DEEP LEARNING DENOISING FROM MULTIVIEW VIDEOS OF FREELY MOVING MOUSE

HOW DENOISING AFFECTS THE 3D ANIMAL POSE RECONSTRUCTION PIPELINE

THOMAS WYNDHAM BUSH

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2114162

COMMITTEE

dr. Sharon Ong
dr. Seyed Mostafa Kia

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

June 23rd, 2025

WORD COUNT

7281

ACKNOWLEDGMENTS

I deeply thank my supervisor dr. Sharon Ong, for the precious advices and support throughout the writing of this thesis. I also wish to thank all the staff at the Iurilli Lab for their support and insights.

DEEP LEARNING DENOISING FROM MULTIVIEW VIDEOS OF FREELY MOVING MOUSE

HOW DENOISING AFFECTS THE 3D ANIMAL POSE RECONSTRUCTION PIPELINE

THOMAS WYNDHAM BUSH

Abstract

Understanding how behavior is represented in the brain is a key challenge in neuroscience (Pereira et al., 2020). A robust and quantitative representation of behavior is essential for establishing meaningful links between behavioral outputs and neural activity. In recent years, a variety of computer vision–based tracking methods have been developed to quantify animal behavior, including tools like DeepLabCut (Alexander Mathis et al., 2018), ANIPOSE (Pierre Karashchuk et al., 2020), SLEAP (Pereira et al., 2022) and Lightning Pose (Biderman et al., 2024). However, these methods predominantly focus on the pose estimation task itself, with comparatively little emphasis on preprocessing the input data—for instance, improving video quality through denoising or optimizing the selection of training frames for supervised labeling.

This thesis addresses that gap by developing a tracking-model-agnostic preprocessing pipeline that can enhance the quality of input data for behavior quantification by testing preprocessing techniques on an experimental dataset of videos of freely moving mice in an arena. Specifically, I explore two key components: video denoising and video summarization (i.e., frame selection). For denoising, I compare three deep learning approaches: a standard autoencoder, a residual connection autoencoder (CBDNet), and an attention-based denoising model (PRIDNet). For summarization, I extract features using a pretrained ResNet and evaluate two strategies— further point sampling (FPS) and K-means clustering—for selecting representative keyframes.

The results show that the standard autoencoder (MSE: 0.004, SSIM: 0.041) and the residual model CBDNet (MSE: 0.002, SSIM: 0.043) outperform the attention-based PRIDNet (MSE: 0.11, SSIM: 0.149) on unseen video data, demonstrating better generalization and denoising performance. On the summarization task, K-means clustering produces clearer clusters compared to FPS, enabling efficient and meaningful keyframe selection.

Together, these results highlight the importance of accurately preprocessing the data before downstream analysis, as both effective frame selection and robust denoising significantly impact the quality and reliability of behavioral interpretations.

1 DATA SOURCE, ETHICS, CODE, AND TECHNOLOGY STATEMENT

Data Source: The data used in this thesis was collected at the Iurilli Lab (Italian Institute of Technology, IIT), which is the sole owner of the data. It was shared with me during the course of my internship at the lab, with full consent for use in this academic work.

Figures: All figures included in this thesis were created by me. When figures were inspired by or based on other works, appropriate credit and citations have been included in the respective captions.

Code: The code used for this thesis was written by me and is publicly available on the accompanying GitHub repository .[1]. Syntax and style checks have been applied to ensure code quality.

Technology Used: Zotero was used for reference management and citation integration throughout the writing process. To improve the clarity and academic tone of the text, I made occasional use of the Academic Phrasebank and OpenAI's ChatGPT to rephrase or refine certain sections.

2 INTRODUCTION

The overall goal of this thesis is to explore to what extent different Deep Learning (DL) denoising and frame clustering techniques can be applied to videos of freely moving mice used within a pipeline to extract 3D postural data. This goal is motivated by the fact that animal behavior is challenging to study due to its innate variability (Datta et al., 2019), and even state-of-the-art pipelines struggle with tracking freely moving animals (Mathis & Mathis, 2020) leading to add more constraints on the animal movements during the experiments, thus, limiting the ecological validity of the study.

2.1 *Societal Relevance*

Historically, neuroscience has tended to focus on the study of low-level properties of the brain, often without integrating a pluralistic approach that includes both brain and behavior (Krakauer et al., 2017).

Several factors have contributed to this tendency. First, studying animal behavior has traditionally relied on human annotations, which are time-

[1] https://github.com/Thomasbush9/thesis-project

consuming and difficult to generalize (Lukas von Ziegler et al., 2021). Second, behavioral paradigms have often been constrained to accommodate the requirements of neural measurement tools, such as fMRI or MEG (Krakauer et al., 2017). For example, researchers frequently need to limit an animal's movement to ensure precise data collection, yet these restrictions can distort natural behaviors, reducing the generalizability of findings (Kennedy, 2022).

Nevertheless, behavior is increasingly recognized as the brain's most essential output (Mathis & Mathis, 2020), and being able to study with the same precision with which researchers are able to study the brain is crucial to improve the current understanding of the brain (Krakauer et al., 2017).

In light of this, researchers have argued that capturing freely expressed behavior is essential for interpreting neural recordings accurately (Datta et al., 2019). While advances in tracking and computational methods have improved data collection in behavioral studies, significant challenges remain in obtaining high-quality, high-resolution behavioral data from freely moving animals (André E. X. Brown et al., 2018; Datta et al., 2019).

Developing accurate 3D pose reconstructions of freely moving mice can address many of these challenges by providing detailed movement data while preserving naturalistic behavior. High-quality 3D tracking allows researchers to reduce artificial constraints, leading to more ecologically valid experiments and insights into fine-grained motor patterns that are otherwise difficult to quantify (Alexander Mathis et al., 2018; Biderman et al., 2024; Caleb Weinreb et al., 2023; Pereira et al., 2022). Furthermore, understanding animal behavior at this level has far-reaching implications beyond zoology, offering critical insights into neural mechanisms underlying motor control, decision-making, and neuro-psychiatric conditions, as many fundamental principles of movement and cognition are conserved across species (André E. X. Brown et al., 2018).

2.2 *Scientific Motivation*

Modern 3D pose estimation pipelines for freely behaving animals, such as those used in neuroscience and ethology, rely heavily on multi-camera recordings and 2D keypoint tracking to reconstruct the subject's posture in three dimensions (Pereira et al., 2022; Pierre Karashchuk et al., 2020). Despite significant advancements, these pipelines face persistent limitations that compromise accuracy and scalability.

A central challenge lies in the quality and consistency of 2D tracking, which forms the foundation for reliable 3D triangulation. In unconstrained settings—where animals move freely and perform rapid, complex behaviors—2D pose estimates are frequently degraded by motion blur, occlusions,

and variable lighting. These visual artifacts reduce the reliability of keypoint detectors and introduce noise into the resulting 3D reconstructions. As a result, frames with poor visual quality often contain jitter, missing data, or outlier keypoints, making accurate multi-view triangulation difficult or even infeasible.

While some efforts have addressed these issues by incorporating multi-view consistency constraints during training (Biderman et al., 2024) or applying post-hoc spatio-temporal smoothing (Pierre Karashchuk et al., 2020), these methods do not directly improve the input video quality or the representativeness of training data. This motivates the need for preprocessing techniques that improve the quality of the raw input frames before they are passed to tracking algorithms.

In this work, I address this gap by exploring two complementary strategies: (1) deep learning-based video denoising, which aims to improve the visual quality of each frame and thereby the accuracy of keypoint detection, and (2) keyframe selection via feature-space sampling, which selects representative and diverse frames to improve the robustness of supervised tracking models. Together, these methods directly target the quality and informativeness of the video input, offering a general preprocessing pipeline that can be integrated into any 3D tracking system.

3 RESEARCH QUESTIONS

3.1 *To What Extent Feature Extraction and Clustering Methods Can Help Extract Keyframes from Naturalistic Videos?*

Keyframe selection aims to reduce redundancy and prioritize high-quality, behaviorally informative frames. Rather than processing every frame in a long video sequence, keyframe selection methods identify and retain only those frames that capture distinct postures or meaningful transitions—such as sudden limb movements or interactions between animals. By filtering out static or ambiguous frames, this strategy reduces computational cost while improving the diversity and informativeness of the data used for 3D reconstruction. Crucially, keyframe selection also allows human annotators or downstream algorithms to focus on salient behaviors, which can enhance both tracking robustness and interpretability.

There are two popular techniques for key frame selection, which are K-means and Further Point Selection (FPS) . One way to compare them is to compare PCA on the results to indicate if there are any clusters. Therefore, a sub-question is *to what extent do K-means and FPS generate clusters of keyframes as shown in PCA plots.*

3.2 *To What Extent Deep learning Denoising Techniques Can Improve Deep Learning Pipelines to Reconstruct 3D Postural Data from Mice Videos?*

Denoising techniques improve the signal quality of individual frames or pose sequences (Elad et al., 2023; Tian et al., 2020). These methods aim to remove noise caused by motion blur, poor lighting, or camera artifacts before keypoint detection and triangulation. By applying image-level denoising (e.g., using deep learning models trained to remove Poisson-Gaussian noise) or pose-level smoothing (e.g., using learned priors or trajectory filters), pipelines can produce more stable and accurate 2D predictions. This, in turn, leads to more precise 3D reconstructions. In recent work, denoising has also been shown to mitigate the impact of missing or corrupted keypoints, making pose estimation more robust under real-world conditions.

Together, these two strategies address complementary weaknesses in current pipelines: keyframe selection reduces the volume of low-information data, while denoising improves the quality of the retained frames. Integrating these preprocessing steps has the potential to make 3D behavioral analysis pipelines both more accurate and more computationally efficient, especially in large-scale or long-term experiments. However, the joint impact of these techniques on 3D pose quality has not been systematically evaluated—an open question that this thesis seeks to address.

3.3 *Error Analysis Over Keyframe Selection*

To address the research question concerning the quality and representativeness of keyframes extracted from behavioral videos, I conducted a targeted error analysis (Section 6.1.1) on the output of the two selection algorithms: K-means clustering and Farthest Point Sampling (FPS). The goal of this analysis was to understand whether the selected frames capture sufficient structural diversity and avoid redundancy in the latent embedding space.

By comparing pairs of keyframes that are either very close or far apart in PCA space, I evaluated the ability of each algorithm to reflect meaningful visual variation. This qualitative analysis served to complement quantitative metrics by revealing the spatial distribution and coverage of selected keyframes in the low-dimensional feature space.

4 RELATED WORK

4.1 *Animal Tracking and Pose Reconstruction*

Ethology, the biological study of behavior and social organization, has relied in the past on manual annotations of complex behavioral sequences. These annotations have struggled to capture the complexity of behavior in a rigorous manner, thus many findings were not generalizable.

In recent years, Deep Learning (DL) methods have enabled the automatic identification of animal behavior, improving annotation reliability while reducing manual effort. The adoption of DL for behavioral quantification has allowed researchers to analyze behavior at finer temporal resolutions, revealing its fundamental role in addressing key questions in neuroscience (Krakauer et al., 2017) As a result, the emerging field of computational neuroethology has focused on linking behavioral data with neural activity to uncover the neural mechanisms underlying behavior (Kennedy, 2022). Achieving this goal requires high-quality behavioral data that accurately represent the behaviors under investigation.

