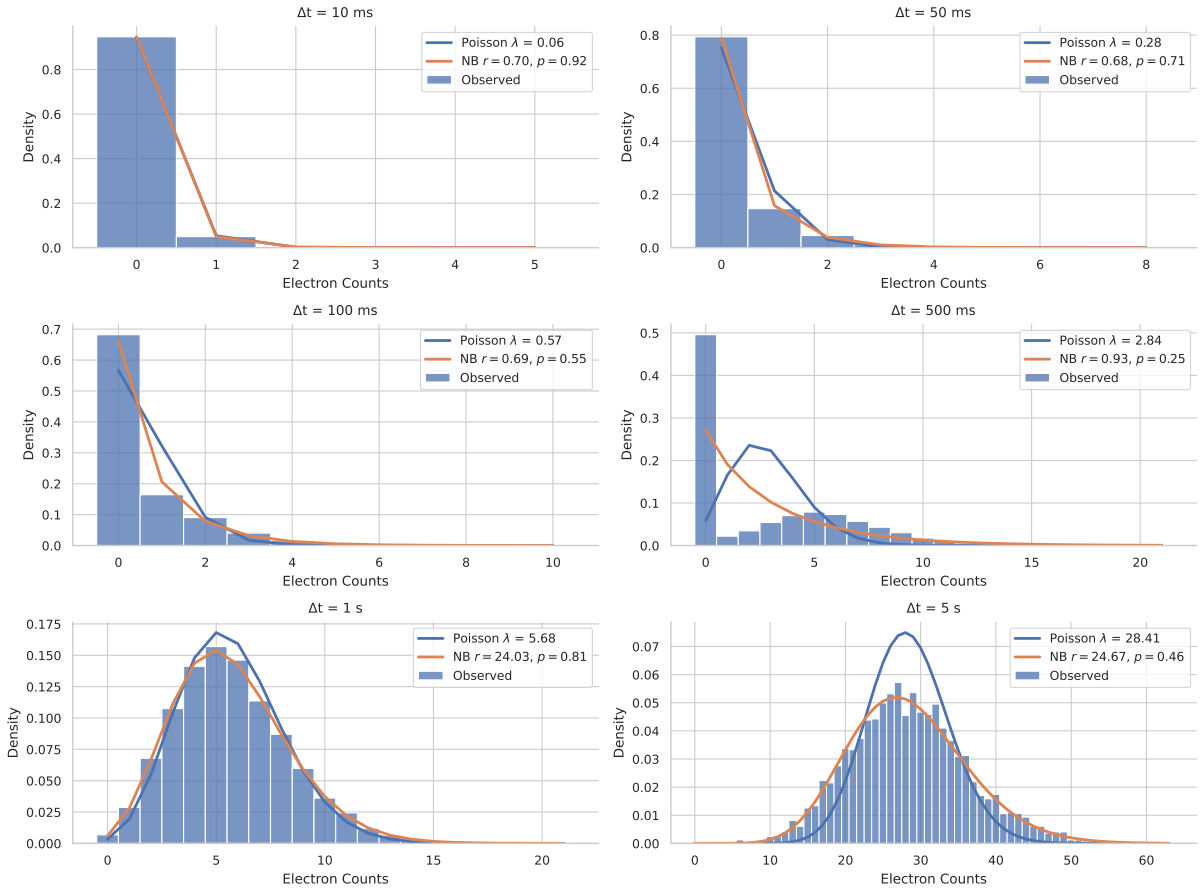


Figure 5.4: *Distribution of photoelectron counts* at time intervals $\Delta t = 10\,\text{ms}$, $50\,\text{ms}$, $100\,\text{ms}$, $500\,\text{ms}$, $1000\,\text{ms}$ and $5000\,\text{ms}$ for a selected volumetric subset of the full Gr/Ir(111) dataset. At shorter time intervals $\Delta t \leq 50\,\text{ms}$, Poisson statistics provide a good fit, reflecting the limiting case where the NB distribution approximates Poisson behavior. However, as the time interval increases, the distribution shows significant over-dispersion, with a pronounced right-skew characteristic of the NB distribution. Notably, at $\Delta t = 500\,\text{ms}$, the pulsed structure of the FEL becomes apparent, resulting in a high occurrence of zero counts. The total counts for this selected region are $n_{\text{count}} = 8 \times 10^5$, with a total observation time of approximately $T = 4\,\text{h}$ (refer to the orange region in Figure 5.3)

.

shows the beam intensity measurements[6] over the time period $T = 30\,\text{hour}$, with data window-averaged at 1 min intervals. The long-term drifts in intensity, and the beam interruptions due to accelerator operation can be observed. For analysis, we look at the data from the orange, green, and red spans. With no beam interruptions in the orange and red span, the expectation is that it follows the NB distribution. Whereas, the counting statistics would differ in the green span due to the beam interruptions.

We continue looking at the Gr/Ir(111) dataset from previous chapters. We look at a broad range of time intervals, as the count rate is much lower compared to the HHG source. The pulse structure of the FEL is more complex, as illustrated in Figure 2.2. The macrobunches (see *train*) arrive at a repetition rate of 10 Hz, meaning a new train of *pulses* arrives every 100 ms.

Figure 5.4 shows photoelectron count distributions for varying time windows ($\Delta t = 10\,\text{ms}$, 50 ms, 100 ms, 500 ms, 1000 ms and 5000 ms) within the orange span of Figure 5.3, with a total observation time of $T = 4\,\text{h}$. At $\Delta t = 10\,\text{ms}$, the Poisson and NB fit approximately match. Increasing the time interval starts to show better fit with NB than Poisson, with a pronounced right-skew characteristic. Notably, $\Delta t = 500\,\text{ms}$ has a high occurrence of zero counts, which could highlight the pulsed structure of the FEL, having regions within each train that have no photons. At $\Delta t = 5000\,\text{ms}$, the deviation from Poisson is apparent, with a significant over-dispersion. Figure B.4 shows the same analysis for the red span, with similar conclusions.

Figure 5.5 shows the interesting regime (green span). The frequency of zero counts clearly deviates from any hypothesis, and is only explainable by the beam interruption, and significant intensity difference. The intensity variation effect is clearly visible at $\Delta t = 5000\,\text{ms}$, where asides from the zero counts, the distribution is bimodal, highlight two distinct count rates.

In summary, the analysis of photoelectron counting statistics from the FEL light source highlights the complexity of the data influenced by the detector's structure, the SASE and long-term intensity fluctuations. The NB distribution is shown to be a more suitable model for such case. For the aim of estimating the latent distribution from incomplete observations (denoising), the spatial correlations in the image, and the detector structure should also be considered in the estimation process.

The deviation from Poisson distribution could also explain the lack of improvement in denoising performance when employing the Anscombe-BM3D scheme for denoising (see Figure 4.14). Applying the Anscombe transform (Equation (3.15)) on NB data reveals that the variance is not effectively stabilized, as can be seen in Figure 5.6. Further studies are hence necessary to analyze the effect of applying the transformations and inversions, as discussed in Sections 3.3.1 and 3.3.2, which were developed for Poisson noise. Anscombe identified an optimal VST for NB distribution [40], which could be beneficial for the FEL data. Additionally, designing an optimal inversion scheme could also benefit the denoising process.

[6] Measured using GMD, see *gas monitor detector*

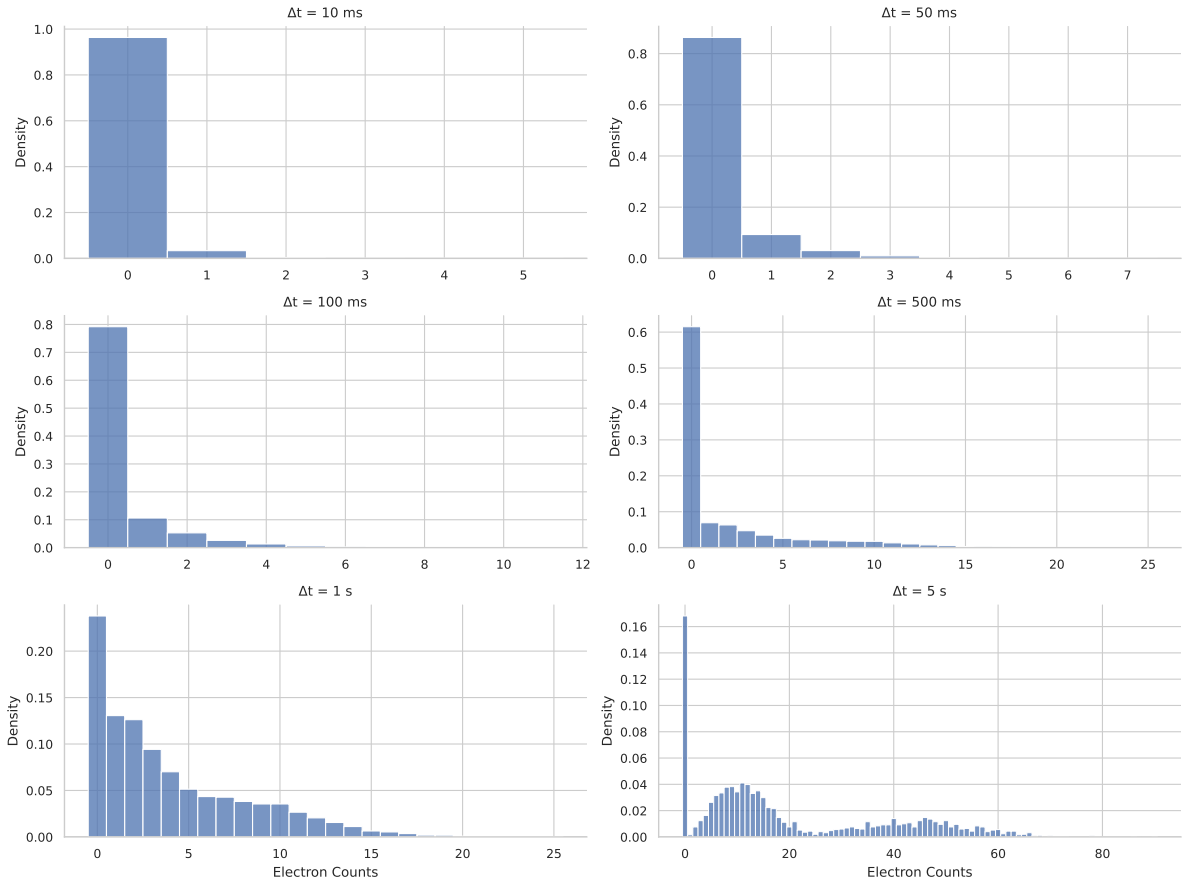


Figure 5.5: *Distribution of photoelectron counts* at time intervals $\Delta t = 10\,\text{ms}$, 50 ms, 100 ms, 500 ms, 1000 ms and 5000 ms for a selected volumetric subset of the full Gr/Ir(111) dataset. The total counts in this selected region are $n_{\text{count}} = 5 \times 10^5$ with total observation time $T \approx 3.5\,\text{h}$ (See green region in Figure 5.3). No fits are provided as the deviation is apparent. The frequency of zero counts is significantly higher, and the distribution is bimodal at $\Delta t = 5000\,\text{ms}$, indicating a significant intensity difference.

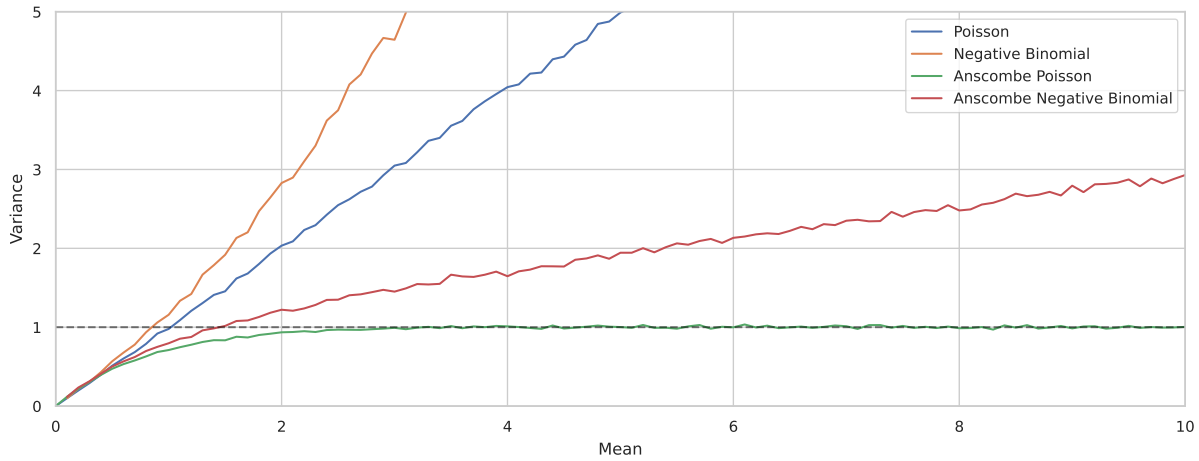


Figure 5.6: *Effect on variance by applying the Anscombe transformation to the synthetic Poisson and negative binomial distributed data.* The Anscombe transform, as expected, stabilizes the variance for the Poisson data. For the NB data, the variance stabilization is not as effective as for the Poisson data. This is expected considering the transform is optimal for Poisson data.

6. Learning from Noise

In Chapter 5, we saw that the photoemission statistics are influenced by the characteristics of the light source, that exhibit well-defined statistical properties. Understanding these specific characteristics can be helpful in designing tailored, model-based denoisers. For Poisson noise, we employed the Anscombe transform combined with BM3D (Section 3.3). While the Anscombe transform is capable of stabilizing Poisson-distributed noise into approximately Gaussian noise for moderate to high counts, it is less effective for very low counts (< 3) and the approximation breaks down as discussed previously in Section 3.3. Whereas, a VST tailored for NB statistics in conjunction with BM3D could be more effective to reduce noise.

Model-based approaches such as these can be powerful but they assume that the noise characteristics are accurately known and that the underlying signal conforms to the priors encoded. However, FEL-based photoemission experiments involve complex and heterogeneous noise sources. Factors such as long-term fluctuations in FEL intensity (Section 5.2.2), advanced detector designs (Section 2.5.3) and environmental variations can lead to deviations from idealized models like Poisson or NB noise prior. Adapting model-based methods to account for these complexities requires significant effort and may still fall short when dealing with unknown or dynamically changing noise patterns.

Unlike model-based methods, which depend on predefined assumptions about the noise and other image priors, deep learning, a subset of machine learning, can learn these relationships directly from data. This data-driven approach allows deep learning models to handle diverse and complex noise distributions, including those that are non-standard, multi-modal, or vary across a dataset. Furthermore, convolutional neural network (CNN)-based architectures excel at extracting non-linear and hierarchical features from high-dimensional data structures such as images extracted from MPES datasets. This allows to take advantage of the spatial and temporal correlations across multiple dimensions more effectively than many traditional approaches.

Machine learning can be broadly categorized in two categories: the *supervised* and the *unsupervised* setting. In supervised learning, the mapping (*model*) is learned by exposure to labeled data (input-target pairs). The exposure to such data, *training*, enables the model to establish a relationship between the inputs and the corresponding target. The model's expertise can then be used to map (predict) new inputs to the desired targets. In the context of image restoration, the model can gain expertise from pairs of corrupted (input) and clean images (target), and then use this expertise to restore other corrupted images.

Conversely, in the unsupervised setting, the model is exposed to the data without any labeled information, and the model is supposed to learn the underlying structure of the data. In context of image restoration, that would imply only giving the model the corrupted images, and expecting to map to the desired target–the clean image [59], [60].

In this chapter, the foundations of machine learning, covering the elements of how and why learning works are discussed. This is general to both the supervised and unsupervised setting. There the concepts of *hypothesis class*, *capacity*, *realizability*, and the concept of *generalization* capability are discussed. Then we take a look at a statistical learning[1] paradigm known as *empirical risk minimization (ERM)*, aiming to minimize the training error/empirical risk. ERM comes with its own set of challenges, such as overfitting in hypothesis spaces with high *capacity*, which can be mitigated by *regularization*.

[1] Machine learning and statistical learning are often used interchangeably but the latter places more emphasis on the statistical properties of the learning algorithms, and the former on the algorithmic aspects.

We discuss the combination of *CNN* and *Autoencoders*, which are commonly used in image restoration tasks. Subsequently, we discuss *Noise2Noise*, a learning framework for image restoration that does not require clean target images, introduced by Lehtinen, Munkberg, Hasselgren, *et al.* To fully explain this framework, we look at the *loss function* that allows this paradigm to work. We apply this training scheme using the concrete realization `UNET3D` architecture, an extension of the 2D variant first introduced by Ronneberger, Fischer, and Brox.

These methods are then applied to train a model that maps incomplete (noisy) observations[2] from PES to the true multidimensional image, with $d = 3$ dimensions. The aspects of data generation, training, and evaluation are henceforth, discussed in detail.

Much of the foundational concepts discussed are based on [63]–[66].

6.1 Foundations of Learning

Statistical regression is a classical example of a learning algorithm, where the goal of regression is to learn a function that maps the input data to the output data. Let us approach the image restoration problem from this perspective. Using the general observation model defined in Equation (3.4), the learning problem is then to find a hypothesis h that maps the degraded image X to the true image Y that generalizes well.

$$h : X \mapsto Y \tag{6.1}$$

In machine learning, no explicit assumptions of the *data generating distribution* are made except that all instances of the data are *iid* and generated according to a distribution $\mathcal{D}$ i.e. $Y, X \sim \mathcal{D}$. Continuing forward, since the discussion is not restricted to images, we denote input and target as x and y respectively.