Classical approaches for tracking animal movements treat the animal as a single point, describing its behavior solely based on spatial displacement. These methods rely on algorithms that model the background to segment the animal, but they are constrained by background dependencies, limiting their application in naturalistic experiments (Pereira et al., 2020).

Modern methods, known as animal pose estimation, aim to track the movements of limbs and appendages to capture and distinguish more complex behaviors (Mathis & Mathis, 2020). These approaches leverage deep learning models, such as Convolutional Neural Networks (CNNs) and graph-based architectures, to learn keypoint representations directly from video data. By training on large annotated datasets, these models can generalize across different environments, improving tracking accuracy even in naturalistic settings (Pereira et al., 2022).

While for some animals simple 2D pose estimation is a good representation of their behavior, for other animals (e.g., mice) it can occur that a single point of view fails to capture their poses when occluded. This can be avoided by implementing a 3D approach for animal pose estimation (Pereira et al., 2020).

The most used approach for 3D pose estimation consists of using multiple 2D views to reconstruct a 3D estimate through triangulation (Mathis & Mathis, 2020). The triangulation process can be subject to noise from the 2D predictions, thus researchers commonly perform a refinement procedure over the triangulated predictions to eliminate unrealistic data

points by applying spatio-temporal constraints (Pierre Karashchuk et al., 2020).

However, current pipelines largely assume clean input data and rarely incorporate active preprocessing, making them vulnerable to errors from video noise or redundant training data.

4.2 *Keyframe Selection*

Keyframe selection aims to identify a subset of representative frames from a video that best summarize its visual content. This reduces redundancy, enhances interpretability, and enables more efficient downstream processing—especially in high-resolution or long-duration videos, such as those used in 3D pose estimation and behavioral neuroscience.

Traditional approaches for keyframe selection rely on clustering-based techniques in features space, where keyframes are selected as cluster centroids. Methods such as K-means, hierachical clustering (Ran et al., 2023) and density based clustering like DBSCAN (Ester et al., n.d.) have shown to be appropriate for this task when combined with dimensionality reduction techniques such as PCA to preserve the the underlying structure of the data (Apostolidis et al., 2021). In the context of animal behavior analysis, keyframe selection is often used to select meaningful and representative frames to annotate (Pereira et al., 2022) before training a model for postural tracking.

However, these methods have important limitations. Clustering approaches are sensitive to hyperparameter choices (e.g., number of clusters) and often assume a fixed notion of similarity that may not align with task-relevant behavioral variation. Additionally, many approaches overlook temporal coherence or motion-based dynamics, which are crucial in animal behavior where fine-grained postural shifts occur across short timescales.

Recent work in the broader computer vision field has explored more advanced keyframe selection methods using deep features, reinforcement learning, or attention mechanism to learn task-specific frame importance scores (Liang et al., 2024; Tan et al., 2024; Tang et al., 2025). Yet, these approaches are computationally intensive and often require supervision or ground truth summaries, which are rarely available in behavioral datasets.

To date, relatively few studies have evaluated keyframe selection specifically for improving model training in multi-view pose estimation pipelines. This presents an opportunity to systematically benchmark lightweight, unsupervised keyframe selection methods—such as feature-based clustering and farthest point sampling—in the context of freely behaving animal videos.

4.3 *Image Denoising*

Image denoising plays a key role in improving the quality of input frames used for 2D keypoint detection, especially in behavior videos affected by motion blur or sensor noise. Denoising techniques can be broadly categorized into: filter-based methods, such as gaussian smoothing or bilateral filters, which used predefined kernels to remove noise (Tian et al., 2020). Such methods are computationally expansive as they require to estimate the noise before being fitted to the data.

Another approach to denoising is represented by learning-based methods which leverage neural networks to reconstruct the noise-free image (Elad et al., 2023). In this work, I focused on deep learning-based denoising models to enhance video quality before pose estimation. I considered and compared three representative models:

(1) *denoising autoencoders:* Introduced by (Vincent et al., 2008), they are trained to reconstruct clean images from artificially corrupted inputs. The architecture typically consists of an encoder that maps the noisy input to a latent representation and a decoder that reconstructs the image. DAEs can learn to suppress noise patterns and preserve semantic content even with limited supervision. While simple and effective, DAEs may struggle with highly structured or non-additive noise without architectural or loss-function modifications.

(2) *CBDNet* (Guo et al., 2019), is designed for real-world blind denoising, where both the type and level of noise are unknown. It employs a two-branch architecture: a noise estimation subnetwork that learns a noise map, and a denoising subnetwork that uses both the noisy image and noise map to recover the clean image. CBDNet is trained with synthetic Poisson-Gaussian noise combined with a camera image processing pipeline to approximate real-world degradations. It also includes an asymmetric loss function to balance over- and under-estimation of noise. This architecture makes CBDNet well-suited for noisy behavior videos where the noise characteristics vary across cameras and sessions.

(3) *PRIDNet*, (Zhao et al., 2019), is a pyramidal network that focuses on capturing multi-scale features through a residual learning framework. It incorporates attention mechanisms to refine features at different levels, enabling it to handle both fine-grained textures and global structure. The pyramid design allows it to adaptively model noise at different resolutions, making it effective for denoising images where noise manifests at multiple spatial scales — a common trait in compressed or high-frame-rate behavior videos. In this work, we evaluate PRIDNet's capacity to preserve fine structural details (e.g., limb contours) critical for accurate keypoint detection.

Interestingly, many of the existing approaches for animal tracking—such as SLEAP (Pereira et al., 2022), Anipose (Pierre Karashchuk et al., 2020), and Lightning Pose (Biderman et al., 2024)—rely on minimal image or video preprocessing prior to keypoint detection and 3D reconstruction. This thesis seeks to extend that pipeline by exploring the role of deep learning-based denoising methods as a preprocessing step, with the goal of improving the quality and reliability of 3D behavioral data reconstruction.

4.4 *Research Gap and Novel Contribution*

While prior work has significantly advanced multi-view animal tracking and behavioral pose estimation, most pipelines remain limited by their reliance on high-quality input data and do not integrate preprocessing steps such as denoising or data selection. Keyframe selection has been explored in related areas like video summarization and annotation, but it is rarely evaluated in the context of 3D tracking performance or downstream behavioral analysis. Likewise, deep learning–based denoising methods have shown strong results in image processing tasks but are underutilized as preprocessing tools in animal tracking pipelines, where frame-level noise or compression artifacts are common due to low-light or high-speed recording setups.

For these reasons, this study aims to systematically evaluate the joint impact of video denoising and keyframe selection on 3D pose estimation performance in freely behaving animals. This approach is novel in that it: (1) Benchmarks deep denoising models directly on behavioral videos prior to 2D keypoint detection; (2) compares unsupervised keyframe selection methods using high-level video features.

This integrated evaluation addresses a key gap in the literature by focusing on improving data quality before model training and triangulation, rather than relying solely on post-hoc corrections. As such, it provides practical guidance for behavioral neuroscience laboratories aiming to improve pose estimation accuracy without increasing annotation burden or retraining complex models.

5 METHODS

The denoising and keyframe selection procedures described in this work are integrated into a larger pipeline designed to extract 3D postural data from freely moving mice. This pipeline begins with multi-view video recordings, followed by 2D pose estimation for each camera view. The resulting 2D keypoints are then processed through denoising and keyframe selection steps to improve the quality and reliability of the data. Finally, the

refined 2D poses are triangulated and optimized to reconstruct accurate 3D postures across time. The goal of this integrated approach is to enhance the robustness of 3D reconstructions, particularly in challenging settings with variable image quality and complex behaviors. A complete diagram of the described pipeline is visible in Figure 1.

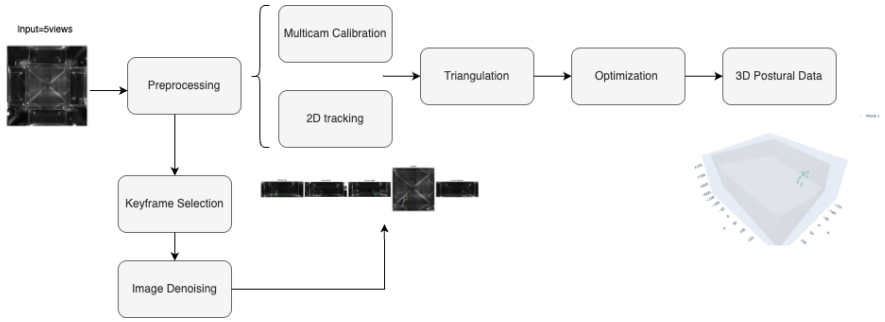

Figure 1: Description of the pipeline used to extract 3D postural data from freely mice

Specifically, SLEAP Pereira et al., 2022 has been used for the data labeling and the 2D predictions of the bottom view and the side view, both views are visible in Figure 2. A total of 750 and 550 annotations have been produced for the side and bottom view respectively.

While ANIPOSE (Pierre Karashchuk et al., 2020) has been used to further optimize the triangulated points by applying spatio-temporal and postural constraints.

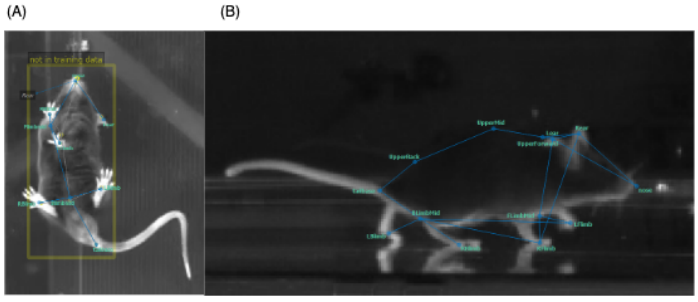

Figure 2: A: example of bottom view label. B: example of side view label

5.1 *Dataset Description*

The dataset employed in this study consists of video sequences extracted from a custom multi-camera pipeline used to record freely moving mice in a controlled arena, a total of 64 videos have been captured with an average duration of 20 minutes per video at 30 fps.

Each original recording comprises five synchronized video streams, Figure 3 top, capturing the animal from four lateral views and one ventral view. To simplify the learning problem and isolate the contribution of visual noise to 2D frame quality, each multi-view video was split into five independent single-view sequences with the use of the napari library Sofroniew et al., 2025, one per camera, resulting in a collection of monocular video streams, Figure 3 bottom.

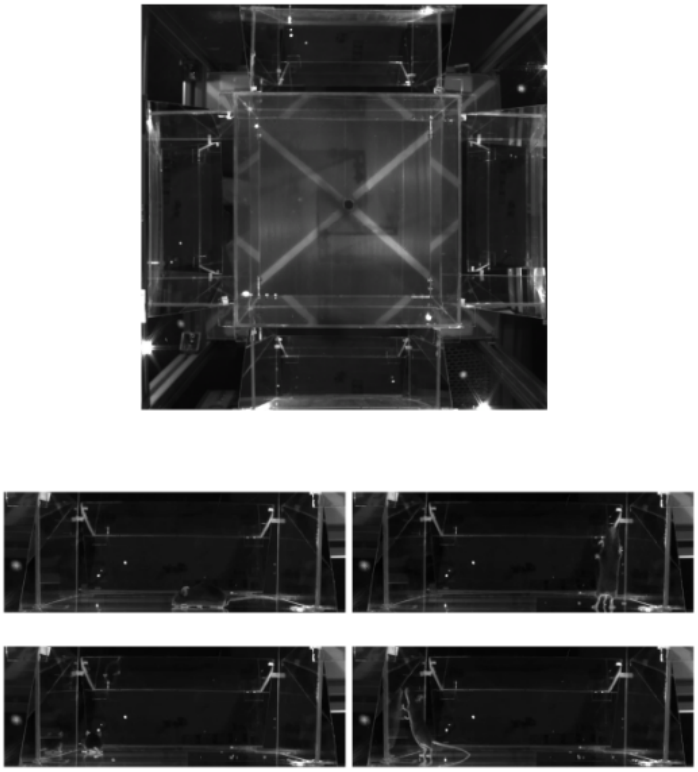

Figure 3: Top: uncropped frame showing all the five views. Bottom: example of cropped frames (single view).

For the purpose of the keyframe selection and denoising, a representative subset of the single-view videos was selected (35000 frames). and tracking points were omitted. This decision allows the models to focus exclusively on enhancing raw pixel quality, independently from any pose estimation system.

To better understand the noise characteristics of the data, two global statistics were computed from a representative video: (1) the Shannon entropy per frame, and (2) the histogram of pixel intensities across all frames. The entropy curve, shown in Figure 4, reveals significant variation

in image complexity, likely corresponding to changes in motion, posture, and lighting. Meanwhile, the intensity histogram in Figure 4 indicates that the dataset is characterized by a strong bias toward low pixel values, confirming the presence of underexposed and noisy frames. These properties underscore the importance of a learned denoising strategy.

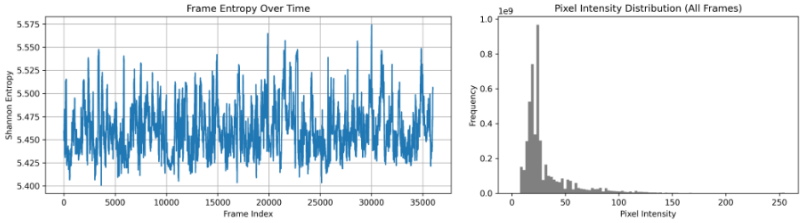

Figure 4: Left: Shannon entropy computed frame-wise across a sample video. High variability suggests changes in structure and illumination. Right: Distribution of pixel intensities across all frames in the same video. The skew toward low intensities reflects naturalistic lighting conditions and sensor noise.

These findings highlight the need for advanced denoising strategies, as traditional pipelines may misinterpret noise as relevant features, especially under low-light and motion-intensive conditions. The motivation for incorporating deep learning-based denoisers stems from this observed data complexity.

5.2 *Keyframe Selection*

Two unsupervised methods were implemented and compared. Both approaches use deep visual features extracted from each frame using a pre-trained ResNet-50 model (He et al., 2015). The produced embeddings, originally of size 2048, are then reduced in dimensionality using Principal Component Analysis (PCA), retaining 90% of the total variance.

In the first method, the reduced feature embeddings are clustered using the k-means algorithm. One keyframe is selected per cluster, defined as the frame whose embedding lies closest to the cluster centroid. This strategy ensures diversity among selected frames and guarantees representativeness by explicitly minimizing intra-cluster variance. This method is widely adopted in video summarization literature (Gong et al., 2014) and aligns well with the distributional properties of deep feature embeddings.

Let $\{z_i\}_{i=1}^{N}$ be the set of PCA-reduced frame features. K-means clusters them into K centroids $\{c_k\}_{k=1}^{K}$. The keyframe f_k for cluster k is selected as:

$$f_k = \arg\min_{i \in \mathcal{C}_k} \|z_i - c_k\|_2$$

where $\mathcal{C}_k$ denotes the set of points assigned to cluster k.

As an alternative to clustering, Farthest Point Sampling was used to promote diversity without relying on centroid-based assignments. FPS begins by selecting an initial frame and then iteratively adds the frame that is farthest (in Euclidean space) from all previously selected frames. This greedy approach explicitly maximizes coverage over the feature space, producing a well-distributed set of keyframes. FPS has been employed effectively in point cloud processing (Qi et al., 2017) and more recently in video frame selection (Xu et al., 2021) due to its simplicity and strong coverage guarantees. Let S be the set of selected keyframe indices. FPS iteratively adds the point f_t that maximizes the minimum distance to the already selected points:

$$f_t = \arg\max_{i \notin S} \min_{j \in S} \|z_i - z_j\|_2$$

This ensures each new keyframe is as far as possible from previously selected ones, promoting coverage of the feature space.

Both approaches are fully unsupervised and operate independently of downstream labels or tracking targets, thus avoiding manual selection bias. Following Biderman et al., 2024, 500 keyframes have been selected for each method, as a compromise between representing behavioral variability and limiting redundancy, ensuring efficient training without overloading the model with near-duplicates frames.

5.3 *Denoising*

5.3.1 *Goals of Denoising*

The rationale for applying denoising techniques prior to 2D keypoint tracking is that sensor noise may be misinterpreted by the model as semantically meaningful features. This risk is particularly high in challenging conditions, such as motion blur or underexposure, where true keypoints or contours are faint or ambiguous. By reducing such noise beforehand, the quality of the 2D predictions can be enhanced, which in turn improves the accuracy and stability of downstream 3D reconstruction.

5.3.2 *Data Preprocessing*

To train the selected denoising models I have used the K-mean keyframes extracted in step 5.2. Thus, the dataset consisted of 500 keyframes containing the most informative frames. Since the recorded videos do not share the same dimensions (width and height), an extra step to ensure consistency across frames was required. During this step each frame is converted from its original shape (C, H, W) to a fixed spatial dimension by

dividing the frame into patches and adding padding to them. The frame, initially of shape (C, H, W), is first padded symmetrically to dimensions (C, H', W') such that both height (H') and width (W') are divisible by the patch size (d). Let

$$\begin{aligned} p_h &= \begin{cases} 0, & \text{if } H \bmod d = 0 \\ d - (H \bmod d), & \text{otherwise} \end{cases}, \\ p_w &= \begin{cases} 0, & \text{if } W \bmod d = 0 \\ d - (W \bmod d), & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

Where p_h, p_w in Equation 1 represents the total padding that we have to add to the respective dimensions. Subsequently, the spatial padding is equally split between the two ends of the image's spatial dimensions.

$$\begin{aligned} p_h = p_h^{\text{top}} + p_h^{\text{bottom}}, \quad p_w = p_w^{\text{left}} + p_w^{\text{right}}, \\ p_h^{\text{top}} = \left\lfloor \frac{p_h}{2} \right\rfloor, \quad p_h^{\text{bottom}} = \left\lceil \frac{p_h}{2} \right\rceil, \\ p_w^{\text{left}} = \left\lfloor \frac{p_w}{2} \right\rfloor, \quad p_w^{\text{right}} = \left\lceil \frac{p_w}{2} \right\rceil \end{aligned} \tag{2}$$

After this operation, Equation 2, we have transformed the original image (C, H, W) into the padded image (C, H', W') where each dimension is perfectly divisible by the patch dimension d. The last step is then to determine the number of patches per image by multiplying the number of patches available for each spatial dimension, Equation 3.

$$N = \frac{H'}{d} \cdot \frac{W'}{d} \tag{3}$$

Once the number of patches of each dimension is determined, we can create the final tensor by rearranging the original frames to have shape: (B, N, d, d), where B is the number of frames, N is the number of patches, and d is their dimensions.

Lastly, the patched tensor is normalized in order to contain values only between 0 and 1. A comparison between the original frame and the patches generated from it is visible in Figure 5.

5.3.3 *Denoising AutoEncoders:*

Autoencoders are a family of models that receive an input x. The input is processed by an encoder $e(x)$, which maps the input into a latent space smaller than the original input space $z = e(x)$. During the course of this process the encoder should be able to learn how to represent the most important features of x in the latent space L. The second component of an autoencoder is the decoder, which takes the latent representation of the

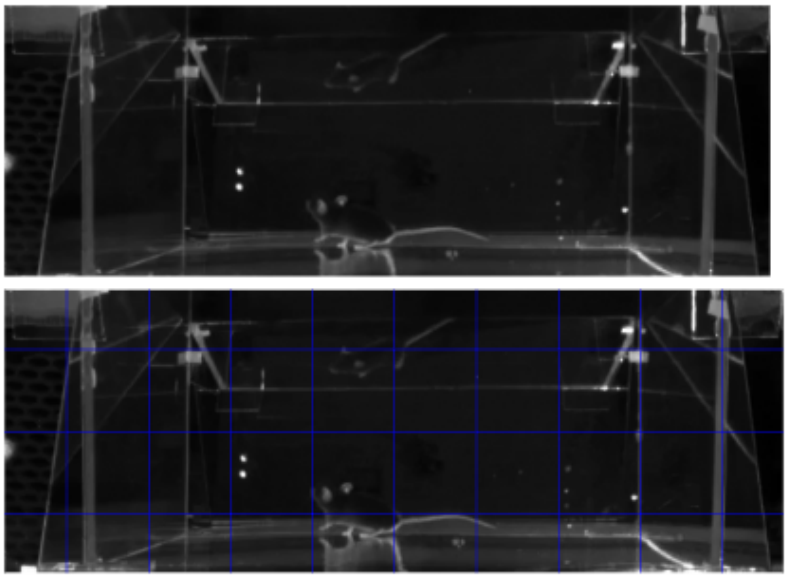

Figure 5: Patches generation. Original frame (top). Patches (64x64) overlaying original frame (bottom)

input, z and it has to reconstruct the input starting from it (Vincent et al., 2008).

Autoencoders can be effectively applied to denoising tasks by providing a noisy version of the input image to the encoder and training the decoder to reconstruct the original, clean image from the resulting latent representation. Through this process, the model learns to transform noisy inputs to noise-free reconstructions.

For this task, inspired by the work of (Vincent et al., 2008), I implemented a convolutional autoencoder composed of an encoder and a decoder module.