6.1.1 Generalization

Generalization refers to the ability of a learning algorithm to perform well on new, unseen data. The goal of learning is to find a hypothesis h that generalizes well to new data.

If the data generating distribution $\mathcal{D}$ is iid and known, the *generalization error/population risk* $\mathcal{R}(\theta)$, with model parameters θ, can be minimized to find the optimal hypothesis $\hat{h}^*$. $\mathcal{R}(\theta)$ is defined as expectation taken across the data generating distribution $\mathcal{D}$, with the predicted output of a hypothesis $h(x;\theta)$ and the true output y:

$$\mathcal{R}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell(h(x;\theta), y)\right] \tag{6.2}$$

where the loss ℓ measures the difference between two quantities, such as of $h(x;\theta)$ and y.

Through optimization, the optimal hypothesis $\hat{h}^*$ can be found that minimizes the expected risk $\mathcal{R}$:

$$\hat{h}^* = \arg\min_{h\in\mathcal{H}} \mathcal{R}(\mathcal{D};\theta) \tag{6.3}$$

6.1.2 Hypothesis Class, Capacity and Realizability

The hypothesis h is chosen from a hypothesis space $\mathcal{H}$, where the hypothesis space $\mathcal{H}$ is the set of functions that the learning algorithm can choose from to approximate the true function. For

[2] In this case, incomplete observations can be understood as noisy data. For example, to completely resolve the bandstructure of a material, long observation times are required. However, using short observation times lead to noisy incomplete observations. Refer to Section 2.1 for the stochastic nature of PES.

example, in linear regression, the hypothesis space $\mathcal{H}$ consists of all possible linear functions[3] of the form:

$$\mathcal{H} = \left\{ h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0 \mid \mathbf{w} \in \mathbb{R}^d, w_0 \in \mathbb{R} \right\} \tag{6.4}$$

where $\mathbf{w}$ is the weight vector, $\mathbf{x}$ is the input vector, and w_0 is the bias term.

The *capacity* (the measure of size[4]) of this hypothesis space is smaller than other hypothesis spaces such as polynomial class of functions. Choosing a hypothesis class with a larger capacity allows for more complex functions to be learned, but we will see that this comes with a trade-off.

We want to find a space that makes our learning problem *realizable*. This means that there exists a element in the hypothesis space that can perfectly model the true mapping. An example for an unrealizable problem is if the data generation function is a $\sin(x)$ function, but the learning algorithm has a linear hypothesis space. In most real-world scenarios, the true space for complex data such as images is seldom known, and we forego the realizability condition. This is the *agnostic* learning setting. For classes with high capacity, even if they can not perfectly model the true function, they might approximate it well.

6.1.3 Empirical Risk Minimization

When the true data distribution $\mathcal{D}$ is unknown, the population risk can no longer be directly minimized. Instead, we approximate it by minimizing the empirical risk, also known as *training error*. . For a training set $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $\mathcal{S} \sim \mathcal{D}^n$, and model parameters θ, the empirical risk $\mathcal{L}(\theta; \mathcal{S})$ is defined as:

$$\mathcal{L}(\theta; \mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} \ell(h(x;\theta), y), \tag{6.5}$$

where ℓ is the loss function that measures the error between the predicted output $h(x;\theta)$ and the true output y. Empirical risk minimization (ERM) finds the hypothesis $\hat{h}$ from hypothesis space $\mathcal{H}$ that minimizes the training error $\mathcal{L}$

$$\hat{h} = \arg\min_{h\in\mathcal{H}} \mathcal{L}(\mathcal{S}; \theta) \tag{6.6}$$

by optimizing the model parameters θ. We shall later see that the assumption of having access to a true y can be relaxed in the *Noise2Noise* framework using L_2 (MSE) as the loss ℓ.

The gap between the $\mathcal{L}(\theta; \mathcal{S})$ and $\mathcal{R}(\theta)$ is known as generalization error and a hypothesis with a large gap generally implies *overfitting*.

6.1.4 Regularization

The ERM algorithm bears the risk of overfitting. This means that the hypothesis $\hat{h}$ might perform well on the training data $\mathcal{S}$ but poorly on new data drawn from the same distribution $\mathcal{D}$. Usually, this happens for rich hypothesis classes (high capacity), or when $\mathcal{S}$ is not representative of $\mathcal{D}$ and the ERM algorithm chooses a hypothesis that is too complex.

To counter this, *regularization* techniques are used. A penalty term based on model parameters is added to prevent overfitting. The ERM then finds the hypothesis $\hat{h}$ that minimizes the regularized loss function[5]:

$$\hat{h} = \arg\min_{h\in\mathcal{H}} \left(\mathcal{L}(\theta; \mathcal{S}) + \lambda R(\theta) \right). \tag{6.7}$$

[3] We also already seen an example of a problem where the hypothesis space $\mathcal{H}$ is constrained to linear functions: the linear minimum MSE estimator, Wiener filter discussed in Section 3.1.1.

[4] The VC dimension and Rademacher complexity are two popular measures of capacity.

[5] This is also known as Regularizated Loss Minimization.

where $R(\theta)$ is the regularization term that penalizes complex models, and λ is the parameter controlling the trade-off between the training error and the regularization term. Common regularization techniques include L_1 (Lasso) and L_2 (Ridge) regularization. L_1 regularization can be written as:

$$R(\theta) = \|\theta\|_1 = \sum j|\theta_j|$$

and L_2 regularization as:

$$R(\theta) = \|\theta\|_2^2 = \sum_j \theta_j^2$$

Regularization can also be seen as a way of restricting the hypothesis space $\mathcal{H}$ by encoding prior knowledge into the model. In the case of image restoration, it might be known a priori that the image is smooth, and can encode this prior knowledge by adding a penalty term that penalizes sharp changes in the image.

6.1.5 Uniform Convergence

It is always possible that with a small probability, the training data is not representative of the data distribution $\mathcal{D}$. Hence, every learning algorithm has a confidence and accuracy level that, in practice, is hard to quantify. For simpler cases, these bounds can be theoretically proven but practically, other evaluation methods are assumed, e.g. by empirically evaluating some test data, we can ascertain if the learned model generalizes well.

For a hypothesis space $\mathcal{H}$ that is finite, and iid training data all drawn from the same distribution $\mathcal{D}$, the ERM algorithm can be shown to have a generalization error that converges to zero as the number of training samples n goes to infinity. This is known as the *uniform convergence* property of the ERM algorithm. This can further be generalized to infinite hypothesis spaces, through non-uniform convergence. However, considering that most machine learning takes place through discrete data, making infinite hypothesis spaces finite , the finite hypothesis space is a reasonable assumption.

6.2 Learning Algorithms

There are a myriad of supervised learning algorithms built upon the foundation of ERM, that aim to minimize the expected loss by minimizing the empirical risk (Equation (6.6)). Some examples include linear regression that minimizes the MSE loss, logistic regression for binary classification[6] that minimizes the cross-entropy loss, and support vector machines, that can be framed as a hinge loss minimization problem, aim to maximize the margin between different classes [67].

Let us develop towards one broad class of learning algorithms that have shown remarkable success in countless applications, including image restoration: *Neural Networks*.

6.2.1 Linear Regression and Perceptrons

Linear regression[7] is among the simplest forms of machine learning models, seeking to model a relationship between the continuous output $y \in \mathbb{R}$ as a weighted sum of input features $\mathbf{x} \in \mathbb{R}^d$ with an added bias term w_0:

$$y(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0 \tag{6.8}$$

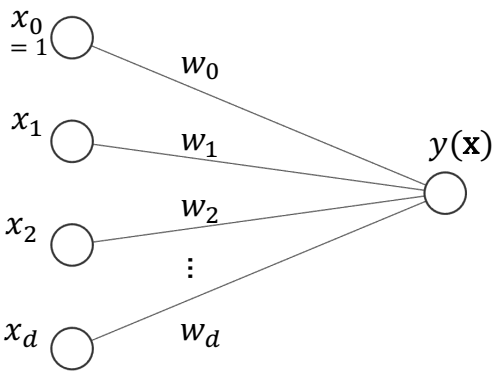
[6] With softmax regression as a generalization for multi-class classification. The term "regression" is conventionally used, but this is actually a classification task.

[7] We briefly saw in Equation (6.4) the hypothesis space of linear regression being all linear functions.

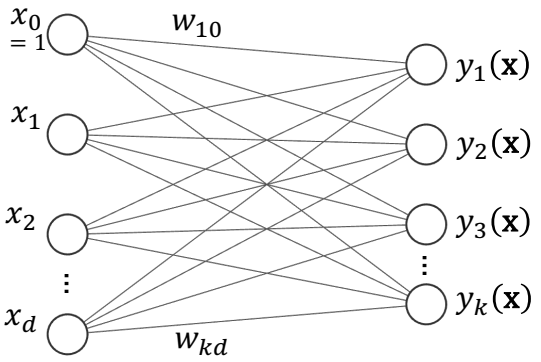
The model parameters can be learned based on ERM i.e. minimizing a loss function to learn the weights. For example, using the MSE loss, the objective becomes:

$$\ell(\mathbf{w}, w_0) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w}^\top \mathbf{x}_i - w_0)^2 \tag{6.9}$$

where n is the number of training samples, and $(\mathbf{x}_i, y_i)$ are the input-target pairs. This can be minimized in closed-form to find the optimal weights $\hat{\mathbf{w}}$ and bias $\hat{w}_0$ [67].



(a) A computational graph of a standard perceptron. The input features $\mathbf{x}$ are linearly combined with the weights $\mathbf{w}$ and bias w_0 to produce the output y.



(b) A computational graph of a multi-output perceptron. The input features $\mathbf{x}$ are linearly combined with the weights $\mathbf{w}$ to produce outputs y_k for each dimension k.

Figure 6.1: *Computational graphs for perceptron architectures.* (a) Standard perceptron. (b) Multi-output perceptron.

Equation (6.8) can be expressed as a computational graph, as shown in Figure 6.1a. This architecture is commonly referred to as the standard perceptron, the simplest of neural networks.

6.2.2 Generalization of Linear Models

Linear regression can be extended to more complex settings, such as multi-output models (e.g. multi-class classification or multidimensional regression), with y_k representing the output for the k-th dimension, and x_0 often set to 1 as a convention for including the bias term w_{k0}:

$$y_k(\mathbf{x}) = \sum_{j=0}^{d} w_{kj} x_j \tag{6.10}$$

Such an architecture is known as a single layer perceptron, and can be visualized as shown in Figure 6.1b.

This can be further generalized by introducing non-linear basis functions $\phi_j(\mathbf{x})$, transforming the input features $\mathbf{x}$ before the linear combination:

$$y(\mathbf{x}) = \sum_{j=0}^{d} w_j \phi_j(\mathbf{x}) \tag{6.11}$$

These non-linear transformations allow models to approximate non-linear relationships in data. Due to this, the loss can no longer be minimized in closed-form so iterative optimization algorithms such as gradient descent are used [67]. Conventionally, these were basis functions were predefined and hand-crated based on domain knowledge, but we next look at deep learning models that can learn these basis functions.

6.2.3 Deep Learning

Multi-layer perceptrons build upon the foundations of linear models but eliminate the need to predefine input transformations. This is done by learning the parametrized basis functions from the data.

One way to generalize the linear model is by applying a non-linear activation function g to each input in Equation (6.10). For example:

$$y_k(\mathbf{x}) = g\left(\sum_{j=0}^{d} w_{kj} x_j\right). \tag{6.12}$$

To achieve this, neural networks add multiple layers to the computational graph, with each layer applying a linear transformation followed by a non-linear activation. A simple architecture is shown in Figure 6.2 (with the addition of non-linearities), illustrating how the standard perceptron is extended to deeper networks.

Due to the non-linearities and layered structure, neural networks have high-capacity hypothesis spaces, suited to approximate the complex mappings such as multidimensional data such as images.

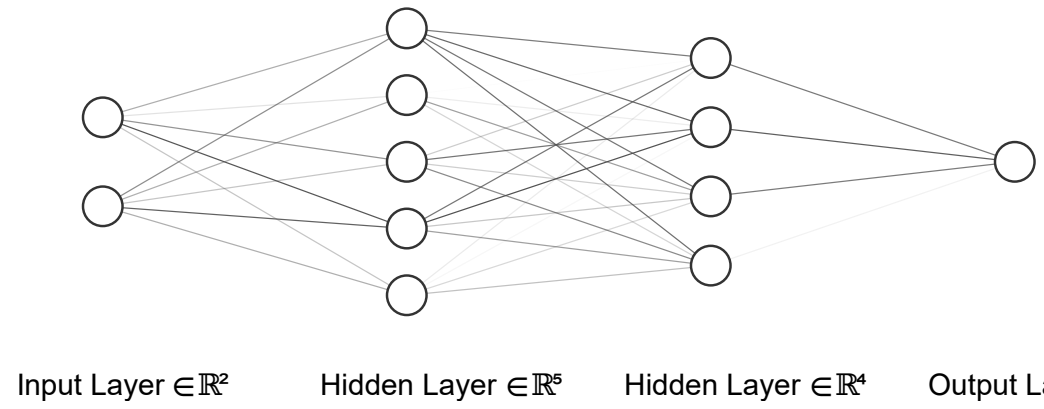


Figure 6.2: *An example trained fully connected neural network architecture* with an input layer, two hidden layers, and an output layer. The input features are transformed through weights and biases, followed by a non-linear activation function in the first hidden layer. This process is repeated in the second hidden layer from outputs of the first (hidden) layer, producing the final outputs. The weights are represented by the lines connecting the neurons, with opacity indicating the weights' strength.

Activation functions are generally differentiable non-linear functions that introduce non-linearity into the network, allowing it to model complex relationships in the data. Common activation functions include the sigmoid function, that maps input values to a range between 0 and 1, useful to interpret probabilistically, and used in binary classification tasks.

$$S(x) = \frac{1}{1 + e^{-x}} \tag{6.13}$$

Current state-of-the-art architectures often employ Rectified Linear Unit (ReLU) or its variants as the activation function, owing to their superior performance in training deep networks. ReLU defined as

$$\mathrm{ReLU}(x) = \max(0, x) \tag{6.14}$$

sets negative inputs to zero while preserving positive inputs.

While ReLU is not differentiable at $x = 0$, it is widely favored due to its simplicity and effectiveness in training deep networks. Variants of ReLU, such as Leaky ReLU and Parametric ReLU have been introduced to address the issue of dead neurons, where neurons cease to learn during training. Leaky ReLU introduces a gradient for negative inputs

$$\text{Leaky ReLU}(x) = \max(\alpha x, x) \tag{6.15}$$

defined by a small constant α. Parametric ReLU extends this concept by allowing the gradient to be learned during training.

Training a neural network (NN) involves two primary phases: the forward pass and the backward pass. During the forward pass, the input data is propagated through the network, layer by layer, to generate an output. Following this, the backward pass occurs, in which the error, defined as the difference between the predicted output and the true output, is propagated back through the network using the gradient of the loss function. This step is crucial for updating the weights of the network. The training process utilizes optimization algorithms, such as Stochastic Gradient Descent (SGD) [68] or Adam Optimization [69], which iteratively adjust the network's weights to reduce the loss.

6.3 Neural Networks for Image Restoration using Noise2Noise Framework

In the classical supervised-learning setting for image restoration, training assumes access to clean target images. Similar to Section 6.1.3, training can be formulated as

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \frac{1}{|\mathcal{S}|} \sum_{(X,Y) \in \mathcal{S}} \ell(h(X;\theta), Y), \tag{6.16}$$

where $\mathcal{S} \sim \mathcal{D}^n$ is the training set with images $\mathcal{S} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ with (X, Y) the noisy and clean target images.

Most often, instead of a multilayered perceptron architecture, CNN are preferred for image restoration tasks. CNN are well-suited for image data due to their ability to learn spatial hierarchies of features through convolution operations. he convolutional layers in a CNN automatically detect patterns such as edges, textures, and shapes, while the pooling layers downsample the data to reduce both the number of parameters and computational complexity. The output of the convolutional layers, known as feature maps, captures these learned patterns, and max-pooling further enhances efficiency by retaining the most significant values in a region, reducing spatial dimensions. This pooling process also contributes to the network's robustness to spatial variance. Furthermore, CNNs impose a strong prior on the hypothesis space, assuming that the data is translationally invariant, which is a reasonable assumption for images [66]. In the context of this work, we will explore a specific realization of CNNs, namely the UNet, which is known for its effective use in image restoration tasks.

In many practical scenarios such as our MPES data, obtaining the true clean image Y is not feasible, although sufficiently high quality noisy data exists. In the instance when multiple noisy realizations of the same underlying signal are available, Lehtinen, Munkberg, Hasselgren, *et al.* demonstrated that it is possible to train a NN to learn a restoration mapping to the underlying clean signal using only noisy observations for both inputs and targets [61], a framework known as *Noise2Noise*.

In the following sections, we see why Noise2Noise works and how it is applied to our image restoration problem.

6.3.1 Point Estimation and Loss Functions

We discussed in Section 6.1.1 that we aim to learn a hypothesis that generalizes to new data coming from the distribution $\mathcal{D}$. This can be done by minimizing the risk (Equation (6.3)).

Consider an example point estimation problem where the objective is to find the scalar $\hat{x}$ that minimizes the expected deviation with respect to a set of observations $x_1, x_2, \ldots, x_n$. This can be formalized as:

$$\hat{x} = \arg\min_{\hat{x}} \mathbb{E}[\ell(\hat{x}; x)] \tag{6.17}$$

Different loss functions yield varying estimates based on the observations. For the case of L_2 loss, the minimization recovers the mean of the observations:

$$\hat{x} = \mathbb{E}[x] \tag{6.18}$$

Similarly, for the L_1 loss, the estimate is the median of observations.