The encoder, illustrated in Figure 6 (left), consists of two convolutional blocks, each comprising a convolutional layer followed by a ReLU activation function (Nair & Hinton, n.d.). A Batch Normalization layer (Ioffe & Szegedy, 2015) is applied after the first convolutional block. These layers extract spatial features while progressively reducing the spatial resolution. The output is then flattened and passed through two fully connected linear layers that project the representation into the latent space.

$$z = \text{Linear}\left(\text{ReLU}\left(\text{Linear}\left(\text{Flatten}\left(\text{Conv2D}\left(\text{Conv2D}(x)\right)\right)\right)\right)\right) \tag{4}$$

The decoder, illustrated in Figure 6 (right), takes the latent representation z as input. It first passes through two fully connected layers, each followed by a ReLU activation. The output of the second linear layer is

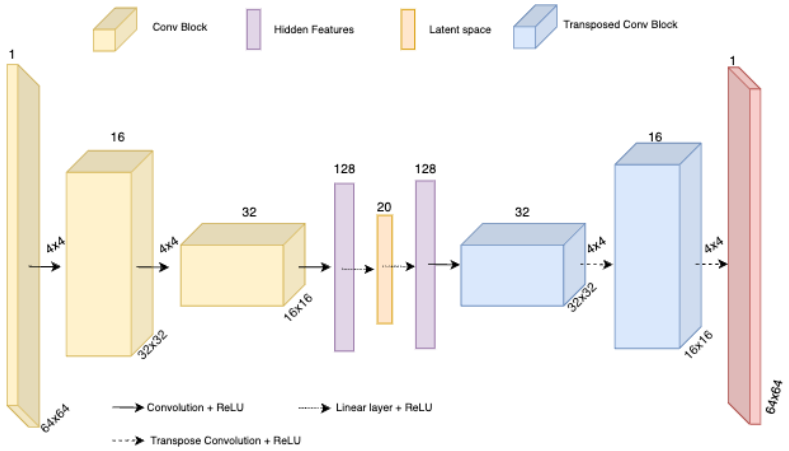

Figure 6: Architecture of the convolutional autoencoder used for denoising, inspired by (Vincent et al., 2008). The encoder reduces the spatial resolution through convolutional blocks and maps the feature maps to a latent vector of size 20 via fully connected layers. The decoder symmetrically reconstructs the image using transpose convolutions.

then reshaped to match the spatial dimensions of the final encoder feature map. This reshaped tensor is subsequently passed through two transposed convolutional blocks to progressively restore the original resolution. As in the encoder, the first transposed convolutional block includes a Batch Normalization layer to stabilize training. This decoding process can be summarized as:

$$x' = \text{TConv}\left(\text{ReLU}\left(\text{BatchNorm}\left(\text{TConv}\left(\text{Reshape}\left(\text{Linear}\left(\text{ReLU}\left(\text{Linear}(z)\right)\right)\right)\right)\right)\right)\right) \tag{5}$$

5.3.4 *Convolutional Blind Denoising Network (CBDNet)*

Deep learning models have shown strong performance in image denoising tasks. However, simple architectures composed of sequential convolutional layers often fail to generalize to real-world noise, primarily due to overfitting to synthetic noise patterns (Soltanayev & Chun, 2018). To address this limitation, (Guo et al., 2019) proposed the Convolutional Blind Denoising Network (CBDNet), which decomposes the denoising process into two separate stages: a *noise estimation subnetwork* and a *non-blind denoising subnetwork* conditioned on the estimated noise.

Inspired by this approach, I implemented and evaluated the CBDNet architecture to denoise frames extracted from naturalistic videos. The goal was to assess its effectiveness within a real-world pipeline and com-

pare its performance against traditional autoencoders and other denoising methods.

The model is designed to address the challenge of real-world image denoising by explicitly modeling the noise distribution. It consists of two main components: aNoise Estimation Network and a Blind Denoising Network, as illustrated in Figure 7.

The Noise Estimation Network (Figure 7A) is a fully convolutional module composed of five consecutive convolutional layers, each with 32 output channels, a 3×3 kernel, and ReLU activation. Its role is to learn a pixel-wise estimation of the noise present in the input image x. This noise map captures spatially varying noise characteristics, which are especially important in naturalistic and sensor-dependent noise scenarios.

The estimated noise map is concatenated with the original noisy image along the channel dimension, forming the input to the second module.

The Blind Denoising Network (Figure 7B) adopts a U-Net architecture (Ronneberger et al., 2015), consisting of a contracting path (encoder) and an expansive path (decoder). The encoder reduces the spatial resolution using strided convolutions, while the decoder progressively restores it via transposed convolutions. Skip connections are included between corresponding layers of the encoder and decoder to facilitate the flow of low-level information. All layers in the denoising network use 3×3 kernels and ReLU activations; batch normalization is omitted following the design in the original CBDNet.

The model outputs a residual image $\hat{r}$, which is added to the original input x to produce the final denoised image $\hat{y}$:

$$\hat{y} = x + \hat{r} \tag{6}$$

This residual learning strategy encourages the network to focus specifically on estimating the noise to be removed, rather than reconstructing the entire image (Kai Zhang et al., 2017).

5.4 *Pyramidal Patch-wise Denoising Network (PRIDNet-P)*

The last model architecture implemented for the denoising experiment is the Pyramid Real Image Denoising Network (PRIDNet) (Zhao et al., 2019). The original PRIDNet architecture was designed for RGB images, where attention mechanisms operate over spatial and channel dimensions. In contrast to the original PRIDNet formulation, which uses an attentional mechanisms between channels, I have decided to implement a local attention within the spatial dimensions of each frame (Tiantian et al., 2024). Specifically, I use a sliding-window mechanism that computes attention

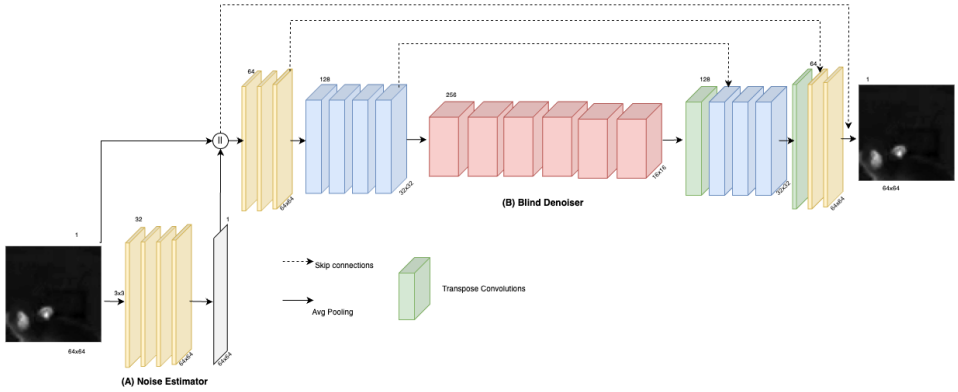

Figure 7: Architecture of CBDNet, inspired by (Guo et al., 2019). (A) Noise Estimation Network. (B) U-Net-based Blind Denoising Network. The noise map predicted by (A) is concatenated with the noisy image and passed to (B).

weights over local neighborhoods, allowing the network to capture fine-grained spatial dependencies without restructuring the input into patch-based channels. Points 1 to 4 highlight the different parts of the variation of the PRDINet implemented in this work following chronologically the part of the model that process the input x. Lastly, the differences between the original model and my implementation are summarized.

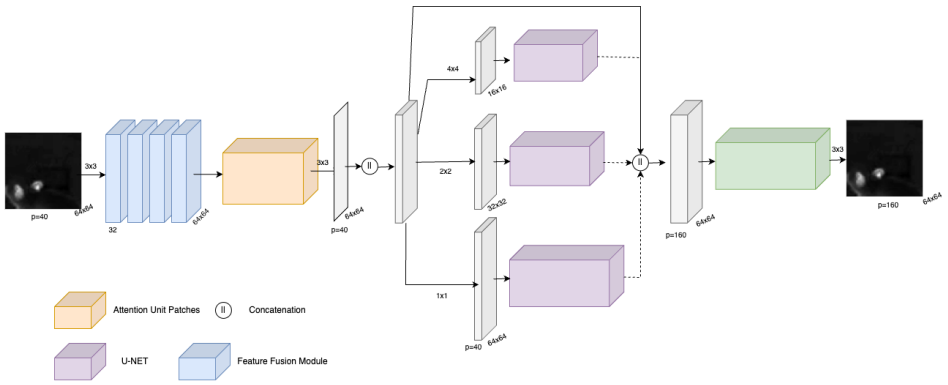

Figure 8: PRIDNet-P, modified version of the original model, (Zhao et al., 2019)

1. Local Context Attention. Let $x \in \mathbb{R}^{C \times H \times W}$ be the input feature map, where C is the number of channels and $H \times W$ are the spatial dimensions. The input is passed through a four-layer convolutional encoder, producing an intermediate representation $x' \in \mathbb{R}^{64 \times H \times W}$. A local attention mechanism is then applied over spatial neighborhoods of x' using a sliding window of size $k \times k$ centered on each spatial position (i, j).

For each position, a local patch $\mathcal{P}_{i,j} \in \mathbb{R}^{C \times k^2}$ is extracted. Concurrently, a per-position attention map $\alpha \in \mathbb{R}^{C \times 1 \times H \times W}$ is generated using a 1×1 convolutional projection and sigmoid activation:

$$\alpha = \sigma\left(W_2 \cdot \mathrm{ReLU}(W_1 \cdot x')\right) \tag{7}$$

These attention weights are broadcast across the k^2 neighborhood and normalized via softmax:

$$w_{i,j} = \mathrm{softmax}(\mathcal{P}_{i,j} \cdot \alpha_{i,j}) \tag{8}$$

The final attended output at each position is computed as a weighted sum:

$$y_{i,j} = \sum_{n=1}^{k^2} w_{i,j}^{(n)} \cdot \mathcal{P}_{i,j}^{(n)} \tag{9}$$

This mechanism allows the model to focus on relevant local features while maintaining full spatial resolution and alignment.

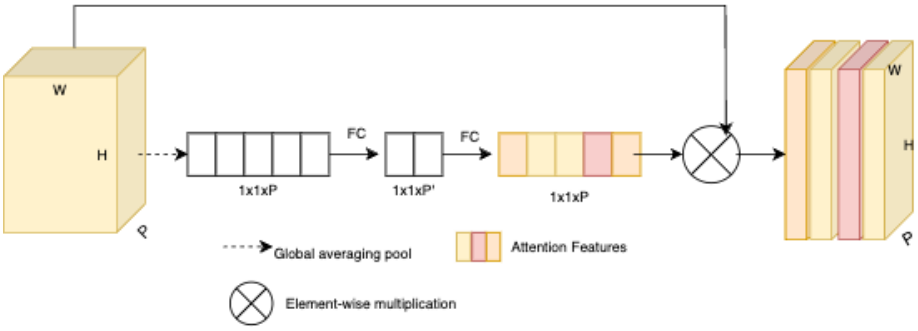

Figure 9: Channel Attention Module. Each channel is a patch of the original frame.

2. Pyramid Pooling and Multi-scale U-Nets. To model noise at different spatial scales, the output of the attention unit is further processed through a *pyramid pooling module*, yielding three scales: $p_1 = x'$, $p_2 = \mathrm{AvgPool}_{2\times2}(x')$, and $p_3 = \mathrm{AvgPool}_{4\times4}(x')$. Each scale is then independently processed by a dedicated U-Net:

$$u_i = \mathrm{UNet}_i(p_i) \quad \text{for } i \in \{1, 2, 3\} \tag{10}$$

Each U-Net follows an encoder-decoder structure with skip connections and varying depth depending on the spatial scale of the input.

3. Multi-branch Feature Fusion. , illustrated in Figure 10 After processing through their respective U-Nets, all outputs are upsampled to a common

spatial resolution and concatenated along the channel axis along with x'. A three-branch convolutional fusion module with kernel sizes 3×3, 5×5, and 7×7 processes the concatenated tensor. Each branch output is weighted by soft attention scores α, β, γ computed via:

$$s = \text{ReLU}(W_s \cdot \text{GAP}(x)) \tag{11}$$

$$[\alpha, \beta, \gamma] = \text{Softmax}([W_\alpha(s), W_\beta(s), W_\gamma(s)]) \tag{12}$$

and the final fused output is:

$$f = \alpha \cdot u_1 + \beta \cdot u_2 + \gamma \cdot u_3 \tag{13}$$

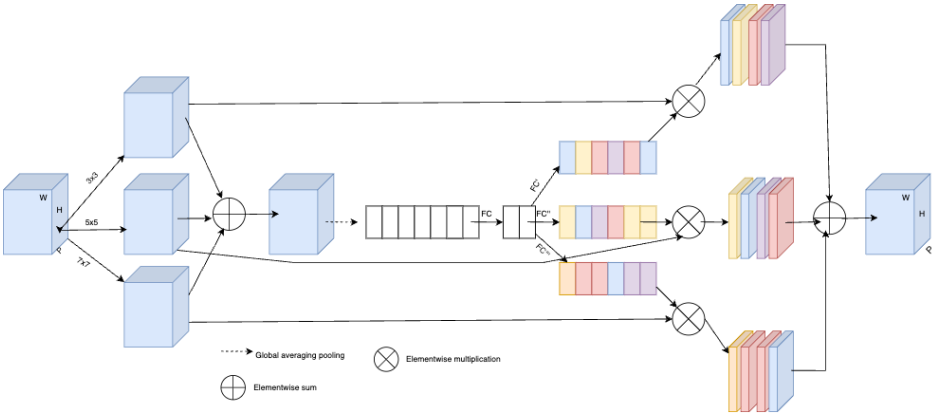

Figure 10: Multi-Channel Feature Fusion

4. Final Output. The fused tensor f is passed through a final 1×1 convolution to compress the channels and produce the denoised output $\hat{x} \in \mathbb{R}^{P \times H \times W}$, reconstructing all patches in parallel.

Summary of Differences. Compared to the original PRIDNet architecture, which relies on patch-as-channel representation and channel attention, this version applies a local attention mechanism over spatial neighborhoods, preserving the native structure of convolutional feature maps. This change enables fine-grained modeling of local context while avoiding the need for explicit patch decomposition. The architecture remains suitable for single-channel grayscale inputs, such as behavior videos, and improves interpretability by aligning attention directly with image structure.

5.5 *Training Regime*

All denoising models presented in this work were trained under a standardized training protocol to ensure comparability across architectures.

The models were trained on batches of patches extracted from keyframes, with synthetic noise applied on-the-fly to simulate realistic degradation in image quality.

To simulate realistic sensor noise, I adopt a physically-inspired noise model that combines signal-dependent and constant components (Guo et al., 2019). Given a clean image $x \in [0, 1]$, the following steps are applied:

1. Inverse Camera Response Function (CRF): The image is linearized to approximate raw sensor values:
$$x_{\text{lin}} = x^{\gamma}, \quad \text{with } \gamma = 2.2$$
2. Shot Noise (Signal-Dependent): Random noise proportional to the signal intensity:
$$\eta_s \sim \mathcal{N}\left(0, \sigma_s^2 \cdot x_{\text{lin}}\right)$$
3. Read Noise (Constant): Additive Gaussian noise independent of the signal:
$$\eta_c \sim \mathcal{N}(0, \sigma_c^2)$$
4. Final Noisy Image:
$$\hat{x}_{\text{lin}} = x_{\text{lin}} + \eta_s + \eta_c, \quad \hat{x} = \text{clip}\left(\hat{x}_{\text{lin}}, 0, 1\right)^{1/\gamma}$$

This formulation approximates real sensor behavior by modeling both photon-related noise (shot noise) and electronic readout noise, producing more challenging and realistic training conditions.

The objective of the training is to reconstruct clean image patches from their noisy counterparts using a combination of reconstruction and perceptual losses.

Each model receives as input a noisy patch x_{noisy}, generated by applying a physically inspired noise model that combines Poisson and Gaussian noise components. The corresponding clean patch x is used as supervision. The loss function $\mathcal{L}_{\text{total}}$ combines Mean Squared Error (MSE) and Structural Similarity Index (SSIM) loss (Zhou Wang et al., 2004), given by:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{SSIM}}. \tag{14}$$

This dual-objective loss encourages both pixel-wise accuracy and perceptual similarity in the reconstruction. The SSIM term, in particular, helps preserve structural features that are perceptually important but not emphasized by MSE alone.

All models were trained using the Adam optimizer (Kingma & Ba, 2017). During training, every 250 steps a qualitative progress was monitored by reconstructing full-resolution images from the patch-wise outputs on a fixed validation image.

5.5.1 *Hyperparameter Tuning*

To identify the optimal configuration for each denoising model (Autoencoder, CBDNet, PRIDNet), I conducted automated hyperparameter tuning using Bayesian optimization (Frazier, 2018) via Weights and Biases (Biewald, 2020) sweeps. Each model has been trained with a different configuration for 10 times total. Then the best configuration for each model has been selected and used for testing.

For the denoising autoencoder, I decided to optimize over the hidden and latent space size, learning rate and batch size, full details of the training are visible in the Table 1.

Hyperparameter	**Type**	**Range / Values**
Latent dimension	Discrete	{8, 16, 32, 64}
Hidden dimension	Discrete	{64, 128, 256}
Learning rate	Continuous	Log-uniform: 10^{-5}–10^{-1}
Batch size	Discrete	{16, 32, 64}

Table 1: Search space for hyperparameter tuning of the Autoencoder model.

For the CBDNet model, the hyperparameter search focused on the learning rate, batch size, and the weight of the SSIM loss term. The learning rate and SSIM weight were sampled from continuous uniform distributions, while the batch size was selected from a predefined discrete set. The full search space is summarized in Table 2.

Hyperparameter	**Type**	**Range / Values**
Learning rate	Continuous	Uniform: 0–0.1
Batch size	Discrete	{16, 32}

Table 2: Search space for hyperparameter tuning of the CBDNet and PRIDNet models.

The same search space was adopted for the Pyramidal Residual and Attention Denoising Network (PRIDNet), ensuring comparability between the two architectures during evaluation.

5.5.2 *Testing*

Following hyperparameter optimization, the best-performing configuration of each denoising model—Autoencoder, CBDNet, and PRIDNet—was selected for final evaluation. To assess their generalization ability and potential impact on downstream behavioral tracking, each model was applied to a previously unseen video sample (frame number: 35000) from

the same experimental setup described in Section 5.1. This allowed for a qualitative and comparative analysis of how well each approach denoises naturalistic input beyond the training distribution. In Figure 11 it is possible to observe a summary of the pipeline used for the model comparison.

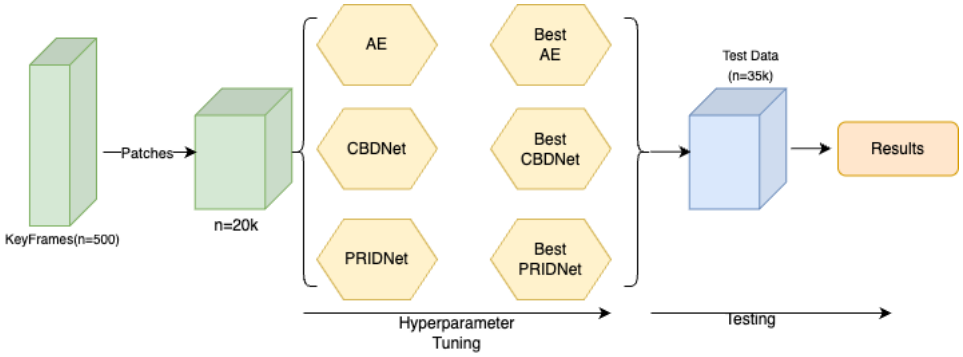

Figure 11: Model Comparison Pipeline

6 RESULTS

6.1 *Results Key-frame Selection*

Figure 12 and Figure 13 visualize the keyframe selection from the two methods in the PCA embedding space. While both strategies cover the major modes of the feature distribution, FPS appears to produce a more sparse representation of the frames, indicating how it struggles to separate them. On the other hand K-mean is able to produce clear clusters that are associated with frames of the mouse in different poses, this indicates that is able to correctly separate meaningful frames.

Figure 12: Keyframes selected via K-means clustering in 2D PCA space. Colors represent cluster assignment; black crosses denote selected keyframes.

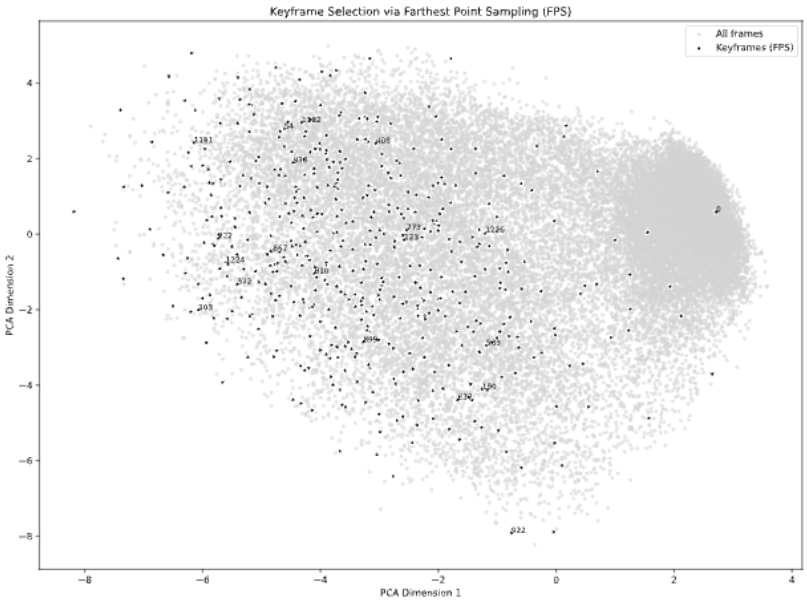

Figure 13: Keyframes selected via farthest point sampling (FPS) in 2D PCA space. Selected points are spread across the embedding, covering the feature space more uniformly.

6.1.1 *Error Analysis for Keyframes Selection*

To further analyze the performances of the two algorithms used during keyframe selection (K-means and FPS); I have conducted a qualitative error analysis by plotting two keyframes that are particularly close in the PCA space and two keyframe that are particularly further away. The intuition is that close keyframes should share a similar structure, while distant keyframes should be different.

As shown in Figure 13, the FPS method selected keyframes that are concentrated in a specific region of the PCA space—primarily on the left—while leaving the right side relatively underrepresented. This clustering tendency results in many selected frames being visually similar, as illustrated in Figure 14. Conversely, when distant keyframes are plotted (Figure 15), they depict markedly different scenes, highlighting the diversity FPS can capture. However, the sparsity in certain regions of the embedding space suggests that FPS may overlook structurally relevant variations in the data.

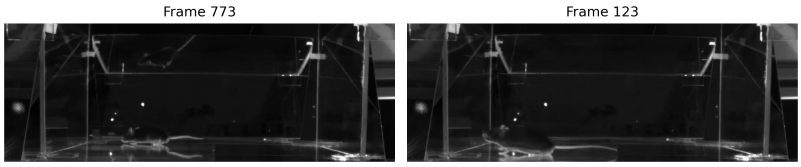

Figure 14: Error analysis FPS close frames

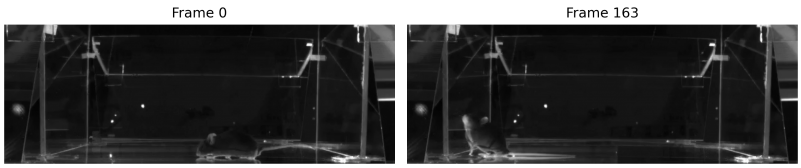

Figure 15: Error analysis FPS distant frames

Unlike FPS-selected keyframes, those obtained via K-means are more evenly distributed across the PCA space, suggesting better overall coverage of the video content. This observation is supported by the qualitative examples in Figures 16 and 17, where keyframes that are close in the PCA space correspond to visually similar scenes, while distant keyframes capture distinct visual contexts. This indicates that K-means effectively captures both redundancy and diversity across the video.