6.3.2 Zero-Mean Noise

Combining Equation (6.2) and Equation (6.3), we can write the risk minimization problem as:

$$\arg\min_{h \in \mathcal{H}} \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y|x \sim \mathcal{D}} \left[\ell(h(x; \theta), y) \right] \right] \tag{6.19}$$

where using the law of total probability, the joint expectation $p(x, y)$ can be factorized to $p(x) \cdot p(y|x)$, and hence the expectation as well. This formulation is equivalent to solving the point estimation problem for each input sample separately. Due to this, the NN inherits the properties of the loss function.

We established in Equation (6.18) that minimizing the L_2 loss recovers the mean of the observations. Now, consider the case where the training targets y are corrupted with zero-mean noise, $\mathbb{E}[n] = 0$. The expectation remains unchanged i.e. $\mathbb{E}[y] = \mathbb{E}[x'|y]$, with x' being the corrupted data. For example, we previously demonstrated Poisson noise being zero-mean in Section 3.3 (see Equations (3.10) and (3.11)), and hence, the L_2 loss could also recover, on expectation, the true value of a Poisson noise corrupted variable. Using other losses such as L_1 could allow recovery of signals with significant outlier content.

This principle can extend to NN training as we already said that the network inherits the properties of the loss. When minimizing the risk, a NN trained with zero-mean noise-corrupted targets will converge to the same optimal hypothesis as it would with clean targets. This holds true in the context of ERM as well. With infinite training data, minimizing the empirical risk using noisy observations is mathematically equivalent to minimizing it with clean targets (Equation (6.6)).

6.3.3 Noise2Noise Training for Finite Data

Let us formalize the Noise2Noise training for a realistic case of finite examples. Consider a training set $\mathcal{S}' = \{(X_1, X_1'), \ldots, (X_n, X_n')\}$ where $\mathcal{S}' \sim \mathcal{D}^n$, each pair (X, X') independent noisy realizations of the same underlying signal $y \sim \mathcal{D}$. Using the L_2 loss, we can redefine ERM formulation from Section 6.1.3 as

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \frac{1}{|\mathcal{S}'|} \sum_{(X, X') \in \mathcal{S}'} (h(X; \theta) - X')^2 \tag{6.20}$$

giving us a hypothesis $\hat{h}$ that minimizes the training error using only noisy observations for both inputs and targets.

For finite data, the quality of the estimate depends on the variance of the noise in the targets, divided by the number of samples N [61, supplementary material]. This means that increasing the dataset size reduces the variance of the estimate, bringing it closer to the hypothesis had we minimized with clean targets. For image data, N corresponds to the total number of scalar components[8] across the dataset, so having more voxels and data effectively brings us closer to the infinite data case.

6.3.4 Regularization through Noisy Targets

It is important to note that the earlier discussion assumed that learning with clean targets is inherently successful, while in fact, as we discussed before, ERM is prone to overfit (generalizes poorly) in high-capacity hypothesis spaces. Using noisy targets actually has the unexpected benefit to act as a form of regularization. The noise in the targets perturbs the training objective, preventing the network from relying excessively on precise input-output mappings. For image restoration tasks, this can lead to better generalization, as the network focuses on recovering the underlying clean signal structure rather than spurious details in the data.

6.4 Training an MPES Denoiser

As seen in Section 4.2, the experimental setup (refer to Sections 2.4 and 2.5) for MPES puts us in a unique position of having access to multiple noisy realizations[9] of the same latent clean image. Hence, we can leverage the Noise2Noise ERM, as shown in Equation (6.20), to train a NN to denoise MPES images. We perform all the shown experiments with the L_2 loss[10] as the underlying images we aim to recover are the expectation of the noisy images.

Selecting the training data and NN architecture are imperative for the success of the denoising task. Given the inherent multidimensional nature of MPES data, it is advantageous to exploit cross-dimensional correlations to better estimate the underlying image. While 2D image restoration methods already utilize such principles in two dimensions (an example we previously discussed with BM3D in Section 3.2), higher dimensional data can provide additional information to improve the denoising performance[11].

The consideration of cross-dimensional correlation is necessary as the 2D image slices from the highest count dataset (n_{count} of 1.86×10^8 from Gr/Ir(111)) remain noisy. In fact, in Section 4.4, this is addressed by slice-summing before denoising and evaluation. However, this summation leads to feature blurring, whereas harnessing correlations across higher dimensions provides a more effective alternative.

To this end, the UNET3D architecture is employed instead of their 2D counterparts by utilizing 3D convolutions, upsampling and pooling. In the following, we detail the data generation process, network architecture and the training and validation schemes.

6.4.1 Training Data Generation

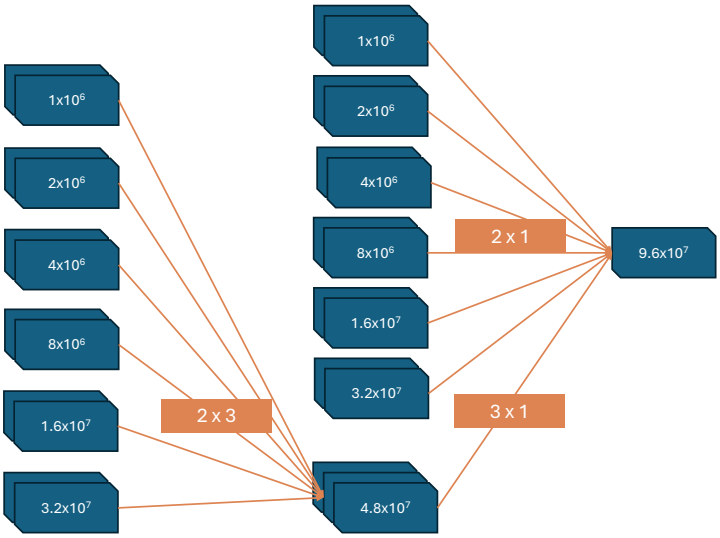
The Gr/Ir(111) dataset, with its high count, is used as the training dataset. While Noise2Noise training allows training with noisy data, it is not immediately apparent at what noise levels the training should be conducted. Naturally, the noise levels we aim to denoise should be reflected in the training set. Particularly, datasets obtained with shorter acquisition times T corresponding

[8] Number of images n x number of voxels per image x number of color channels.

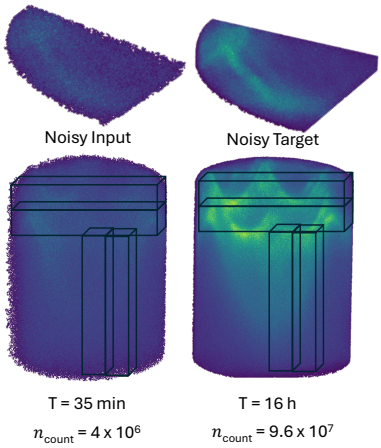
[9] through binning subsets of the single-event dataset

[10] Some preliminary study using the L_1 loss also shows promise. However, this could be attributed to the significantly high noise at low count levels.

[11] Algorithms such as BM4D [34] attempt to exploit this by considering the 3D structure of the data.



(a) Flowchart illustrating the input-target dataset pairs, derived from subsets of the Gr/Ir(111) dataset. A total of 34 unique noisy datasets are represented, with each dataset corresponding to one blue box in the chart. The different combinations shown in the chart generate 51 input-target pairs across counts $1 \times 10^6 - 9.6 \times 10^7$.



(b) Showing example noisy input and noisy target and how 3D subsets (also called patches) are extracted for training. On top, how extracted subsets could look.

Figure 6.3: *Illustration of generating neural network training pairs.* (a) The flowchart showing the input-target dataset pairs used for training, and (a) an example of noisy input and noisy target showing how 3D subsets are extracted from the volume. Through the combination with patch-based extraction, a total of 5967 training pairs are generated. The 4.8×10^7 and 9.6×10^7 counts serve as the target datasets, with 9.6×10^7 additionally functioning as the target for the 4.8×10^7 datasets.

to lower n_{count} are of interest. Lehtinen, Munkberg, Hasselgren, *et al.* reported succesful training and inference on Poisson corrupted images with $\lambda \in [1, 50]$. Assuming Poisson noise[12] for our lowest-count (noisy) dataset, which has an average count per voxel of 5.98×10^{-3} and the highest count dataset (with 1.13 per voxel; see Table 4.2), the noise levels lie largely outside the range previously studied. Only the highest count dataset aligns with the range explored in the literature [61].

Hence, we go for the range that is most interesting for us, with inputs spanning $1 \times 10^6 - 4.8 \times 10^7$ and targets using 4.8×10^7 and 9.6×10^7. To address this problem within the constraint of having independent datasets, a limited number (34) of input-target image pairs are used, as seen in Figure 6.3a.

In an idealized situation, noisy dataset realizations are independent. For instance, 1×10^6 n_{count} should not be sampled from the same region which is a subset of a 2×10^6 dataset, as overlapping events would introduce dependencies between datasets. While achieving complete independence is straightforward for lower-count datasets, it becomes increasingly challenging when sampling from the highest n_{count} of 1.86×10^8. To mitigate this issue, the maximum n_{count} are avoided during training to increase the independence of datasets.

Prior work by Lehtinen, Munkberg, Hasselgren, *et al.* demonstrated that using cleaner targets, even with fewer noisy realizations (e.g., two), significantly enhances denoising performance. Future improvements could involve incorporating other MPES datasets to increase the pool of independent images. Additional studies could also examine whether including partially dependent datasets improves denoising performance.

We employ a *patch*-based approach to getting training pairs. Firstly, to increase the training

[12] Although the noise is described to not follow the Poisson statistics in Chapter 5, it sufficies for the present argument.

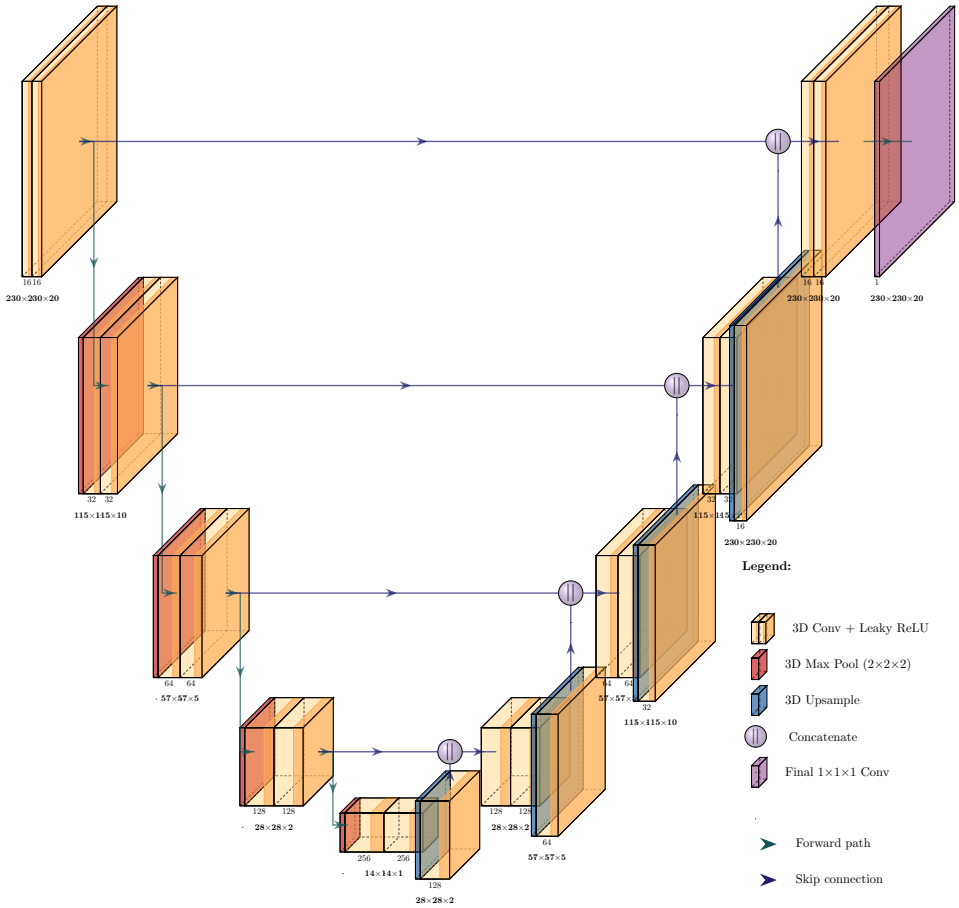


Figure 6.4: *UNET3D architecture* used for training the NN, with example inputs propogated through the network. The architecture consists of an encoder-decoder structure with skip connections. The encoder compresses spatial information using a series of convolutional layers and 3D max-pooling operations, while the decoder reconstructs the data to its original dimensions using upsampling layers and convolutional operations. Skip connections link corresponding encoder and decoder levels, allowing feature reuse and aiding gradient flow during training.

data by having many independent training pairs and secondly to make the problem computationally feasible. This approach is appointed as training on entire 3D volumes would be prohibitively expensive, making batch-based training challenging. We extract 3D patches from the 3D volumes, as shown in Figure 6.3b, generating 5967 training pairs. The patches are extracted at $60 \times 240 \times 240$ size, with a stride of $30 \times 120 \times 120$. The patches are then normalized to have zero mean and $-1-1$ ranges which follows the common preprocessing procedure that centers the data and enhances NN learning performance [70].

To further augment the data, random rotations and flips are applied to the extracted patches. This augmentation not only increases the variability in the training data but also helps the NN generalize better during inference.

6.4.2 Model Architecture: UNET3D

While in principle any denoising CNN architecture could be used, we employ the UNET architecture, a popular choice for image restoration tasks, especially in medical imaging, where it has shown remarkable success [62]. We extend the UNET architecture to train 3D data using 3D patches [71]. Due to its relatively shallow architecture in comparison to deeper models, UNET accelerates the 3D learning process.

The name, UNET, takes after the architecture's structure, a distinctive "U-shaped", as illustrated in Figure 6.4. UNET, similar to an autoencoder, consists of an encoder-decoder structure, forming the left and right halves of the "U-shape". The difference from traditional autoencoders is the presence of skip connections (purple arrow) that link corresponding encoder and decoder levels. These connections allow the network to reuse features from the encoder during the decoding process, aiding gradient flow during training. The skip connections are also particularly useful in recovering fine-grained details lost during downsampling in the encoder.

The encoder compresses spatial information using a series of convolutional layers and 3D max-pooling operations. This progressively reduces the spatial dimensions while increasing the feature channels. On the other hand, the decoder reconstructs the data to its original dimension using upsampling layers and convolution operations.

The model leverages 3D convolutions, allowing it to capture spatial features across the depth, height, and width dimensions simultaneously. Downsampling within the encoder is achieved through 3D max-pooling layers $2 \times 2 \times 2$ kernels, while the decoder employs 3D upsampling to restore spatial dimensions. The input data is processed in patches of size $60 \times 240 \times 240$ voxels[13], which are reduced and then restored in the encoder-decoder pipeline. As illustrated in Figure 6.4, intermediate feature maps progressively decrease in spatial dimensions (e.g., $230 \times 230 \times 20 \rightarrow 115 \times 115 \times 10 \rightarrow 57 \times 57 \times 5$) in the encoder and are restored in reverse order in the decoder.

The model uses Leaky ReLU activation functions, as seen in Equation (6.15), with a negative slope of 0.01, which enhances gradient flow in regions with low activation values. Within each feature channel, group normalization (of size 8) is employed to improve training stability and performance. Group normalization performs normalization independent of the batch sizes by dividing the channels into groups and computing the mean and variance within each group of the feature maps[72]. However, Batch Normalization calculates the mean and variance across the whole batch for every channel of features and is suited for larger batch sizes.

Additionally, due to memory limitations, we use a batch size of 16. Hence, within each feature, group normalization (of size 8) is applied here instead, since batch normalization is known to be unstable, particularly for small batch sizes.

The final layer of the network consists of a $1 \times 1 \times 1$ convolution, which maps the feature space to a single output channel, producing the denoised result.

6.4.3 Training and Validation

For training, the model is optimized using the Adam optimizer with an initial learning rate of 0.001 and beta values of 0.9 and 0.99. The learning rate is adjusted dynamically using a `ReduceLROnPlateau` scheduler, which reduces the rate by a factor of 0.1 if the primary evaluation metric does not improve for 10 validation runs. The training is conducted with a batch size of 16, for a maximum of 1×10^3 epochs or until 1×10^5 iterations are completed. Validation is performed every 1×10^3 iterations using SSIM as the primary metric and PSNR as evaluation metrics. This training was performed prior to the results regarding MS-SSIM in Appendix B.2. Therefore, while SSIM was used here, a better validation could be obtained using MS-SSIM in future work.

For validation, we split our 3D dataset in Figure Figure 4.5b with a 80-20 ratio along the E dimension, with the top 80% used for training[14] and the remaining bottom 20% for validation. The key features of the dataset exist in the training set, however, due to limited data availability, the indistinguishable features in the bottom 20% are employed for validation. This high count dataset represents the validation target. Subsequently, the 3D patches from the 3D dataset

[13]This is just how we train the data but CNNs can process any arbitrary resolution, provided sufficient memory. In fact, during inference, we employ a larger patch size.

[14]The top most energy represents the E_F.

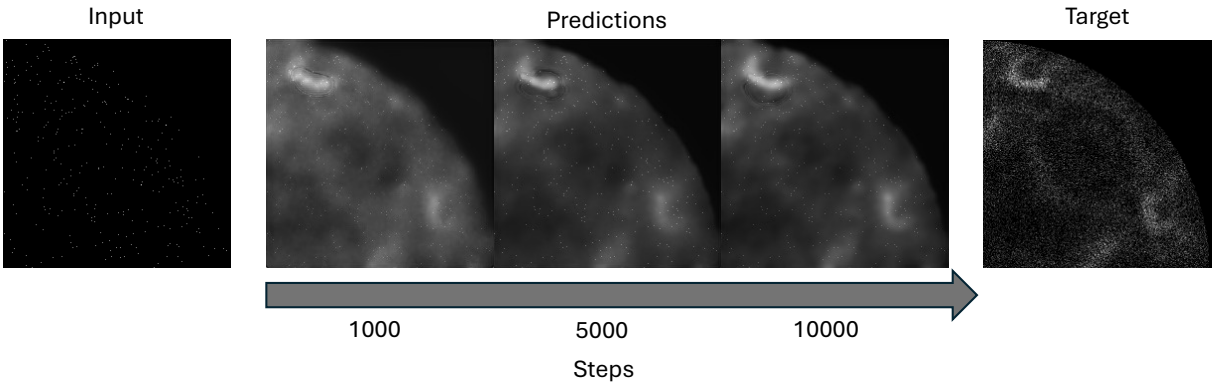


Figure 6.5: *Denoising predictions during training* at step 1000, 5000 and 10 000 are shown, along with the input and target images. The key features, though blurry, are quickly discernible in the denoised images, even at early stages of training.

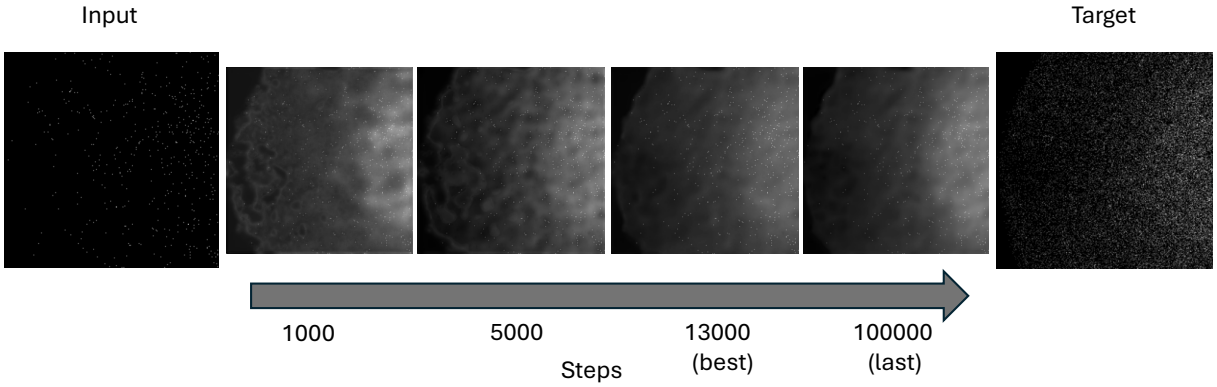


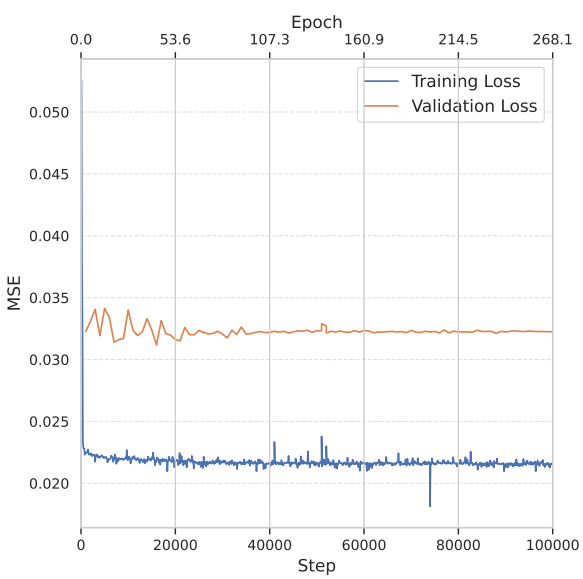
Figure 6.6: *Denoising predictions during validation* at step 1000, 5000, 13 000 and 100 000 are shown, the latter two of which correspond to the best and last models. The input and target images are also shown. The validation set features not as discernible as the training set, as they correspond to energies much above the Fermi level.

are split with the same 80-20 ratio and these validations are compared against the high count target validation. The validation set is used to monitor the model's performance and prevent overfitting.

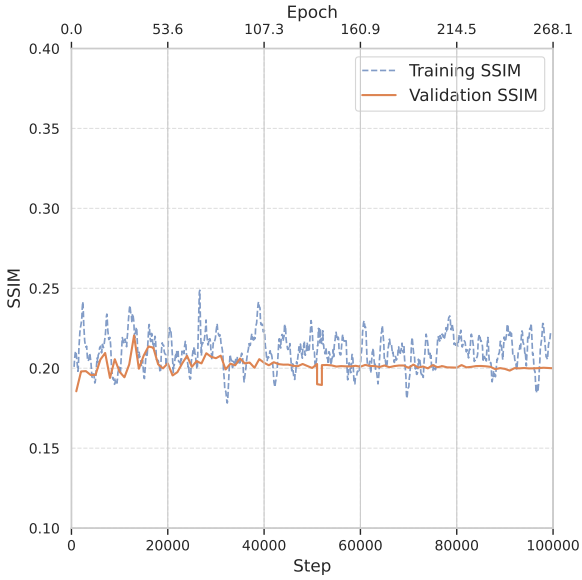
The trained models are saved after each validation run, with the previous version being overwritten to maintain the latest state. Additionally, the model achieving the highest validation score, as measured by SSIM, is separately saved to ensure the preservation of the best-performing version.

The training and validation loss are shown in Figure 6.7a. It has already been exemplified in [61] that the training loss does not decrease during training, as the network can not possibly learn to transform one instance of noise to another, as the training asks for the impossible. Hence, other than the initial decrease, the network loss stays plateaued consistently across 1×10^5 steps. An example of the model's predictions during training is shown in Figure 6.5, illustrating the output for a single (training) input across different training iterations. Notably, the denoising performance exhibits perceptually significant improvements even in the early stages of training.

Since we employ the validation target to be the highest count dataset (1.86×10^8), the expectation would be that it reports higher loss values. But as discussed earlier, the validation set data is much below E_F, with features not as well-defined as near E_F. Nonetheless, the validation loss is also shown to converge at 4×10^4 steps in Figure 6.7a.



(a) Training and validation loss. The gap exists due to the validation set being compared with higher quality targets. The validation loss converges to a stable value at 4×10^4 steps.



(b) SSIM scores across training and validation. The scores show constant fluctuations around an average value during training but show less fluctuations during validation, where validation also converges to a stable value at 4×10^4 steps.

Figure 6.7: *Training validation losses and SSIM scores* (a) Training and validation loss and (b) SSIM scores over training steps and epochs. The UNET3D architecture is trained, using noisy input-targets pairs.

The metric assessment, using SSIM, depicted in Figure 6.7b, shows constant fluctuations around an average value during training but is seen to converge in the validation. Therefore, two models are proposed: The (*best*) model with the highest validation SSIM of 0.23 that happens early on during training, after about 25 epochs[15], and the (*last*) model at end of training, where the validation SSIM has converged to 0.2.

Figure 6.8 demonstrates the predictive capability of the trained model $\hat{h}$, predicted using larger patch sizes[16]. While smaller patches are utilized during training due to the computational demands of batch-based learning, larger patches can be processed during inference, as illustrated in the figure. The predictions shown are derived from the 4×10^6 count dataset using the best model.

Similarly, in Figure 6.9, the noisy and denoised k_y-k_x cuts for the Gr/Ir(111) dataset can be seen for different noise (count) levels. The noisy images show no discernible features at n_{count} of $< 8 \times 10^6$, while the denoised images start resolving clear features similar to the target. The model evaluated on the training set hence shows promising results. To really be sure that the model achieved generalized capability of denoising the experimental data, and not just learned the features of the training set, we need to evaluate it on independent test data.

6.4.4 Evaluating Model Denoising Performance

We use the Gd/W(110) dataset for evaluating the model's generalization capability. Figure 6.10 shows a k_x–E slice from noisy datasets with different n_{count}. The figure shows the noisy and denoised images for best and last models, as well as a comparison with the BM3D–Anscombe denoising. The plot also show the slice-summed image with w of 10 for noisy image, showcasing

[15] Meaning, the training set passes 25 times through the network to improve predictions, during training.
[16] Only 8 patches are employed.

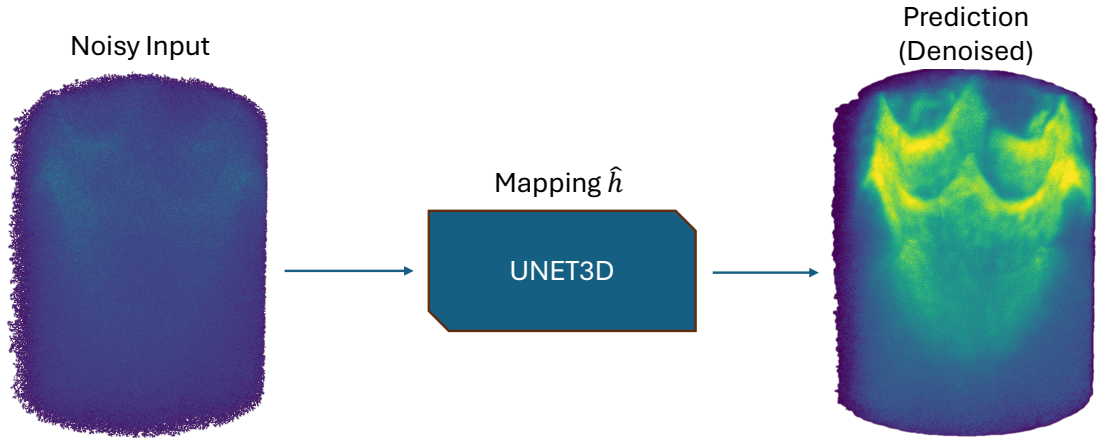


Figure 6.8: *Example prediction* (forward pass) from 4×10^6 count dataset, using the best model. The prediction resolves the key features in the noisy input, showing the effectiveness of the denoising model.

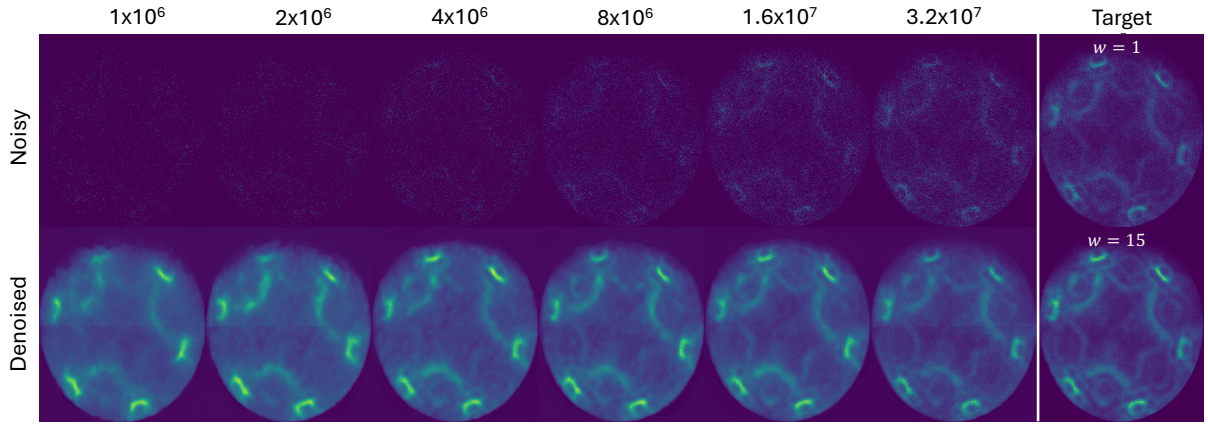


Figure 6.9: *Training set example noisy and denoised images:* k_y-k_x cuts with window size $w = 1$ shown for Gr/Ir(111) dataset. Each column corresponds to 1×10^6, 2×10^6, 4×10^6, 8×10^6, 1.6×10^7, 3.2×10^7 and 1.86×10^8 counts, respectively; where the last column is the target image with $w = 1$ in row 1 and $w = 15$ in row 2. Below 8×10^6 counts, none of the features are discernible for noisy images, while the denoised images show clear features similar to the target.

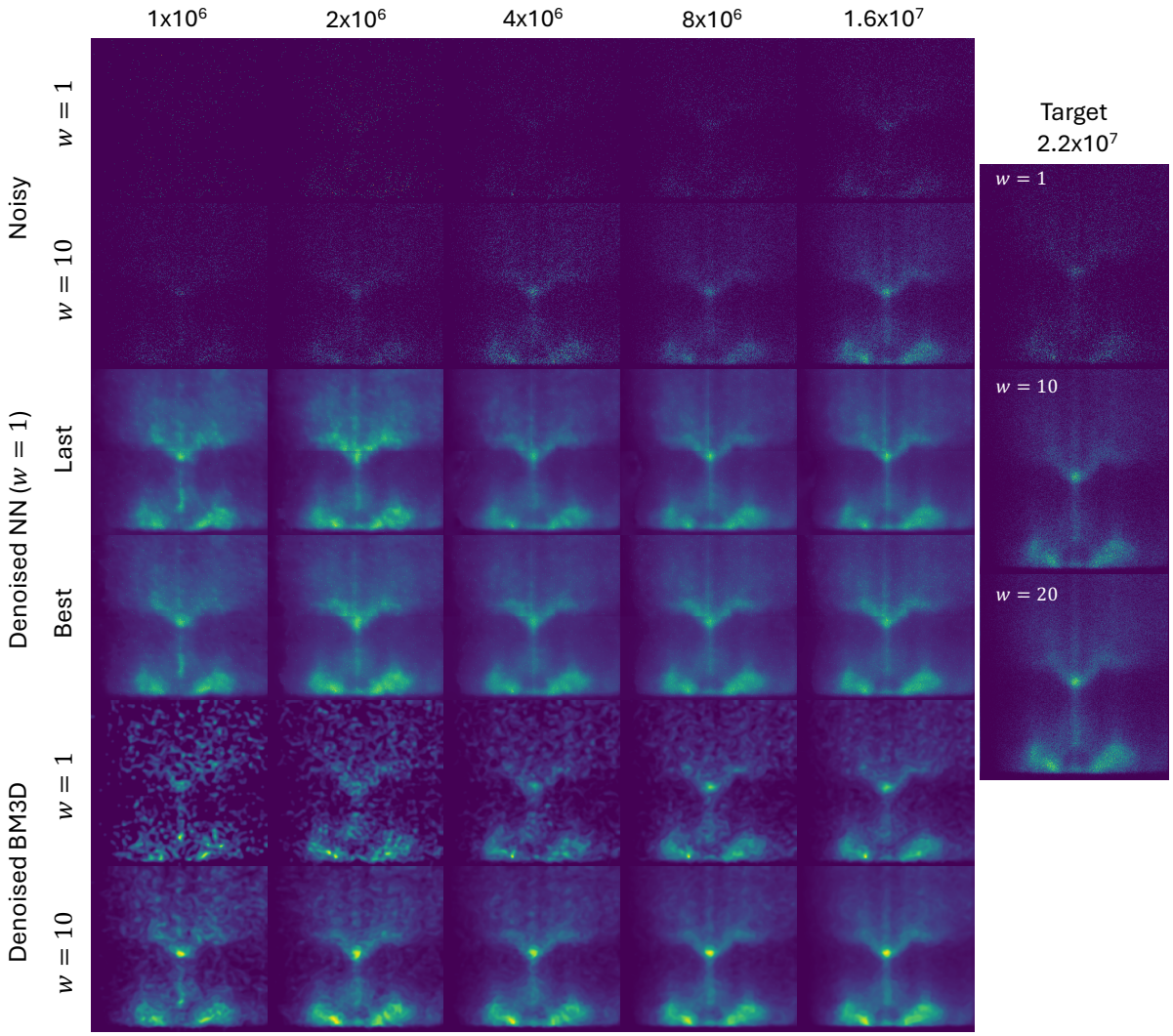


Figure 6.10: *Comparison of k_x–E slices for various n_{count}.* The target image is shown for a single-slice and for slice-summed images with w values of 10 and 20, demonstrating the effect of summing slices for visual quality improvement. The noisy image is also depicted as a slice-summed result, emphasizing that slice-summing acts as a basic form of noise reduction. Denoised results are shown for the best and last trained NN models ($w = 1$), alongside BM3D–Anscombe denoising applied with the optimal parameter σ_{o} for $w = 10$ and the same parameter used for $w = 1$. The results illustrate that the NN outperforms BM3D–Anscombe denoising at lower counts (1×10^6). At higher counts and when slices are summed, BM3D's performance improves relative to the NN.