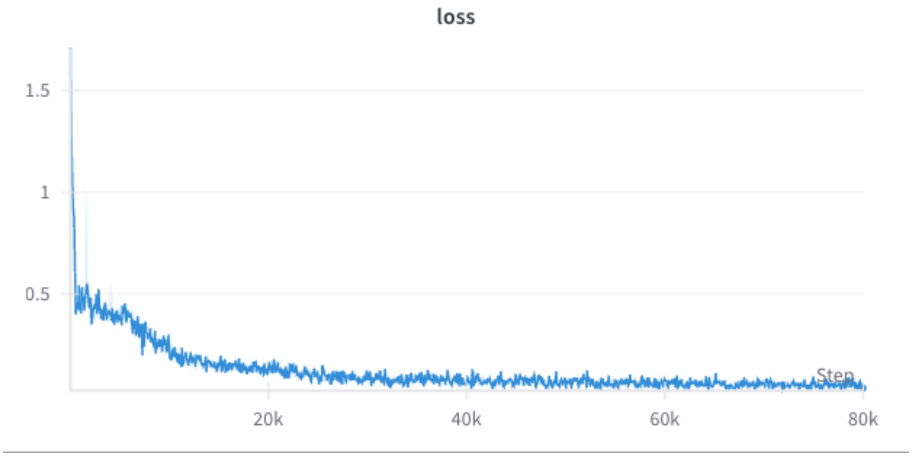

Figure 18: Loss for best AutoEncoder configuration during training

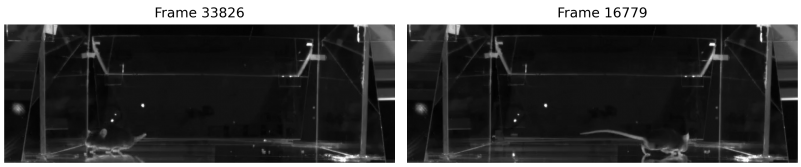

Figure 16: Error analysis KMean close frames

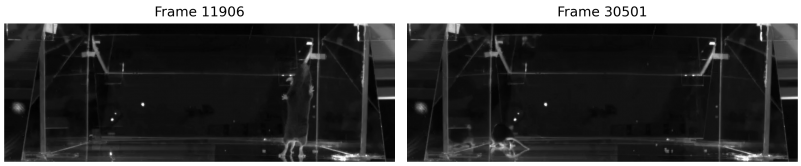

Figure 17: Error analysis KMean distant frames

6.2 *Results from Hyper-parameters Tuning*

6.2.1 *Denoising AutoEncoder*

After 10 sweeps, the best configuration for the denoiser autoencoder was: hidden layer size: 64, latent dimension size: 64, learning rate: 0.002 and batch size: 32.

The model achieved a final loss ($MSE + SSIM$) of: 0.038 (MSE: 0.001, SSIM: 0.3). The total loss is visible in Figure 18, while a qualitative comparison between the initial input and the result post training is visible in Figure 19.

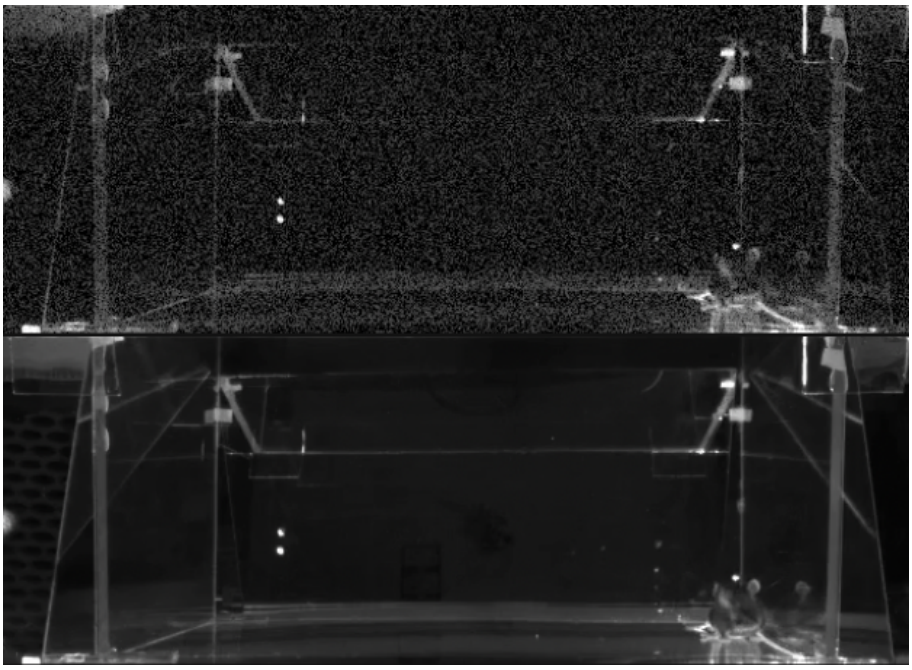

Figure 19: Qualitative Comparison AutoEncoder. (Top) Noisy Image. (Bottom) Reconstructed Image.

While effective at removing noise, the model shows limitations in restoring high-frequency content, which is expected given the simplicity of its bottleneck architecture and the lack of noise modeling component.

6.2.2 *CBDNet*

After 10 sweeps, The best configuration for the CBDNet was: batch size: 32, learning rate: 0.087.

The model achieved a final loss ($MSE + SSIM$) of: 0.038 (MSE:0.0025, SSIM: 0.036). The training curve for the total loss is visible in Figure 20, while in Figure 21.

6.2.3 *PRIDNet*

After 10 sweeps, The best configuration for the PRIDNet was: batch size: 32 and learning rate: 0.002. The model achieved a final loss ($MSE + SSIM$) of: 0.19 (MSE:0.001, SSIM:0.19). The training curve is visible in Figure 22, while in Figure 23 it is possible to notice the comparison between the noisy image and the reconstructed one.

Despite the numerically good performance, the model occasionally produced overly smoothed reconstructions, suggesting that the denoiser may suppress fine-grained details along with noise. This behavior is

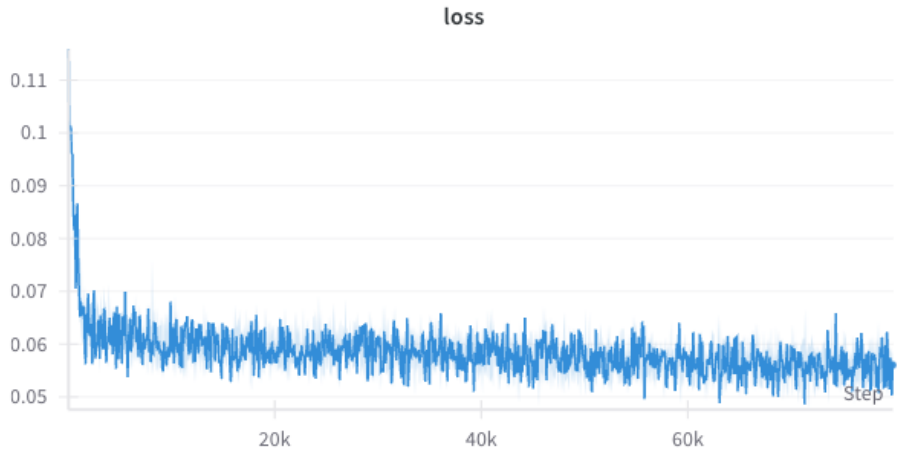

Figure 20: Curve loss for CBDNet best model.

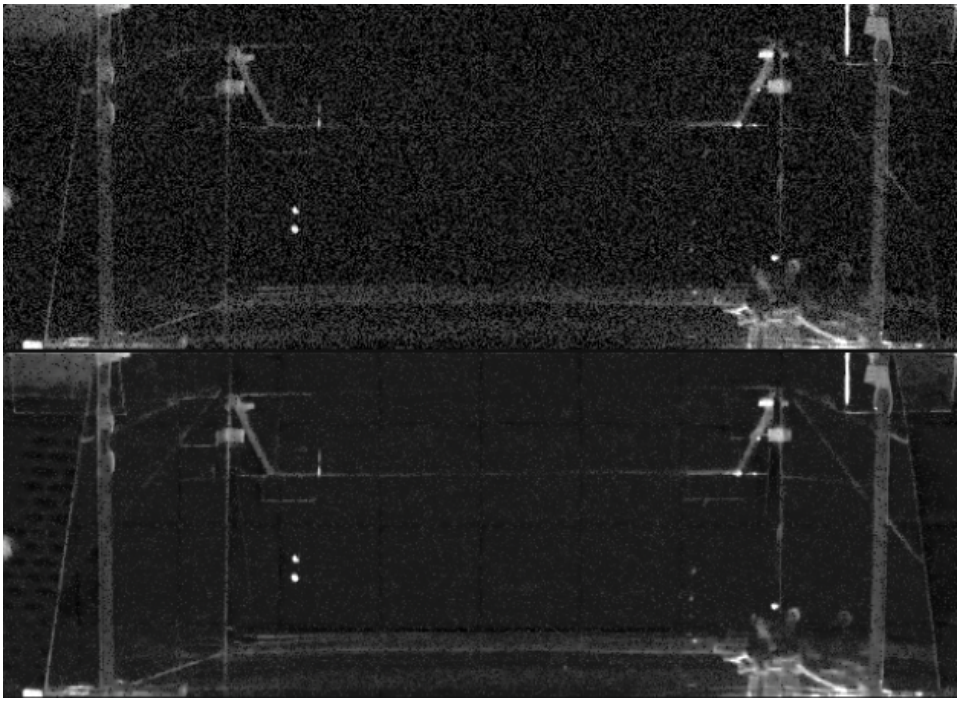

Figure 21: Qualitative Comparison CBDNet. (Top) Noisy Image. (Bottom) Reconstructed Image.

consistent with known limitations of attention-based patch models when trained using pixel-wise losses alone.

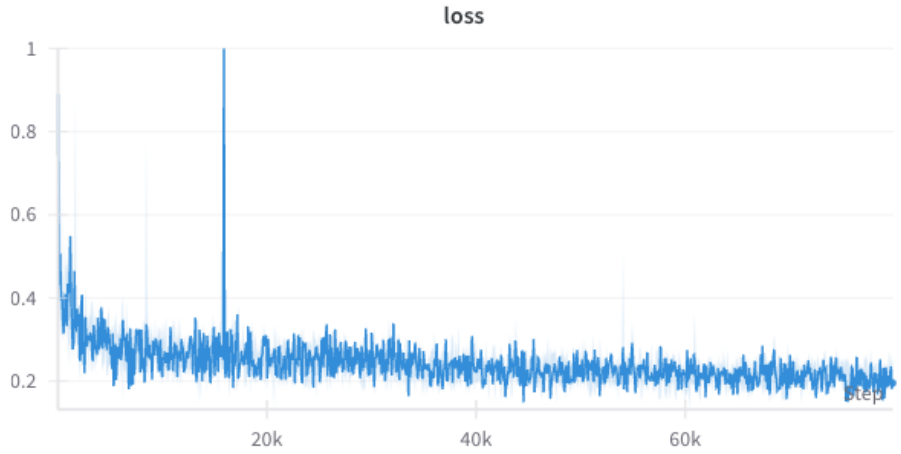

Figure 22: Curve loss for PRIDNet best model.

6.2.4 *Testing*

In Table 3 is it possible to see comparison between the best model configurations and the respective losses.

Model	Batch Size	Learning Rate	Final Loss	MSE [95% CI], SSIM [95% CI]
Autoencoder	32	0.002	0.038	0.001 [0.0008, 0.0012], 0.3 [0.28, 0.32]
CBDNet	32	0.087	0.038	0.0025 [0.0020, 0.0030], 0.036 [0.034, 0.038]
PRIDNet	32	0.002	0.19	0.001 [0.0009, 0.0011], 0.19 [0.17, 0.21]

Table 3: Best hyperparameter configurations and final loss values (MSE + SSIM) for each denoising model after 10 sweep runs.

To further test the models performances and generalization, I have run the best models for each architecture over an unseen video. The video is composed by 35000 frames at 30 fps. The results are visible in Table 4

Model	**Total Loss**	**MSE**	**SSIM**
Best Autoencoder	0.043	0.004	0.041
Best CBDNet	0.045	0.002	0.043
Best PRIDNet	0.250	0.11	0.149

Table 4: Loss values computed on a previously unseen video of 35,000 frames for the best-performing model of each architecture.

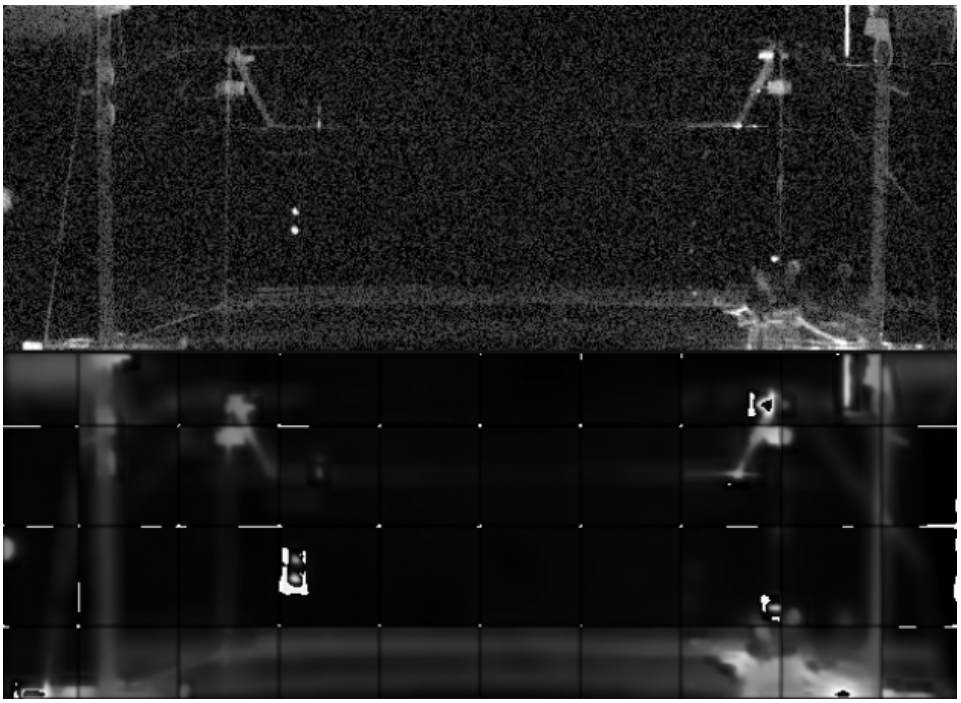

Figure 23: Qualitative Comparison PRIDNet. (Top) Noisy Image. (Bottom) Reconstructed Image.

The results reported in Table 4 highlight notable differences in the behavior of the denoising models when applied to unseen data. Both the Autoencoder and CBDNet achieved comparable total losses, with low mean squared error and relatively minimal SSIM loss, suggesting that they were able to preserve both pixel-level fidelity and structural similarity. In contrast, PRIDNet, despite achieving a very low MSE, exhibited a significantly higher SSIM loss. Qualitative inspection of the output reveals that the model tends to produce overly smooth reconstructions, likely due to the attention mechanism over-regularizing local regions and suppressing high-frequency details.

7 DISCUSSION

The aim of this work was to explore how deep learning feature extraction helps to extract keyframes from naturalistic videos. In addition, it has also been explored how different keyframe selection techniques performed in the PCA space. The comparison between K-means and farthest point sampling (FPS) revealed practical trade-offs in selecting representative video frames. While K-means emphasizes dense regions of the feature space, FPS promotes broader sparsity, leaving many regions of the space

empty and a significant portion of the data isolated. These findings suggest that the choice of keyframe strategy may influence downstream denoising or behavior modeling tasks, particularly in domains with structured variability like animal behavior.

In addition to the cluster distribution analysis, a qualitative error analysis between the two methods (FPS and K-means) revealed that, although both approaches produced visually similar close keyframes, K-means resulted in more semantically diverse distant keyframes. This suggests that K-means better captures the global structure of the embedding space, offering a broader and more representative sampling of the video content.

The second research question that has been explored in this work was to measure the extent to which deep learning image denoising can improve current 3D animal pose estimation pipelines. Current state-of-the-art pipelines for 3D animal pose estimation (Biderman et al., 2024; Pereira et al., 2022; Pierre Karashchuk et al., 2020) typically rely on post-hoc filtering or manual curation to mitigate noise, while little emphasis is placed on pre-processing techniques such as learned denoising.

For example in (Pierre Karashchuk et al., 2020), they use a series of spatio-temporal constraints over the model predictions to limit the effects of noise. Another approach (Biderman et al., 2024), uses semi-supervised learning during inference to penalize the model for estimations that are spatially or temporally unrealistic.

This work proposes an upstream integration of denoising models as a preprocessing step, aimed at improving 2D keypoint predictions before triangulation.

From a denoising perspective, the three models evaluated in this work span distinct design paradigms: basic bottleneck reconstruction (Autoencoder) (Vincent et al., 2008), residual learning with noise estimation (CBDNet) (Guo et al., 2019), and attention-based multi-scale feature integration (PRIDNet) (Zhao et al., 2019). While CBDNet and Autoencoder produced comparable losses, PRIDNet, despite its more sophisticated architecture, performed worse in structural fidelity (SSIM). This is consistent with findings in the image restoration literature, where over-parameterized attention models sometimes over-smooth natural textures when trained on limited or narrow data distributions (Tian et al., 2020).

This work contributes to the advancement of automated behavioral analysis in neuroscience and biology by improving the preprocessing of animal video recordings through deep learning-based denoising. Enhanced video quality can lead to more accurate tracking and 3D pose estimation, which are essential for understanding animal behavior in naturalistic environments (Krakauer et al., 2017).

In this context, better behavioral tracking directly translates to a deeper understanding of brain function, especially in naturalistic settings where behavior is unconstrained and complex. Improved preprocessing reduces noise at the source, potentially leading to more reliable downstream analyses in studies of decision-making, motor control, and social interaction. In the long term, these enhancements can support the development of more accurate animal models of neuropsychiatric disorders and foster better translational research across neuroscience, psychology, and biology.

7.1 *Limitations*

This work has several limitations. First, both keyframe selection and denoising were evaluated using a single dataset recorded from one session involving one animal. This may limit the generalizability of the findings, as the models were not tested across varying lighting conditions, camera perspectives, or subject identities. In particular, real-world behavioral studies often involve substantial variability in arena layouts, animal appearance, and background complexity—factors not captured in the current setup.

Second, while the proposed pipeline is ultimately intended to improve 3D pose estimation, its effectiveness was not tested directly on 3D reconstruction metrics. Although denoising is assumed to benefit triangulation by improving 2D keypoint accuracy, this assumption should be validated in future work using direct 3D reprojection error or behavioral syllable classification accuracy.

Lastly, the methods employed in this study have been sampled by a broader pool of methods for denoising or keyframe selection. For denoising, following (Tian et al., 2020), I focused on deep learning methods for image denoising, leaving aside model based approaches. Several alternative clustering techniques were not explored in this study. Methods such as DBSCAN (Ester et al., n.d.) or hierarchical clustering (Ran et al., 2023) may offer better adaptability to the underlying data structure, particularly when the number of keyframes is not fixed or clusters are unevenly distributed.

7.2 *Future Directions*

Future work should address the current limitations by testing the proposed preprocessing pipeline across multiple animals, sessions, and camera setups to assess generalizability. Additionally, the downstream impact of these preprocessing steps should be explicitly evaluated in terms of 3D pose accuracy, neural decoding performance.

It would also be valuable to explore temporal-aware keyframe selection strategies such as reinforcement learning or contrastive learning–based

methods, which could capture dynamic transitions in behavior more effectively. From a denoising perspective, incorporating multi-frame or video-level models could better exploit temporal information, especially in sequences affected by motion blur or occlusions. Finally, integrating these preprocessing steps into an end-to-end pose estimation framework could allow for joint optimization and greater robustness in noisy, real-world data.

8 CONCLUSION

This thesis presented a comparative study of key-frame selection and denoising techniques applied to grayscale video recordings of freely moving animals with the intent of showing how deep learning techniques can improve classical pose reconstructions pipelines.

A realistic noise model was implemented to simulate sensor degradation, and keyframes were selected using ResNet-based embeddings and clustering/diversity sampling. Each model was trained with hyperparameter optimization, evaluated both quantitatively and qualitatively, and tested on unseen behavioral data.

The Autoencoder and CBDNet models achieved similarly low reconstruction losses, with CBDNet showing a slight edge in structural similarity. In contrast, PRIDNet, despite its architectural complexity, suffered from oversmoothing and higher SSIM loss. These results suggest that simpler models may generalize better under limited data regimes or when high-frequency detail is essential.

Given the sensitivity of 2D-to-3D pose estimation pipelines to video quality, integrating a denoising step may offer practical improvements, particularly under suboptimal recording conditions. While this work did not directly measure downstream tracking improvements, it lays the groundwork for incorporating learned pre-processing into scientific tracking workflows such as SLEAP and Anipose.

It is important to highlight that the evaluation focused on reconstruction loss rather than pose estimation accuracy. Moreover, the models were trained on limited data and tested on a single unseen video, which may not capture the full variability of experimental setups.

Future directions include evaluating the effect of denoising on keypoint detection accuracy, integrating perceptual or task-specific losses, and testing in larger-scale, multi-animal, or multi-camera datasets.