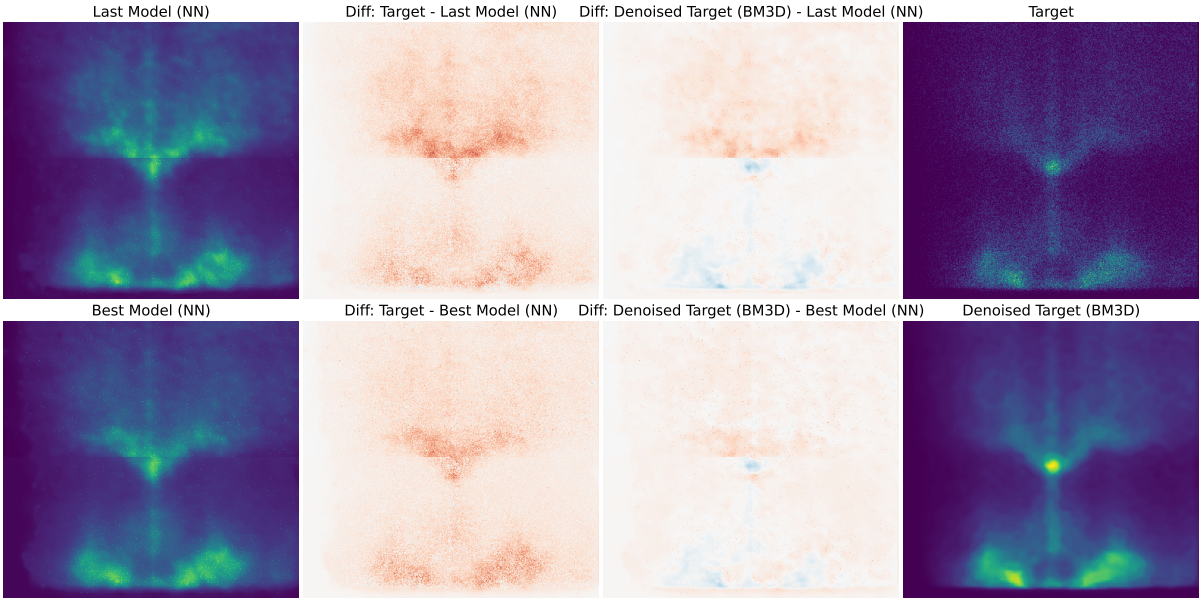


Figure 6.11: *Difference images comparing the best and last models for* n_{count} 2×10^6. The differences are computed relative to two references: the slice-summed target image with w of 10, and its BM3D-denoised version. The red regions indicate where the prediction is brighter than the noisy target, while blue regions indicate where the target is brighter. The target itself is noisy, so when subtracting the denoised outputs from the target, mostly red regions are visible. However, when using the BM3D-denoised target as a reference (assumed to approximate the correct image), the differences are mostly white (close to 0), with some blue regions where predicted features are less intense compared to the reference. These difference plots highlight that the best model preserves features more effectively and aligns better with the BM3D-denoised target compared to the last model.

the effect of improved statistics with slice-sums. The BM3D denoising results are shown as reference (using the optimal sigma) to compare against the NN denoised. Figure B.5 shows the k_y–k_x slice for the same models and BM3D–Anscombe denoising.

From the figure, it can be seen that the NN-based denoising performs much better at lower counts (even at 1×10^6) than BM3D based denoising. At higher counts and summed-slices, the BM3D denoising also has good performance, a conclusion we already drew in Chapter 4.

To see if slice-summing has any improvement in denoising performance of the NN (as it had with BM3D), the last NN model predictions are also slice-summed and shown in Figure B.6. It can be seen that slice-summing does not have a significant impact on denoising performance as the features are already well resolved in the individual slices. As with slice-summing earlier, this actually leads to feature blurring. This can be explained by the fact that through the usage of a 3D CNN, the 3 dimensional correlations across the depth dimension are already considered, and hence the slice-summing does not provide any additional information.

Let us now explore whether the longer training time to obtain the last model is worthwhile, since the best model is obtained earlier on in the training. To explore this, we can look at the difference images between the predicted output and target image. Since the target is noisy, we look at slice-summed images with w of 10 to get a better idea of the features. Furthermore, we also compare the best and last models with a BM3D denoised target image. Since BM3D is performing well at these counts, it can be used as a reference for the NN-denoising performance.

The difference images are shown in Figure 6.11 and Figure B.7 for 2×10^6 and 8×10^6 counts, respectively. Comparing with the target, both models show unsharp features at Figure 6.11, with the last model performing worse. Whereas, in Figure B.7, the predictions approach much closer to the target (more zero intensity).

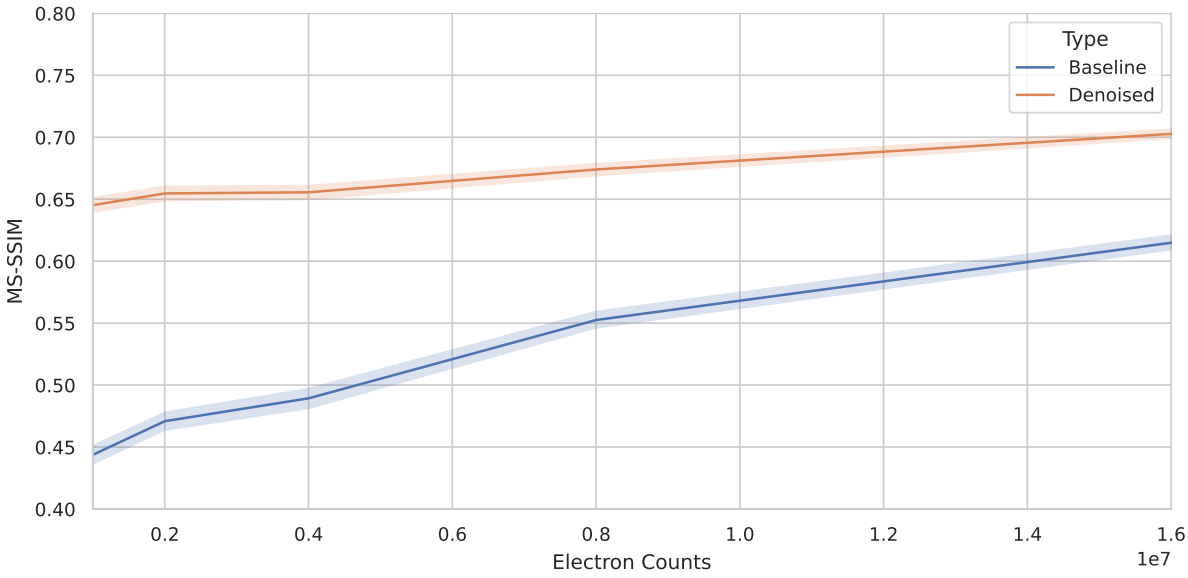


Figure 6.12: *Denoising performance of the best neural network model* as a function of n_{count}, measured using MS-SSIM. The comparison is done using slice-summed targets with $w = 10$. The baseline metric is computed using the noisy image as input. Different from BM3D, the NN denoising performance stays relatively consistent across the n_{count}, and is especially effective at lower counts.

When looking at MS-SSIM scores, the best models report better values for lower counts, i.e., best 0.68 vs. last 0.65 at n_{count} of 2×10^6, and both best and last with 0.73 at n_{count} of 8×10^6. Both best and last model produce similar difference images at n_{count} of 8×10^6, highlighting they only differ in the lower count regime. Since we are especially interested in lower counts denoising, it is not clear if the longer training time is justified, and further investigation is needed. It should also be noted that the target image is not as representative of clean target (with only count n_{count} of 2.21×10^7) when compared with the Gr/Ir(111) target image (n_{count} of 1.86×10^8).

At the end, we look at how the MS-SSIM scores improves with increasing counts, similar to how it is done earlier with BM3D in Figure 4.14. While we only look at the best model here, the same could be done for the last model. The MS-SSIM scores are shown in Figure 6.12, where the comparison is done between single-slice denoised images with a slice-summed target image ($w = 10$). Perceptually (see Figure B.6) the denoising performance with NN far exceeds that of BM3D at low counts, even when denoising slice-summed images with BM3D. The NN denoising performance is also better than BM3D at higher counts for single-slices, and the slice-summed images at high counts ($\geq 8 \times 10^6$) are just as good, or better.

7. Conclusion and Outlook

This thesis has presented an investigation into image denoising methodologies within the scope of MPES data, addressing inherent limitations posed by the probabilistic nature of photoemission events, the multidimensional sample spaces, and compounded by the experimental constraints such as space-charge effect, sample degradation from radiation damage and practical constraints such as limited beamtime allocations. These challenges are particularly pronounced in modern MPES experiments leveraging light sources such as FELs and HHG lasers that present unique statistical and operational challenges.

The work initially considered the block-matching and 3D filtering (BM3D), with and without variance stabilization through the Anscombe transform. While these methods performed quite well for high-count data, the performance degrades considerably in the low-count regimes. Moreover, reflecting the inability to exploit the multidimensional correlations of the MPES data. Although perceptual improvements were observed with the Anscombe transform compared to solely using BM3D, the lack of clean target datasets rendered quantitative assessment of this effect challenging. We showed that MS-SSIM is a better evaluation metric, compared to other reference-based metrics, when the target image is noisy. However, domain-specific evaluation metrics, tailored to the physics of MPES such as line shape analysis of momentum and energy distribution curves, could prove more beneficial.

This work has also contributed to the statistical analysis of photoemitted electrons, with an emphasis on electron counting distributions under FEL illumination. Our analysis confirmed that the stochastic SASE process is described by a doubly stochastic Poisson point process giving rise to negative binomial counting statistics. The results presented here highlights the importance of revising the commonly assumed Poisson model within photoemission data and reevaluating models for variance stabilization and noise assessment in MPES, especially when utilizing nonlinear HHG sources or FELs. These insights can be extended beyond MPES to other disciplines reliant on similar photon-based counting experiments.

In order to overcome the constraints of traditional techniques within the extremely low-count regime, this thesis focused on deep learning methodologies, specifically the UNET3D architecture trained within the self-supervised learning paradigm, Noise2Noise. Exploiting the correlations intrinsic to multidimensional data and using single-event datasets in order to create training examples, this model achieves remarkable improvements in denoising in sparse MPES datasets. We showed that for as little as 1×10^{-3} average-counts per voxel, the model predictions (corresponding to a 1×10^6 dataset) exceed higher count BM3D-denoised images.

Advancements in variance stabilization techniques and their inversions for NB noise, paired with an algorithm such as BM4D, leveraging 3D data, could prove useful when faced with no training data. The deep learning training would benefit from retraining using the MS-SSIM metric of evaluation and if possible, clean images. It would be interesting to incorporate other experimental setup datasets, and try pretrained networks, or explore unsupervised approaches such as Deep Image Prior and Noise2Void.

The impact of this work could enable more efficient data acquisition and analysis. The methods developed here streamlining experiments at large-scale facilities such as FELs or table-top laboratory setups, optimizing beamtime usage and experimental parameter space that can be explored more efficiently and thus also expanded. In addition, the knowledge about photon counting statistics and denoising methods can potentially be applied to other experiments, such as X-ray diffraction, scattering, and other multidimensional spectroscopies. These methods are

also of particular interest for applications outside the field involved with sparse, noisy datasets, such as medical imaging, astronomy, and particle physics.

Despite its successes, this work faced several limitations, including restricted test datasets and the lack of clean images for metric evaluation. In a future study, these limitations should be addressed, which could explore validation with domain-specific metrics, and expand datasets to a wider range of noise levels and experimental conditions.

While this work has addressed several critical challenges in MPES data processing, future studies should make considerations for designing specialized experiments which directly study the statistical properties of photoemitted electrons at FEL and HHG sources. By carefully varying acquisition parameters, including photon energy, pulse duration, and repetition rate, the relationship between counting statistics and photoemission processes could be probed in greater depth. These experiments could provide insights into the photoemission statistics, revealing subtle many-body effects, electron correlation phenomena, or nonlinear interactions specific to the photoemission process.

Bibliography

[1] W. Heisenberg, C. Eckart, F. C. Hoyt, and W. Heisenberg, *The Physical Principles of the Quantum Theory* (Dover Books on Physics), Nachdr. Mineola, NY: Dover Publ, 2009, ISBN: 978-0-486-60113-7.

[2] J. J. Sakurai and J. Napolitano, *Modern Quantum Mechanics*, 3rd ed. Cambridge University Press, Sep. 2020, ISBN: 978-1-108-58728-0 978-1-108-47322-4. DOI: 10.1017/9781108587280.

[3] J. Binney and D. B. Skinner, *The Physics of Quantum Mechanics*. Oxford: Oxford university press, 2014, ISBN: 978-0-19-968857-9.

[4] M. Cardona and L. Ley, Eds., *General Principles* (Photoemission in Solids / Ed. by M. Cardona 1). Berlin: Springer, 1978, ISBN: 978-3-662-30919-3 978-3-540-08685-7 978-0-387-08685-9.

[5] W. Feller, *An Introduction to Probability Theory and Its Applications* (Wiley Series in Probability and Mathematical Statistics), Third ed. rev. New York Chichester Brisbane [etc.]: J. Wiley, 1991, ISBN: 978-0-471-25708-0.

[6] J. Maklar, S. Dong, S. Beaulieu, *et al.*, "A quantitative comparison of time-of-flight momentum microscopes and hemispherical analyzers for time- and angle-resolved photoemission spectroscopy experiments," *Review of Scientific Instruments*, vol. 91, no. 12, p. 123 112, Dec. 2020, ISSN: 0034-6748. DOI: 10.1063/5.0024493.

[7] B. Schönhense, K. Medjanik, O. Fedchenko, *et al.*, "Multidimensional photoemission spectroscopy—the space-charge limit," *New Journal of Physics*, vol. 20, no. 3, p. 033 004, Mar. 2018, ISSN: 1367-2630. DOI: 10.1088/1367-2630/aaa262.

[8] V. Shokeen, M. Heber, D. Kutnyakhov, *et al.*, "Real-time observation of non-equilibrium phonon-electron energy and angular momentum flow in laser-heated nickel," *Science Advances*, vol. 10, no. 5, eadj2407, Feb. 2024, ISSN: 2375-2548. DOI: 10.1126/sciadv.adj2407.

[9] T. Sjodin, H. Petek, and H.-L. Dai, "Ultrafast Carrier Dynamics in Silicon: A Two-Color Transient Reflection Grating Study on a (111) Surface," *Physical Review Letters*, vol. 81, no. 25, pp. 5664–5667, Dec. 1998, ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.81.5664.

[10] H. Petek and S. Ogawa, "Femtosecond time-resolved two-photon photoemission studies of electron dynamics in metals," *Progress in Surface Science*, vol. 56, no. 4, pp. 239–310, Dec. 1997, ISSN: 00796816. DOI: 10.1016/S0079-6816(98)00002-1.

[11] C. Laulhé, T. Huber, G. Lantz, *et al.*, "Ultrafast Formation of a Charge Density Wave State in 1 T - TaS 2 : Observation at Nanometer Scales Using Time-Resolved X-Ray Diffraction," *Physical Review Letters*, vol. 118, no. 24, p. 247 401, Jun. 2017, ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.118.247401.

[12] A. Einstein, "Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt," *Annalen der Physik*, vol. 322, no. 6, pp. 132–148, 1905. DOI: 10.1002/andp.19053220607.

[13] M. Fox, *Quantum Optics: An Introduction* (Oxford Master Series in Physics 15). Oxford: Oxford University Press, 2006, ISBN: 978-0-19-856672-4.

[14] J. J. Macklin, J. D. Kmetec, and C. L. Gordon, "High-order harmonic generation using intense femtosecond pulses," *Physical Review Letters*, vol. 70, no. 6, pp. 766–769, Feb. 1993, ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.70.766.

[15] W. Ackermann, G. Asova, V. Ayvazyan, *et al.*, "Operation of a free-electron laser from the extreme ultraviolet to the water window," *Nature Photonics*, vol. 1, no. 6, pp. 336–342, Jun. 2007, ISSN: 1749-4893. DOI: 10.1038/nphoton.2007.76.

[16] M Svandrlik, E Allaria, L Badano, *et al.*, "Development Perspectives at FERMI," *Free Electron Lasers*, 2017.

[17] K. Tiedtke, A. Azima, N. von Bargen, *et al.*, "The soft x-ray free-electron laser FLASH at DESY: Beamlines, diagnostics and end-stations," *New Journal of Physics*, vol. 11, no. 2, p. 023 029, Feb. 2009. DOI: 10.1088/1367-2630/11/2/023029.

[18] B. Faatz, E. Plönjes, S. Ackermann, *et al.*, "Simultaneous operation of two soft x-ray free-electron lasers driven by one linear accelerator," *New Journal of Physics*, vol. 18, no. 6, p. 062 002, Jun. 2016, ISSN: 1367-2630. DOI: 10.1088/1367-2630/18/6/062002.

[19] D. Kutnyakhov, R. P. Xian, M. Dendzik, *et al.*, "Time- and momentum-resolved photoemission studies using time-of-flight momentum microscopy at a free-electron laser," *Review of Scientific Instruments*, vol. 91, no. 1, p. 013 109, Jan. 2020, ISSN: 0034-6748, 1089-7623. DOI: 10.1063/1.5118777.

[20] G. Schönhense, K. Medjanik, and H.-J. Elmers, "Space-, time- and spin-resolved photoemission," *Journal of Electron Spectroscopy and Related Phenomena*, Special Anniversary Issue: Volume 200, vol. 200, pp. 94–118, Apr. 2015, ISSN: 0368-2048. DOI: 10.1016/j.elspec.2015.05.016.

[21] J. Ladislas Wiza, "Microchannel plate detectors," *Nuclear Instruments and Methods*, vol. 162, no. 1, pp. 587–601, Jun. 1979, ISSN: 0029-554X. DOI: 10.1016/0029-554X(79)90734-1.

[22] R. Paschotta, "Microchannel Plates - an encyclopedia article," in *RP Photonics Encyclopedia*, RP Photonics AG, 2019. DOI: 10.61835/u9s.

[23] J. Correa, A. Ignatenko, D. Pennicard, *et al.*, "TEMPUS, a Timepix4-based system for the event-based detection of X-rays," *Journal of Synchrotron Radiation*, vol. 31, no. 5, pp. 1209–1216, Sep. 2024, ISSN: 1600-5775. DOI: 10.1107/S1600577524005319.

[24] A. Oelsner, O. Schmidt, M. Schicketanz, *et al.*, "Microspectroscopy and imaging using a delay line detector in time-of-flight photoemission microscopy," *Review of Scientific Instruments*, vol. 72, no. 10, pp. 3968–3974, Oct. 2001, ISSN: 0034-6748. DOI: 10.1063/1.1405781.

[25] A. Oelsner, M. Rohmer, C. Schneider, D. Bayer, G. Schönhense, and M. Aeschlimann, "Time- and energy resolved photoemission electron microscopy-imaging of photoelectron time-of-flight analysis by means of pulsed excitations," *Journal of Electron Spectroscopy and Related Phenomena*, vol. 178–179, pp. 317–330, May 2010, ISSN: 03682048. DOI: 10.1016/j.elspec.2009.10.008.

[26] M. Heber, "Studies on ultrafast dynamics in correlated electron systems with time- and angle-resolved photoemission spectroscopy," Ph.D. dissertation, University of Hamburg, Hamburg, 2024.

[27] M. Knipfer, S. Meier, T. Volk, J. Heimerl, P. Hommelhoff, and S. Gleyzer, "Deep learning-based spatiotemporal multi-event reconstruction for delay line detectors," *Machine Learning: Science and Technology*, vol. 5, no. 2, p. 025 019, Apr. 2024, ISSN: 2632-2153. DOI: 10.1088/2632-2153/ad3d2d.

[28] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007, ISSN: 1941-0042. DOI: 10.1109/TIP.2007.901238.

[29] A. Buades, B. Coll, and J. M. Morel, "A Review of Image Denoising Algorithms, with a New One," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, Jan. 2005, ISSN: 1540-3459, 1540-3467. DOI: 10.1137/040616024.

[30] M. Diwakar and M. Kumar, "A review on CT image noise and its denoising," *Biomedical Signal Processing and Control*, vol. 42, pp. 73–88, Apr. 2018, ISSN: 1746-8094. DOI: 10.1016/j.bspc.2018.01.010.

[31] A. V. Oppenheim and G. C. Verghese, *Signals, Systems & Inference*, Global edition. Harlow: Pearson, 2017, ISBN: 978-1-292-15620-0.

[32] L. P. Yaroslavsky, *Digital Picture Processing. An Introduction* (Springer Series in Information Sciences 9), With 87 figures. Title of the original russian edition: Vvedynie v Tsifrovuyu Obrabotku Izebrazheny, Moscow, 1979. Berlin: Springer, 1985, ISBN: 978-3-540-11934-0.

[33] D. Wang, J. Xu, and K. Xu, "An FPGA-Based Hardware Accelerator for Real-Time Block-Matching and 3D Filtering," *IEEE Access*, vol. 8, pp. 121 987–121 998, 2020, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3006773.

[34] Maggioni M, Katkovnik V, Egiazarian K, and Foi A, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *PubMed*,

[35] R. M. Bell and Y. Koren, "Lessons from the Netflix prize challenge," *ACM SIGKDD Explorations Newsletter*, vol. 9, no. 2, pp. 75–79, Dec. 2007, ISSN: 1931-0145, 1931-0153. DOI: 10.1145/1345448.1345465.

[36] G. Dror, N. Koenigstein, Y. Koren, and M. Weimer, "The Yahoo! Music Dataset and KDD-Cup'11," in *Proceedings of KDD Cup 2011*, PMLR, Jun. 2012, pp. 3–18.

[37] M Makitalo and A Foi, "Optimal Inversion of the Anscombe Transformation in Low-Count Poisson Image Denoising," *IEEE Transactions on Image Processing*, vol. 20, no. 1, pp. 99–109, Jan. 2011, ISSN: 1057-7149. DOI: 10.1109/TIP.2010.2056693.

[38] Y. Kim, D. Oh, S. Huh, *et al.*, "Deep learning-based statistical noise reduction for multidimensional spectral data," *Review of Scientific Instruments*, vol. 92, no. 7, p. 073 901, Jul. 2021, ISSN: 0034-6748. DOI: 10.1063/5.0054920.

[39] M. S. Bartlett, "The Square Root Transformation in Analysis of Variance," *Supplement to the Journal of the Royal Statistical Society*, vol. 3, no. 1, pp. 68–78, 1936, ISSN: 1466-6162. DOI: 10.2307/2983678. JSTOR: 2983678.

[40] F. J. Anscombe, "The Transformation of Poisson, Binomial and Negative-Binomial Data," *Biometrika*, vol. 35, no. 3/4, p. 246, Dec. 1948, ISSN: 00063444. DOI: 10.2307/2332343. JSTOR: 2332343.

[41] M. Makitalo and A. Foi, "A Closed-Form Approximation of the Exact Unbiased Inverse of the Anscombe Variance-Stabilizing Transformation," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2697–2698, Sep. 2011, ISSN: 1941-0042. DOI: 10.1109/TIP.2011.2121085.

[42] M. Heber, N. Wind, D. Kutnyakhov, *et al.*, "Multispectral time-resolved energy–momentum microscopy using high-harmonic extreme ultraviolet radiation," *Review of Scientific Instruments*, vol. 93, no. 8, p. 083 905, Aug. 2022, ISSN: 0034-6748, 1089-7623. DOI: 10.1063/5.0091003.

[43] D. Kutnyakhov, *Multidimensional photoemission spectra of Gd/W(110)*, Feb. 2024. DOI: 10.5281/ZENODO.10658470.

[44] J. Maklar, S. Dong, S. Beaulieu, *et al.*, *Time-resolved ARPES RAW data of bulk WSe2 for a quantitative comparison of time-of-flight momentum microscopes and hemispherical analyzers: RAW MM data*, Mar. 2022. DOI: 10.5281/ZENODO.6369727.

[45] R. P. Xian, Y. Acremann, S. Y. Agustsson, *et al.*, "An open-source, end-to-end workflow for multidimensional photoemission spectroscopy," *Scientific Data*, vol. 7, no. 1, p. 442, Dec. 2020, ISSN: 2052-4463. DOI: 10.1038/s41597-020-00769-8.

[46] A. Eskicioglu and P. Fisher, "Image quality measures and their performance," *IEEE Transactions on Communications*, vol. 43, no. 12, pp. 2959–2965, Dec. 1995, ISSN: 1558-0857. DOI: 10.1109/26.477498.

[47] Lin Zhang, Lei Zhang, Xuanqin Mou, and D. Zhang, "FSIM: A Feature Similarity Index for Image Quality Assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011, ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2011.2109730.

[48] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2, Nov. 2003, 1398–1402 Vol.2. DOI: 10.1109/ACSSC.2003.1292216.

[49] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization,"

[50] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 2623–2631, ISBN: 978-1-4503-6201-6. DOI: 10.1145/3292500.3330701.

[51] L. Mandel, "Fluctuations of Photon Beams and their Correlations," *Proceedings of the Physical Society*, vol. 72, no. 6, p. 1037, Dec. 1958, ISSN: 0370-1328. DOI: 10.1088/0370-1328/72/6/312.

[52] L. Mandel, "Fluctuations of Photon Beams: The Distribution of the Photo-Electrons," *Proceedings of the Physical Society*, vol. 74, no. 3, p. 233, Sep. 1959, ISSN: 0370-1328. DOI: 10.1088/0370-1328/74/3/301.

[53] S. N. Chiu, D. Stoyan, W. S. Kendall, *et al.*, Eds., *Stochastic Geometry and Its Applications* (Wiley Series in Probability and Statistics), 3. Aufl., 1. publ. Chichester: Wiley, 2013, ISBN: 978-0-470-66481-0.

[54] B. Saleh, *Photoelectron Statistics* (Springer Series in Optical Sciences), D. L. MacAdam, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, vol. 6, ISBN: 978-3-662-13483-2 978-3-540-37311-7. DOI: 10.1007/978-3-540-37311-7.

[55] C. Mehta, "VIII Theory of Photoelectron Counting," in *Progress in Optics*, vol. 8, Elsevier, 1970, pp. 373–440, ISBN: 978-0-444-10020-7. DOI: 10.1016/S0079-6638(08)70193-8.

[56] E. L. Saldin, E. A. Schneidmiller, and M. V. Yurkov, "Statistical properties of radiation from VUV and X-ray free electron laser," *Optics Communications*, vol. 148, no. 4, pp. 383–403, Mar. 1998, ISSN: 0030-4018. DOI: 10.1016/S0030-4018(97)00670-6.

[57] J. Heimerl, A. Mikhaylov, S. Meier, *et al.*, "Multiphoton electron emission with non-classical light," *Nature Physics*, vol. 20, no. 6, pp. 945–950, Jun. 2024, ISSN: 1745-2481. DOI: 10.1038/s41567-024-02472-6.

[58] A. Gorlach, O. Neufeld, N. Rivera, O. Cohen, and I. Kaminer, "The quantum-optical nature of high harmonic generation," *Nature Communications*, vol. 11, no. 1, p. 4598, Sep. 2020, ISSN: 2041-1723. DOI: 10.1038/s41467-020-18218-w.

[59] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep Image Prior," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1867–1888, Mar. 2020, ISSN: 1573-1405. DOI: 10.1007/s11263-020-01303-4.

[60] A. Krull, T.-O. Buchholz, and F. Jug, *Noise2Void - Learning Denoising from Single Noisy Images*, 2018. DOI: 10.48550/ARXIV.1811.10980.

[61] J. Lehtinen, J. Munkberg, J. Hasselgren, *et al.*, "Noise2Noise: Learning Image Restoration without Clean Data," in *Proceedings of the 35th International Conference on Machine Learning*, PMLR, Jul. 2018, pp. 2965–2974.

[62] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, https://arxiv.org/abs/1505.04597v1.

[63] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, 1st ed. Cambridge University Press, May 2014, ISBN: 978-1-107-05713-5 978-1-107-29801-9. DOI: 10.1017/CBO9781107298019.

[64] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning* (Springer Texts in Statistics). New York, NY: Springer New York, 2013, vol. 103, ISBN: 978-1-4614-7137-0 978-1-4614-7138-7. DOI: 10.1007/978-1-4614-7138-7.

[65] R. Tibshirani, J. Friedman, and T. Hastie, *The Elements of Statistical Learning*.

[66] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (Adaptive Computation and Machine Learning). Cambridge, Massachusetts: The MIT Press, 2016, ISBN: 978-0-262-03561-3.

[67] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York: Springer, 2006, ISBN: 978-0-387-31073-2.

[68] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*, May 2013.

[69] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, Jan. 2017. DOI: 10.48550/arXiv.1412.6980. arXiv: 1412.6980.

[70] C. M. Bishop and H. Bishop, *Deep Learning: Foundations and Concepts*. Cham, Switzerland: Springer, 2024, ISBN: 978-3-031-45467-7. DOI: 10.1007/978-3-031-45468-4.

[71] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation*, Jun. 2016. DOI: 10.48550/arXiv.1606.06650. arXiv: 1606.06650 [cs].

[72] Y. Wu and K. He, *Group Normalization*, 2018. DOI: 10.48550/ARXIV.1803.08494.

[73] S. Hoyer and J. Hamman, "Xarray: N-D labeled Arrays and Datasets in Python," *Journal of Open Research Software*, vol. 5, no. 1, p. 10, Apr. 2017, ISSN: 2049-9647. DOI: 10.5334/jors.148.

[74] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, May 2007, ISSN: 1558-366X. DOI: 10.1109/MCSE.2007.55.

[75] The pandas development team, *Pandas-dev/pandas: Pandas*, Zenodo, Apr. 2024. DOI: 10.5281/ZENODO.3509134.

[76] M. Waskom, "Seaborn: Statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, Apr. 2021, ISSN: 2475-9066. DOI: 10.21105/joss.03021.

[77] A. Paszke, S. Gross, F. Massa, *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.

[78] A. Wolny, L. Cerrone, A. Vijayan, *et al.*, "Accurate and versatile 3D segmentation of plant tissues at cellular resolution," *eLife*, vol. 9, e57613, Jul. 2020, ISSN: 2050-084X. DOI: 10.7554/eLife.57613.

[79] T. Barry, *Tim barry: Gamma, poisson, and negative binomial distributions*, 2020.

Acknowledgements

This thesis would not have been possible without the constant support and guidance of the many people that supported me through the process. Though extremely challenging in the breath and depth I tried to uptake, it allowed me to explore many directions of research and allowed me to learn a lot, and for that I am thankful.

I would like to especially express my gratitude to my supervisors B. Berkels and Kai, and Dima, my acting supervisor throughout the thesis. I am very thankful to B. Berkels for accepting to supervise this external research project. I appreciate the commitment made to constantly guide me in the right direction, especially in the fields of image denoising and mathematics relating. I am thankful to Kai for the detailed discussions on exploring different physics, and supporting me in presenting my research. I would especially like to thank Dima for the consistent guidance throughout this project, for helping me understand all the details surrounding the experimental aspects of MPES, and for always being very understanding.

Naturally, no work is complete without the help of those close and dear. I extend my heartfelt thanks to the closest for the consistent support through the fun and the tough times. Shoutout to Kumarah who stayed up late nights helping me with corrections, in a domain he had little clue of.

I also extend my gratitude to Martin Burger's group. Especially to Lorenz Kruger for the insightful conversations trying to explore the data, its statistical implications and more.

This contribution was made possible through the generosity of the authors who shared their datasets: groups from DESY/FLASH, University of Mainz, FU Berlin, ETH Zürich for the Gd/W(110) dataset [43], Heber, Wind, Kutnyakhov, *et al.* and groups from ETH Zürich, DESY/NanoLab and Aarhus University for the Gr/Ir(111) dataset [42], and Maklar, Dong, Beaulieu, *et al.* for the WSe_2 dataset [44].

We acknowledge DESY (Hamburg, Germany), a member of the Helmholtz Association HGF, for the provision of experimental facilities. Parts of this research were carried out at PG2 beamline of FLASH. This research was also supported in part through the Maxwell computational resources operated at DESY, Hamburg, Germany.

The `Python` data science and visualization ecosystem was heavily employed in this thesis. The usage of `Xarray` [73] for multidimensional dataset transformations, `matplotlib` [74] for plotting of images and other figures, `pandas` [75] for tabular assessment of data, `seaborn` [76] for statistical plotting, `optuna` [50] for hyperparameter optimization, are acknowledged by citation as this is preferred by these scientific libraries.

Extensive use of the `SED` (https://github.com/OpenCOMPES/sed) is made, especially allowing easy manipulation of the single-event dataframes with $>1 \times 10^9$ rows, extremely fast binning to multidimensional images and compatibility with HDF5 and tiff formats.

For deep learning, exclusively `PyTorch` [77] has been used. We use the `UNET3D` shown in [71], implemented (modified to accommodate our needs) by Wolny, Cerrone, Vijayan, *et al.* [78]. `PyTorch Dataset` and `DataLoader` classes were extensively used to ease in experiments other than deep learning.

Many of the concepts about Statistical Learning Theory were introduced to the author by the lecture series *Algorithmic Foundations of Data Science* taught by Prof. M Grohe at the RWTH Aachen University. And much of their mathematical understanding on signal and image processing is based on the lecture series *Mathematical Methods of Signal and Image processing* taught by Prof. B Berkels.

Contributions

A preliminary analysis during the early works of this thesis titled *Efficient Data Acquisition in Multi-Dimensional Photoemission Spectroscopy using Denoising*, was presented, in the form a of poster, at the 87. Annual Meeting of DPG and DPG-Frühjahrstagung (DPG Spring Meeting) of the Condensed Matter Section (SKM) in Berlin, as well as NanoMat Science Day 2024 at DESY in Hamburg.

Results from the deep learning denoising were presented at the 7th Round Table on Deep Learning @DESY, in the form of a flash talk titled *Deep-Learning based Image Denoising of MPES data*.

A. Mathematical Background

A.1 Law of Large Numbers

Consider a sequence of iid random variables $X_1, X_2, \ldots, X_n$ defined on the probability space $(\Omega, \mathcal{A}, P)$ (see Appendix A.3), representing the number of detected events in different intervals. Since the random variable are iid, the expected value μ of each X_i is the same. The true mean (or expected value) of these random variables can be defined as $\mu = \mathbb{E}(X_1)$, and the variance as $\sigma^2 = Var(X_1)$.

Formally, by the law of large numbers[1], the sample mean $\bar{X_n} = \frac{1}{n}\sum_{i=1}^{n} X_i$ converges in probability to the expected value μ as $n \to \infty$ [5]:

$$\lim_{n\to\infty} P(|\bar{X_n} - \mu| > \epsilon) = 0 \tag{A.1}$$

The rate of convergence to $\bar{X_n}$ is of practical importance, as it indicates how quickly the sample statistics approximate the true distribution:

$$\left|\bar{X_n} - \mu\right| = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \tag{A.2}$$

indicating that the relative fluctuations decrease with an increased number of observations n_{count}. Increasing the acquisition time T reduces the sample mean fluctuations, since $n \propto T$. While convergence rate described by $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ is a fundamental result, the constant that accompanies this rate is distribution dependent, and can affect the convergence rate. The Big O notation, denoted as $\mathcal{O}(g(n))$, describes an upper bound on the time complexity of an algorithm or the growth rate of a function.

A.2 Measure Space and Measures

Measurable Space.

Let $\Omega = \emptyset$, $P(\Omega)$ the power set of Ω and $\mathcal{A} \subset P(\Omega)$. If the following conditions are met, $\mathcal{A}$ is a called σ-algebra, and the $(\Omega, \mathcal{A})$ pair make up a measurable space.