REFERENCES

Alexander Mathis, Mathis, A., Pranav Mamidanna, Mamidanna, P., Kevin M. Cury, Cury, K. M., Taiga Abe, Abe, T., Venkatesh N. Murthy, Murthy, V. N., Mackenzie Weygandt Mathis, Mathis, M. W., Matthias Bethge, & Bethge, M. (2018). DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning [MAG ID: 2887114371]. *Nature Neuroscience*, *21*(9), 1281–1289. https://doi.org/10.1038/s41593-018-0209-y

André E. X. Brown, Brown, A. E., Benjamin de Bivort, & de Bivort, B. L. (2018). Ethology as a physical science [MAG ID: 2797173239 S2ID: 2bac53cd5b78dd6c8d740088cf9fa3baboe91947]. *Nature Physics*, *14*(7), 653–657. https://doi.org/10.1038/s41567-018-0093-0

Apostolidis, E., Adamantidou, E., Metsai, A. I., Mezaris, V., & Patras, I. (2021). Video Summarization Using Deep Neural Networks: A Survey. *Proceedings of the IEEE*, *109*(11), 1838–1863. https://doi.org/10.1109/JPROC.2021.3117472

Biderman, D., Whiteway, M. R., Hurwitz, C., Greenspan, N., Lee, R. S., Vishnubhotla, A., Warren, R., Pedraja, F., Noone, D., Schartner, M. M., Huntenburg, J. M., Khanal, A., Meijer, G. T., Noel, J.-P., Pan-Vazquez, A., Socha, K. Z., Urai, A. E., Cunningham, J. P., Sawtell, N. B., & Paninski, L. (2024). Lightning Pose: Improved animal pose estimation via semi-supervised learning, Bayesian ensembling and cloud-native open-source tools [Publisher: Nature Publishing Group]. *Nature Methods*, *21*(7), 1316–1328. https://doi.org/10.1038/s41592-024-02319-1

Biewald, L. (2020). Experiment tracking with weights and biases. https://www.wandb.com/

Caleb Weinreb, Mohammed Abdal Monium Osman, Libby Zhang, Sherry Lin, Jonah Pearl, Sidharth Annapragada, Eli Benjamin Conlin, Winthrop F. Gillis, Maya Jay, Shaokai Ye, Alexander Mathis, Mackenzie Weygandt Mathis, Talmo Pereira, Scott W. Linderman, & Sandeep Robert Datta. (2023). Keypoint-MoSeq: Parsing behavior by linking point tracking to pose dynamics [MAG ID: 4327861594 S2ID: 9e523dcb44e734beef54230a1ba4d46e4ddc7f02]. *bioRxiv*. https://doi.org/10.1101/2023.03.16.532307

Datta, S. R., Anderson, D. J., Branson, K., Perona, P., & Leifer, A. (2019). Computational Neuroethology: A Call to Action [Publisher: Elsevier]. *Neuron*, *104*(1), 11–24. https://doi.org/10.1016/j.neuron.2019.09.038

Elad, M., Kawar, B., & Vaksman, G. (2023, January). Image Denoising: The Deep Learning Revolution and Beyond – A Survey Paper –

[arXiv:2301.03362 [eess]]. https://doi.org/10.48550/arXiv.2301.03362

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (n.d.). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.

Frazier, P. I. (2018, July). A Tutorial on Bayesian Optimization [arXiv:1807.02811 [stat]]. https://doi.org/10.48550/arXiv.1807.02811

Gong, B., Chao, W.-L., Grauman, K., & Sha, F. (2014). Diverse Sequential Subset Selection for Supervised Video Summarization. *Advances in Neural Information Processing Systems*, *27*. Retrieved May 19, 2025, from https://proceedings.neurips.cc/paper_files/paper/2014/hash/5d3b9e06117de70a7e5076cc3ed89e18-Abstract.html

Guo, S., Yan, Z., Zhang, K., Zuo, W., & Zhang, L. (2019, April). Toward Convolutional Blind Denoising of Real Photographs [arXiv:1807.04686 [cs] version: 2]. https://doi.org/10.48550/arXiv.1807.04686

He, K., Zhang, X., Ren, S., & Sun, J. (2015, December). Deep Residual Learning for Image Recognition [arXiv:1512.03385 [cs]]. https://doi.org/10.48550/arXiv.1512.03385

Ioffe, S., & Szegedy, C. (2015, March). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [arXiv:1502.03167 [cs]]. https://doi.org/10.48550/arXiv.1502.03167

Kai Zhang, Zhang, K., Zhang, K., Kai Zhang, Wangmeng Zuo, Zuo, W., Yunjin Chen, Yunjin Chen, Chen, Y., Deyu Meng, Meng, D., Lei Zhang, & Zhang, L. (2017). Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising [MAG ID: 2508457857]. *IEEE Transactions on Image Processing*, *26*(7), 3142–3155. https://doi.org/10.1109/tip.2017.2662206

Kennedy, A. (2022). The what, how, and why of naturalistic behavior. *Current Opinion in Neurobiology*, *74*, 102549. https://doi.org/10.1016/j.conb.2022.102549

Kingma, D. P., & Ba, J. (2017, January). Adam: A Method for Stochastic Optimization [arXiv:1412.6980 [cs]]. https://doi.org/10.48550/arXiv.1412.6980

Krakauer, J. W., Ghazanfar, A. A., Gomez-Marin, A., MacIver, M. A., & Poeppel, D. (2017). Neuroscience Needs Behavior: Correcting a Reductionist Bias [Publisher: Elsevier]. *Neuron*, *93*(3), 480–490. https://doi.org/10.1016/j.neuron.2016.12.041

Liang, H., Li, J., Bai, T., Huang, X., Sun, L., Wang, Z., He, C., Cui, B., Chen, C., & Zhang, W. (2024, August). KeyVideoLLM: Towards Large-scale Video Keyframe Selection [arXiv:2407.03104 [cs]]. https://doi.org/10.48550/arXiv.2407.03104

Lukas von Ziegler, von Ziegler, L., Oliver Sturman, Sturman, O., Johannes Bohacek, & Bohacek, J. (2021). Big behavior: Challenges and opportunities in a new era of deep behavior profiling [MAG ID: 3037927687]. *Neuropsychopharmacology*, *46*(1), 33–44. https://doi.org/10.1038/s41386-020-0751-7

Mathis, M. W., & Mathis, A. (2020). Deep learning tools for the measurement of animal behavior in neuroscience. *Current Opinion in Neurobiology*, *60*, 1–11. https://doi.org/10.1016/j.conb.2019.10.008

Nair, V., & Hinton, G. E. (n.d.). Rectified Linear Units Improve Restricted Boltzmann Machines.

Pereira, T. D., Shaevitz, J. W., & Murthy, M. (2020). Quantifying behavior to understand the brain [Publisher: Nature Publishing Group]. *Nature Neuroscience*, *23*(12), 1537–1549. https://doi.org/10.1038/s41593-020-00734-z

Pereira, T. D., Tabris, N., Matsliah, A., Turner, D. M., Li, J., Ravindranath, S., Papadoyannis, E. S., Normand, E., Deutsch, D. S., Wang, Z. Y., McKenzie-Smith, G. C., Mitelut, C. C., Castro, M. D., D'Uva, J., Kislin, M., Sanes, D. H., Kocher, S. D., Wang, S. S.-H., Falkner, A. L., … Murthy, M. (2022). SLEAP: A deep learning system for multi-animal pose tracking [Publisher: Nature Publishing Group]. *Nature Methods*, *19*(4), 486–495. https://doi.org/10.1038/s41592-022-01426-1

Pierre Karashchuk, Karashchuk, P., Katie L. Rupp, Rupp, K. L., Dickinson, P. S., Evyn S Dickinson, Dickinson, E. S., Sarah Walling-Bell, Sanders, E., Elischa Sanders, Azim, E., Eiman Azim, Brunton, B. W., Bingni W. Brunton, Tuthill, J. C., & John C. Tuthill. (2020). Anipose: A toolkit for robust markerless 3D pose estimation [MAG ID: 3031024919 S2ID: 9a071e7941bb845c70dd1a5cbc7435a1e034c0eb]. *bioRxiv*. https://doi.org/10.1101/2020.05.26.117325

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems*, *30*. Retrieved May 19, 2025, from https://proceedings.neurips.cc/paper/2017/hash/d8bf84be3800d12f74d8b05e9b89836f-Abstract.html

Ran, X., Xi, Y., Lu, Y., Wang, X., & Lu, Z. (2023). Comprehensive survey on hierarchical clustering algorithms and the recent developments. *Artificial Intelligence Review*, *56*(8), 8219–8264. https://doi.org/10.1007/s10462-022-10366-3

Ronneberger, O., Fischer, P., & Brox, T. (2015, May). U-Net: Convolutional Networks for Biomedical Image Segmentation [arXiv:1505.04597 [cs]]. https://doi.org/10.48550/arXiv.1505.04597

Sofroniew, N., Lambert, T., Bokota, G., Nunez-Iglesias, J., Sobolewski, P., Sweet, A., Gaifas, L., Evans, K., Burt, A., Doncila Pop, D., Yamauchi, K., Weber Mendonça, M., Buckley, G., Vierdag, W.-M., Royer, L., Can Solak, A., Harrington, K. I. S., Ahlers, J., Althviz Moré, D., … Zhao, R. (2025, January). Napari: A multi-dimensional image viewer for Python. https://doi.org/10.5281/zenodo.14719463

Soltanayev, S., & Chun, S. Y. (2018). Training deep learning based denoisers without ground truth data. *Advances in Neural Information Processing Systems*, *31*. Retrieved May 12, 2025, from https://proceedings.neurips.cc/paper/2018/hash/c0560792e4a3c79e62f76cbf9fb277dd-Abstract.html

Tan, K., Zhou, Y., Xia, Q., Liu, R., & Chen, Y. (2024, January). Large Model based Sequential Keyframe Extraction for Video Summarization [arXiv:2401.04962 [cs]]. https://doi.org/10.48550/arXiv.2401.04962

Tang, X., Qiu, J., Xie, L., Tian, Y., Jiao, J., & Ye, Q. (2025, February). Adaptive Keyframe Sampling for Long Video Understanding [arXiv:2502.21271 [cs]]. https://doi.org/10.48550/arXiv.2502.21271

Tian, C., Fei, L., Zheng, W., Xu, Y., Zuo, W., & Lin, C.-W. (2020). Deep learning on image denoising: An overview. *Neural Networks*, *131*, 251–275. https://doi.org/10.1016/j.neunet.2020.07.025

Tiantian, W., Hu, Z., & Guan, Y. (2024). An efficient lightweight network for image denoising using progressive residual and convolutional attention feature fusion [Publisher: Nature Publishing Group]. *Scientific Reports*, *14*(1), 9554. https://doi.org/10.1038/s41598-024-60139-x

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th international conference on Machine learning*, 1096–1103. https://doi.org/10.1145/1390156.1390294

Xu, C., Fu, Y., Liu, C., Wang, C., Li, J., Huang, F., Zhang, L., & Xue, X. (2021). Learning Dynamic Alignment via Meta-Filter for Few-Shot Learning, 5182–5191. Retrieved May 19, 2025, from https://openaccess.thecvf.com/content/CVPR2021/html/Xu_Learning_Dynamic_Alignment_via_Meta-Filter_for_Few-Shot_Learning_CVPR_2021_paper.html

Zhao, Y., Jiang, Z., Men, A., & Ju, G. (2019). Pyramid Real Image Denoising Network. *2019 IEEE Visual Communications and Image Processing (VCIP)*, 1–4. https://doi.org/10.1109/VCIP47243.2019.8965754

Zhou Wang, Bovik, A., Sheikh, H., & Simoncelli, E. (2004). Image quality assessment: From error visibility to structural similarity [Publisher: Institute of Electrical and Electronics Engineers (IEEE)]. *IEEE Transactions on Image Processing*, *13*(4), 600–612. https://doi.org/10.1109/tip.2003.819861