1. $\Omega \in \mathcal{A}$.
2. If $A \in \mathcal{A}$, then $\Omega \setminus A \in \mathcal{A}$.
3. If $(A_n)_{n\in\mathbb{N}}$ is a sequence of sets where $A_n \in \mathcal{A}$ for all $n \in \mathbb{N}$, then

$$\bigcup_{n\in\mathbb{N}} A_n \in \mathcal{A}.$$

Positive Measure. Let $(\Omega, \mathcal{A})$ be a measurable space as defined in above. A function $\mu : \mathcal{A} \to [0, \infty]$ is called a *positive measure* if it satisfies the following conditions:

[1] Specifically, the weak law of large numbers, which also has a stronger variant proving almost surely (a.s.) convergence.

1. $\mu(\emptyset) = 0$, (the measure of the empty set is zero)
2. μ is *countably additive*: For any countable collection of disjoint sets $(A_i)_{i\in\mathbb{N}} \subset \mathcal{A}$, we have
$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i).$$

If these conditions are met, then μ is called a *measure* on the measurable space $(\Omega, \mathcal{A})$, and the triple $(\Omega, \mathcal{A}, \mu)$ is called a *measure space*.

A.3 Probability

The theory of probability is necessary to quantify stochastic and uncertain quantities. Throughout this text, it can be seen used the in context of inherently stochastic processes, such as quantum effects, and to quantify uncertainty in measurements.

Probability Measure. A *probability measure* $P : \mathcal{A} \to [0, 1]$ satisfies all the properties of a positive measure (see Section A.2), with the additional property known as the normalization condition:

$$P(\Omega) = 1$$

This leads to the measure space for probability (or probability space) being $(\Omega, \mathcal{A}, P)$. The probability of an event $A \in \mathcal{A}$ is $P(A)$. The probability measure is hence a real number between 0 and 1 that defines the likelihood of an event to occur.

A.3.1 Distributions

Poisson distribution. The PMF for Poisson distribution $\text{Poi}(\lambda)$ with $\lambda > 0$ is defined as

$$P(n; \lambda) = \frac{\lambda^n e^{-\lambda}}{n!}, \quad n \in \mathbb{N}_0 \tag{A.3}$$

1. $\lambda = E(X) = Var(X)$
2. Additivity: If $X_1 \sim \text{Poi}(\lambda_1)$ and $X_2 \sim \text{Poi}(\lambda_2)$ are independent Poisson random variables, then the sum $X_1 + X_2$ also follows a Poisson distribution with parameter $\lambda_1 + \lambda_2$
$$X_1 + X_2 \sim \text{Poi}(\lambda_1 + \lambda_2). \tag{A.4}$$
This property is useful when considering counts from multiple independent sources.

For overdispersed count data, the NB is suitable.

Negative Binomial Distribution. The PMF for the NB distribution, denoted $\text{NB}(r, p)$, where:

- $r > 0$ is the number of successes,
- $p \in (0, 1)$ is the probability of success on each trial,

is defined as:

$$P(k; r, p) = \binom{k+r-1}{r-1}(1-p)^k p^r, \quad k \in \mathbb{N}_0 \tag{A.5}$$

where k is the number of failures that occur before the r-th success. For $X \sim \text{NB}(r, p)$, the mean is given by:

$$E(X) = \frac{r(1-p)}{p} \tag{A.6}$$

and the variance is:

$$Var(X) = \frac{r(1-p)}{p^2} \tag{A.7}$$

For relation between Poisson, NB and Gamma distributions, see [79]. It can be shown that NB converges to Poisson as $r \to \infty$. The NB distribution can also be considered as a mixture of Poisson distributions with a Gamma prior on the rate parameter λ.

A.4 Statistical Inference

A.4.1 Confidence Intervals

Confidence Intervals are a way to quantify the uncertainty in an estimate. They provide a range of plausible values for a parameter, rather than a single point estimate. The confidence level indicates the probability that the interval contains the true parameter value. For example, a 95% confidence interval means that if we were to repeat the experiment many times, 95% of the intervals would contain the true parameter value.

Parametric methods for computing confidence intervals rely on knowing the underlying distribution of the data, such as the normal distribution. However, in many cases, the true distribution is unknown or does not follow a standard form. In such cases, non-parametric methods such as bootstrapping can be used.

Bootstrapping involves resampling the data with replacement to create multiple bootstrap samples, from which the sample statistic is computed. The distribution of the sample statistic across the bootstrap samples provides an estimate of the sampling distribution of the statistic, and becomes more informative when using larger sample sizes.

For the case of computing image metrics, the confidence intervals around the mean sample statistic can be used to quantify the uncertainty in the metric estimate. This is useful to see the variability in the denoising performance, or in metric characteristics.

A.5 Metrics

MSE is a common measure of the average squared differences between predicted values and actual values. It quantifies the error in a model's predictions, where lower values indicate better performance. It is sensitive to outliers since squaring the differences amplifies larger errors.

Mean squared error (MSE).

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{A.8}$$

- x_i: pixel value of the noisy image
- y_i: pixel value of the target image

- N: total number of pixels

PSNR is a measure of the peak error between two images, often used to evaluate image compression quality. It expresses the ratio between the maximum possible power of a signal and the power of corrupting noise. Higher PSNR values indicate better quality.

Peak signal-to-noise ratio (PSNR).

$$\text{PSNR} = 10 \log_{10} \left(\frac{L^2}{\text{MSE}} \right) \tag{A.9}$$

- L: maximum pixel value of the image
- MSE: mean squared error

SSIM is a perceptual metric that quantifies the visual impact of three characteristics of an image: luminance, contrast, and structure. It evaluates the similarity between two images based on their structural information, making it more aligned with human visual perception than MSE or PSNR. Values range from -1 to 1, with 1 indicating perfect similarity.

Structural similarity index measure (SSIM).

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{A.10}$$

- μ_x, μ_y: mean of x and y
- σ_x, σ_y: standard deviation of x and y
- σ_{xy}: covariance of x and y
- $C_1 = (k_1 L)^2, C_2 = (k_2 L)^2$: constants to stabilize the division

MS-SSIM extends SSIM by evaluating image similarity across multiple scales and resolutions. By considering different image scales, it captures more information about structural variations and perceptual quality, making it particularly useful for assessing image quality in applications like compression. Similar to SSIM, higher values indicate greater similarity.

Multi-scale structural similarity index measure (MS-SSIM).

$$\text{MSSSIM}(x, y) = \prod_{j=1}^{J} \text{SSIM}(x_j, y_j)^{\alpha_j} \tag{A.11}$$

- J: number of scales
- x_j, y_j: images at scale j
- α_j: weight for scale j, summing to 1

The metric implementations used in this thesis are either `PIQA` library https://github.com/francois-rozet/piqa, or `skimage` library https://scikit-image.org/.

B. Supplementary Material

B.1 Transforming Raw Data to structured format: Extract, Load, Transform

Raw data from the experiment is stored in HDF5 files. This includes many *beamline* diagnostic information such as beam arrival monitor (BAM), GMD, the delay stage readings, the monochromator energy, sample specific information such as extractor voltage, and the electron counting in the 3 detector dimensions corresponding to the DLD spatial x and y axes and the temporal t_{tof}. This information is resolved at each bunch of electrons coming from the accelerator called a *train*, which are further microbunched into *pulses*.

The Open Community of Multidimensional Photoemission Spectroscopy (OpenCOMPES) was established to develop tools and infrastructure to make analysis easier. To this end, a modular Python library called `SED` was created that provides the entire pipeline from easy data loading to common calibration and corrections, multidimensional binning to create images, and saving the images to standard formats, with proper care of data provenance.

The data pipeline follows an extract, load, transform (ELT) process, wherein raw data is extracted from HDF5 files, transformed into a structured format suitable for analysis, and subsequently stored in intermediate buffer files for further downstream processes such as analysis and visualization.

The following sections provide a detailed exposition of each stage of this ELT process, also shown in Appendix B.1. The first stage, extraction, begins with the loading of raw data from the HDF5 files. These files encapsulate experimental results across multiple channels, including electron-resolved and time-resolved data. The hierarchical structure of the HDF5 files allows the organization of data into groups, where each group contains an index and its corresponding dataset, collectively referred to as a "channel" The paths to these HDF5 files, along with relevant configuration parameters, are provided to the pipeline to dictate the steps of the subsequent transformation process.

Pipeline Overview

To optimize performance and facilitate data management, the pipeline generates buffer files for each type of data (electron and time-resolved). This task is handled by the `BufferFilePaths` class, which initializes file paths and manages the creation of buffer files in the efficient `Parquet` format. By checking the presence of pre-existing buffer files, the class determines whether to reuse these files or regenerate them, based on the `force_recreate` flag.

Each HDF5 file results in the generation of two primary buffer files: one containing electron-resolved data and another containing pulse/train-resolved data. These files are essential for organizing the data at the pulse level and maintaining resolution at the electron level. The data for each train contains roughly 500 pulses, and although some data is resolved at the pulse level, each index often holds an array, necessitating the use of the `pandas` MultiIndex functionality to maintain the data's hierarchical structure. Detector measurements, which are electron-resolved along the X, Y, and temporal axes, are typically represented as three-dimensional arrays and add further complexity to the indexing process.

A critical transformation step involves correcting for offsets in pulse IDs to ensure accurate synchronization of data across different channels. Any data associated with pulse IDs below zero

is removed, as it is considered invalid. Similarly, `NaN` pulses are dropped to avoid introducing inconsistencies in downstream analyses. While it is possible that pulses exceeding 500 may also be invalid, these are not filtered during this stage, as that determination is deferred to the final analysis. Due to machine fluctuations, pulses may become unsorted, and hence the pulses are sorted within each train to maintain temporal order.

Once this cleaning process is completed, electron-resolved channels are combined using an outer join with pulse and train-resolved channels, forming a comprehensive dataframe that contains all relevant information for further analysis. This merged dataset is separated into two primary dataframes: the electron-resolved dataframe and the pulse-resolved dataframe.

The electron dataframe contains only rows where electron events have been detected, with any missing data in non-electron channels forward-filled to ensure completeness. This dataframe serves as the main source of electron-specific data for further analyses. However, given that not all pulses or trains may produce electron events, a separate pulse-resolved dataframe is generated to capture all available train and pulse data, independent of electron detections. This pulse-resolved dataframe is essential for normalization steps involving time-resolved channels, such as those related to the delay axis.

The pipeline also includes validation steps for auxiliary channels, particularly those containing multidimensional data (e.g. 4D arrays), ensuring that all required channels exist within the files before proceeding with further transformations.

After this initial transformation and extraction, the buffer files are saved in `Parquet` format, chosen for its efficiency in storage and speed of access during future computations. The final stage involves the loading of all these buffer files into a unified dataframe using `Dask`, a distributed computing library that enables scalable processing of large datasets. At this stage, forward-filling is again applied to non-electron channels, ensuring that missing values between files are handled consistently. However, care must be taken when forward-filling across different runs, as this could introduce inter-run inconsistencies.

Finally, the schema of the buffer files is cross-validated against the predefined list of channels to ensure consistency and completeness prior to loading the data into `Dask`.

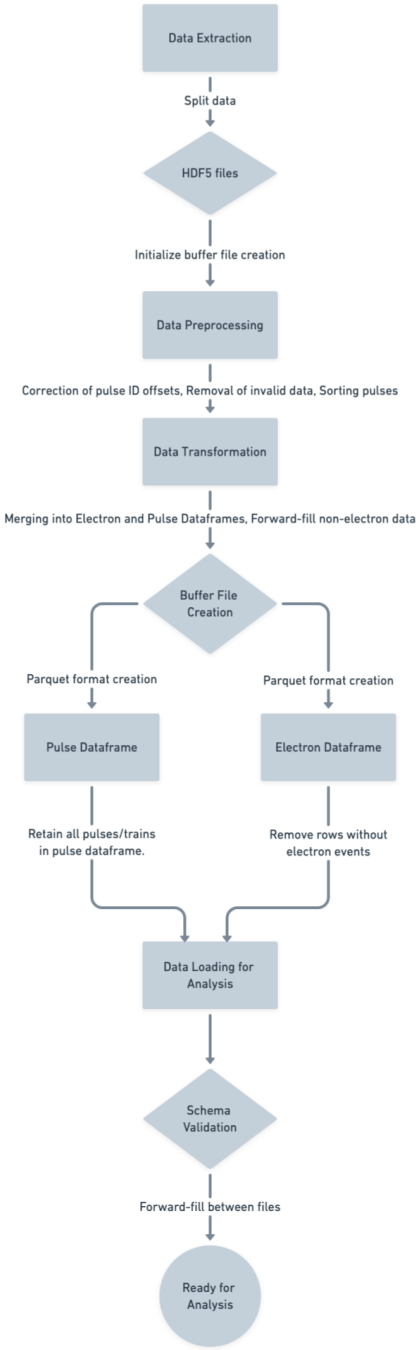
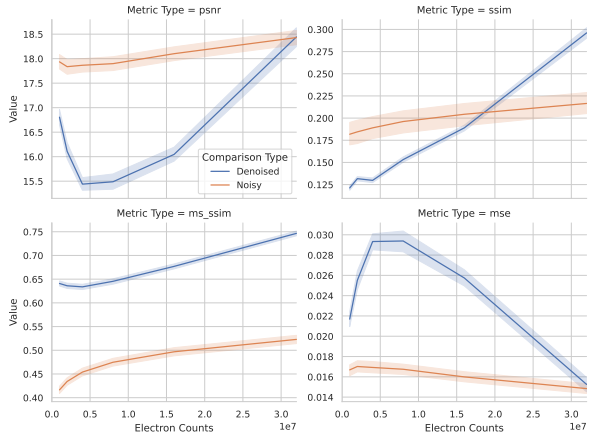


Figure B.1: Complete ELT pipeline for the data from HEXTOF at FLASH.

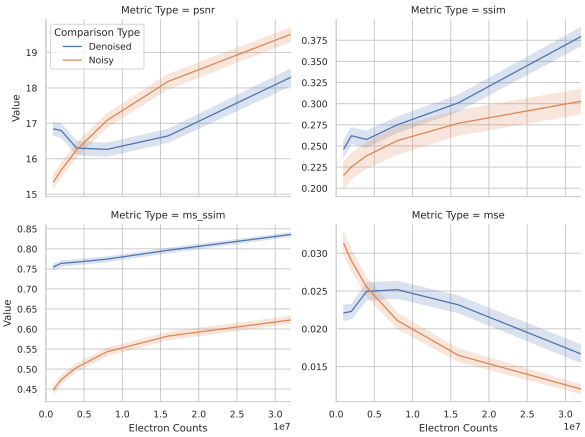
B.2 Experiment: Metric Comparison

The objective of this experiment is to evaluate the effectiveness of various image quality metrics (MSE, PSNR, SSIM, and MS-SSIM) when the provided reference is noisy. Specifically, denoised images from a trained neural network, that has shown clear improvements in perceptual quality[1], are compared against the high-count but noisy reference. Examples of such images at different counts can be seen in Figure 6.9, which shows how the noisy images are devoid of any features at low counts, while the denoised images show clear features similar to the target. For this

[1]This is assessed by an expert, comparing with the high-count target.



(a) Comparison of input and target images window averaged with $w = 1$. All metrics improve as counts increase, though MSE and PSNR show better results for noisy inputs, while SSIM and MS-SSIM favor denoised inputs. MS-SSIM shows the greatest improvement, indicating it is less sensitive to noise and focused more on perpetual quality.



(b) Comparison of input and target images window averaged with $w = 1$ and $w = 10$, respectively. All metrics improve much faster as counts increase. With a window-averaged target, MSE and PSNR start to favor the noisy input. SSIM shows improvement that the perpetually better images are preferred at all counts. MS-SSIM exhibits a further separation between noisy and denoised inputs.

Figure B.2: Evaluation of metrics (MSE, PSNR, SSIM, MS-SSIM) for noisy and denoised inputs, comparing against noisy input and (less noisy) target slices. The lines show the mean value at each n_{count}, and the error bands shown in the plots represent a 95% confidence interval around that mean.

analysis, it is not relevant if the high quality reconstructions are due to overfitting or not, but rather that they produce perceptually better quality images (sometimes even compared to the target). So the hope is that the metrics acknowledge this improvement.

The Gr/Ir(111) dataset is used, with electron counts ranging from $1 \times 10^6 - 3.2 \times 10^7$. For each total count, a set of 2D noisy images (referred to as noisy slices) is extracted from the dataset. Corresponding high-count 2D slices (referred to as target slices) are used as reference images for comparison (from the dataset with 1.86×10^8 counts). Likewise, reconstructed images from the neural network are also compared against the target slices.

The metric evaluation procedure involves a data loader, implemented using the `PyTorch Dataset` class, which iterates through the different datasets and extracts noisy and target 2D image slices (window-averaged or single slice). For each pair of noisy and target slices, the metrics are computed. The comparison results are then grouped by acquisition count and type (noisy vs. denoised).

This metric comparison is performed with 1638 images sliced along each dimension of the dataset. The results are aggregated and a 95% confidence interval is calculated for each metric to assess the reliability of the comparison.

Comparing the noisy images with target, all metrics trivially show an improvement with increased counts (less noise) (see Figure B.2a). However, MSE, PSNR and SSIM show deteriorated metrics when comparing against perpetually high quality images. This can be attributed to the presence of noise in the reference image. For higher counts, SSIM manages to show improved results, whereas MSE and PSNR report consistently worse results compared to the noisy images. MS-SSIM consistently reports better results with the higher quality images, making it the most reliable metric for evaluation.

Additionally, we investigate the effect of window-averaged reference images on the metrics. Results in Figure B.2b indicate that window-averaged image as reference generally makes the metric prefer the noisy realization, except in the case of SSIM and MS-SSIM, in the latter of

which the gap between the metrics calculated with noisy and high quality images split up further. Consequently, we adopt the window-averaged images as reference for the denoising evaluation.

B.3 Deep Learning Infrastructure

Initially, we briefly explored using a UNet2D model for denoising. However, due to the low count rates in the training data, slice-summing was required to achieve meaningful results. To better capture the structural features of the data, we opted for a UNet3D model, which processes volumetric information directly. Although it is theoretically possible to train networks with even higher-dimensional convolutions, this becomes computationally prohibitive due to the exponential increase in the number of parameters. Even with 3D convolutions, the parameter count is significantly larger compared to 2D convolutions, making efficient resource utilization critical.

The neural network, with trainable parameters 4 119 227, was trained on an NVIDIA A100 GPU with 80 GB of memory, using the Maxwell Cluster at DESY. With a patch size of $60 \times 240 \times 240$, a batch size of 16 can be fit in memory. While this works with UNET3D, other CNNs with 3D convolutions would require a much larger memory footprint.

To optimize training, the learning rate was reduced after 10 validation iterations if no improvement in validation loss was observed. While this approach likely facilitated convergence in SSIM values, it may have led to the network settling in suboptimal local minima. This is reflected in the lack of improvement in SSIM values for some cases, particularly when compared to the best-performing model. This suggests that the learning rate drop, while effective in some scenarios, could hinder progress in others.

B.4 Supplementary Figures

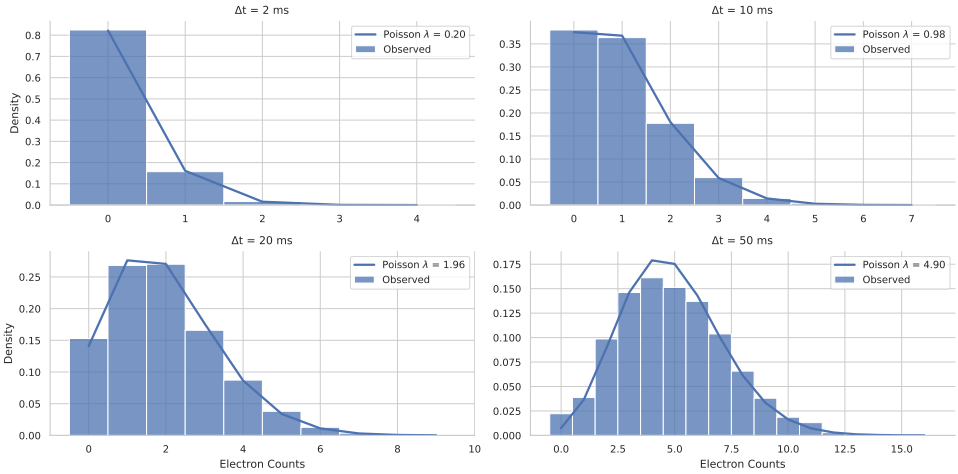


Figure B.3: Distribution of photoelectron counts at time intervals $\Delta t = 2\,\text{ms}$, 10 ms, 20 ms and 50 ms for a different volumetric subset of the full WSe_2 dataset. Poisson statistics are observed at smaller time intervals, but as the time window increases ($\Delta t = 50\,\text{ms}$), the data starts to deviate from the Poisson distribution, as spatial correlations become apparent.

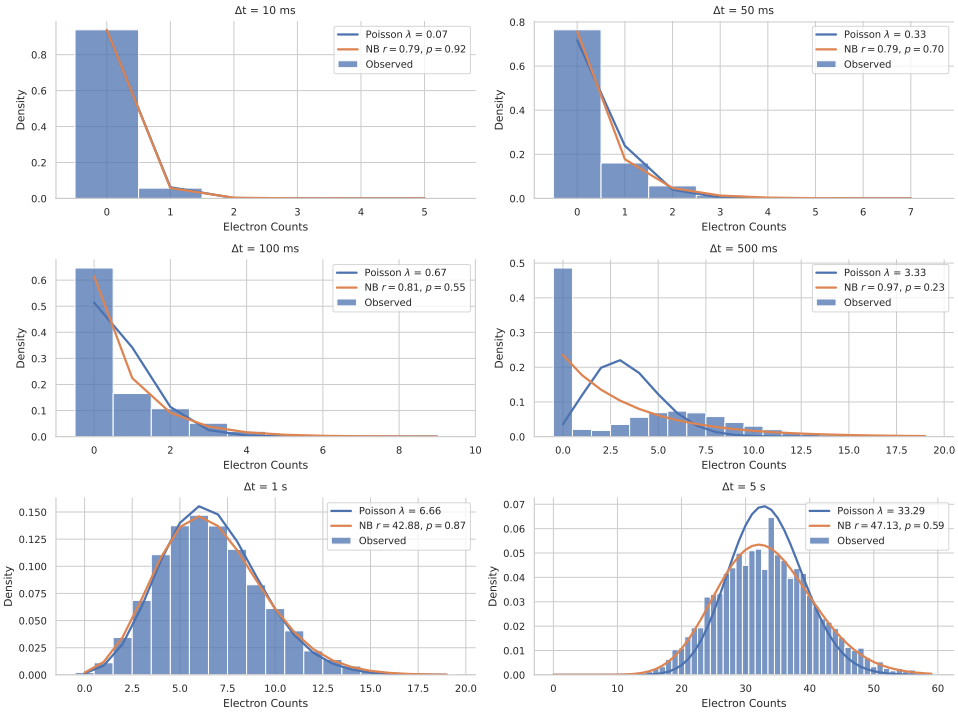


Figure B.4: Distribution of photoelectron counts at time intervals $\Delta t = 10\,\text{ms}$, 50 ms, 100 ms, 500 ms, 1000 ms and 5000 ms for a selected volumetric subset of the full Gr/Ir(111) dataset. Poisson statistics provide a good fit at $\Delta t \leq 100\,\text{ms}$. However, for intervals $\Delta t = 0.5\,\text{s}$, 1 s and 5 s the distribution exhibits over dispersion with a right-skewed tail, characteristic of the NB distribution. The total counts in this selected region are $n_{\text{count}} = 1 \times 10^6$ with total observation time $T \approx 4\,\text{h}$ (See red region in Figure 5.3). The smaller Δt values may represent the limiting case of the NB distribution approaching Poisson statistics.

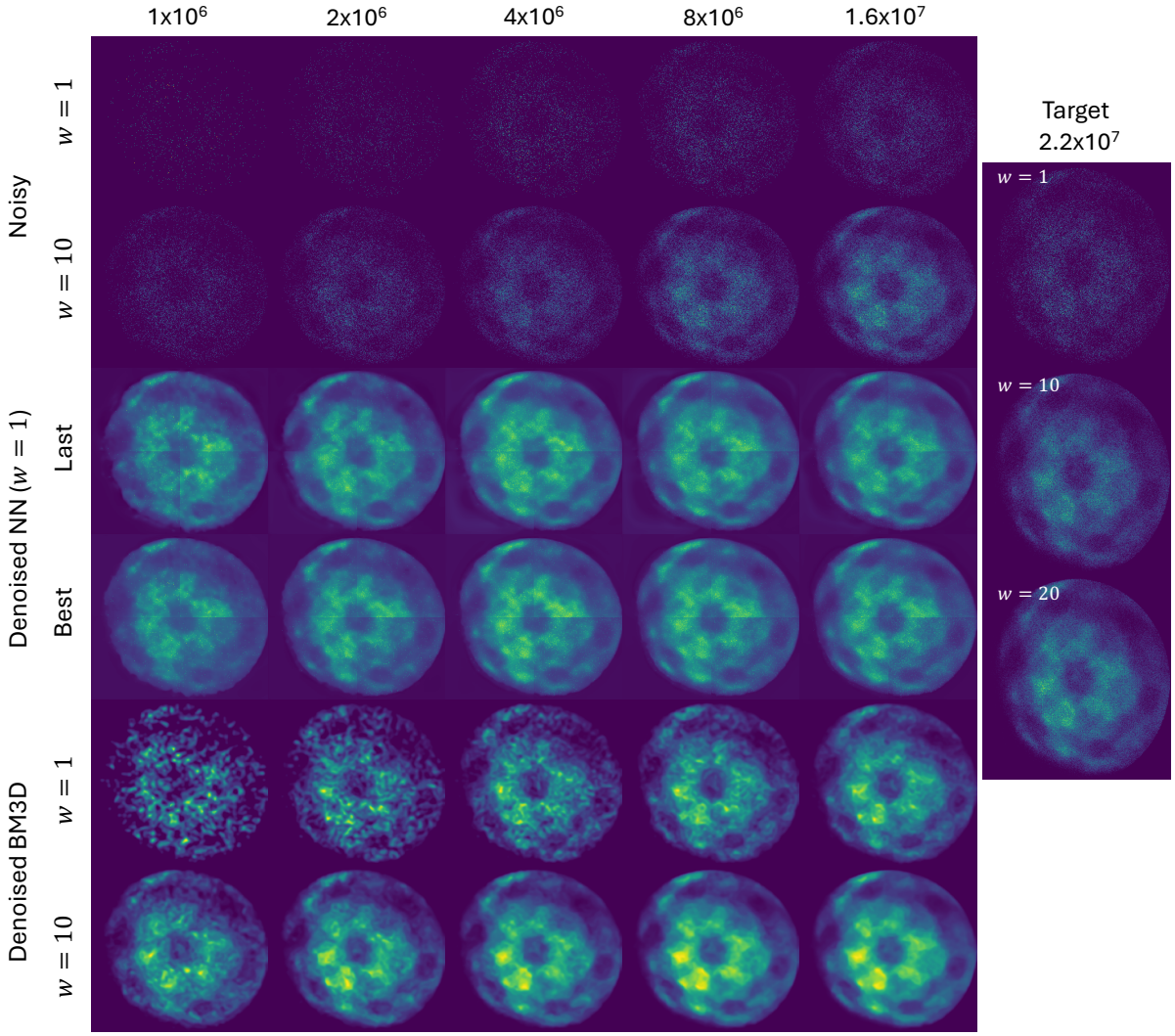


Figure B.5: Comparison of k_x–k_y slices for various n_{count}. The target image is shown for a single slice and for slice-summed images with w values of 10 and 20, demonstrating the effect of summing slices for visual quality improvement. The noisy image is also depicted as a slice-summed result, emphasizing that slice summing acts as a basic form of noise reduction. Denoised results are shown for the best and last trained neural network models ($w = 1$), alongside BM3D–Anscombe denoising applied with the optimal parameter σ_{o} for $w = 10$ and the same parameter used for $w = 1$. The results illustrate that the neural network outperforms BM3D–Anscombe denoising at lower counts (1×10^6). At higher counts and when slices are summed, BM3D's performance improves relative to the neural network.

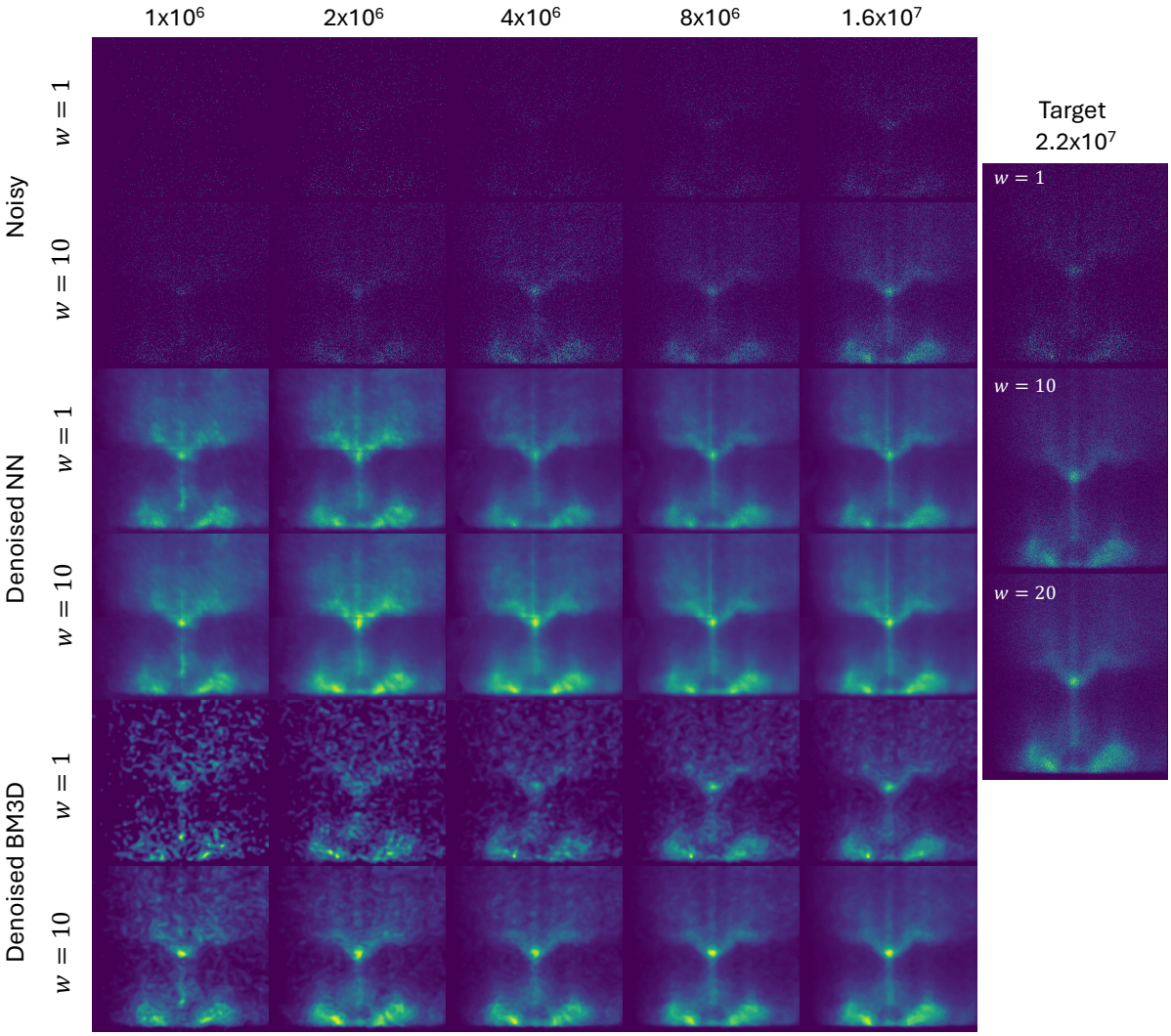


Figure B.6: Denoised results are shown for the best and last trained neural network models, and for BM3D-Anscombe for single slice cuts and slice-summed images. The results indicate similar performance when slice-summed between neural network and BM3D.

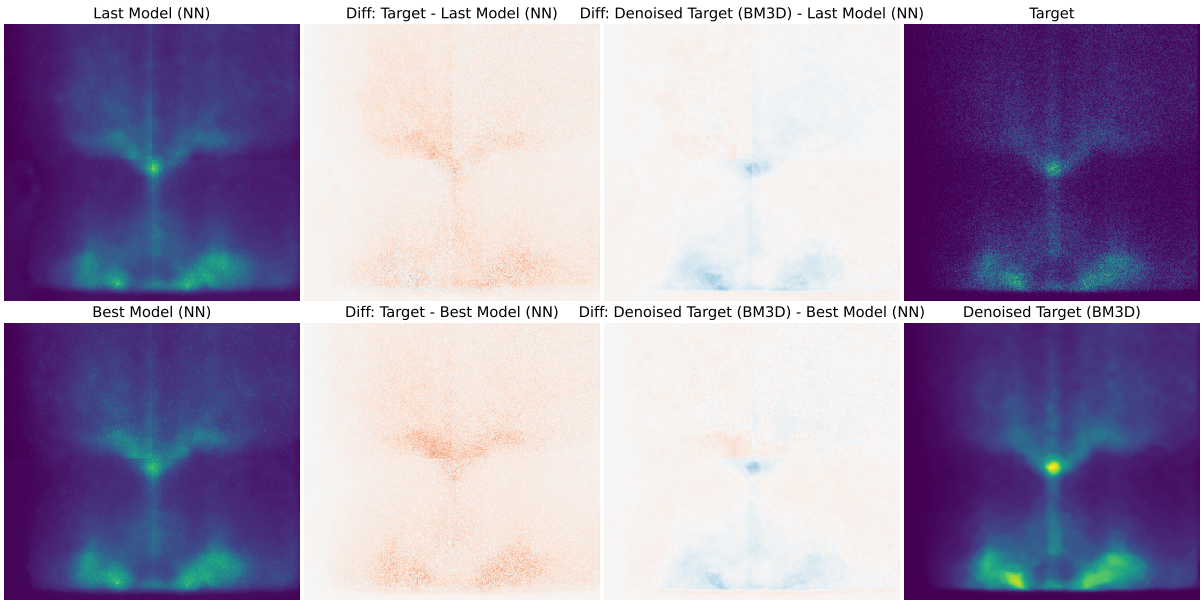


Figure B.7: Difference images comparing the best and last models for n_{count} 8×10^6. The differences are computed relative to two references: the slice-summed target image with w of 10, and its BM3D-denoised version. The red regions indicate where the prediction is brighter than the noisy target, while blue regions indicate where the target is brighter. The target itself is noisy, so when subtracting the denoised outputs from the target, mostly red regions are visible. However, when using the BM3D-denoised target as a reference (assumed to approximate the correct image), the differences are mostly white (close to 0), with some blue regions where predicted features are less intense compared to the reference. Here, the difference plots highlight that the last model preserves features more effectively and aligns better with the BM3D-denoised target compared to the best model.