UNSUPERVISED COORDINATE-BASED VIDEO DENOISING

Mary Aiyetigbo[⋆] *Dineshchandar Ravichandran*[⋆] *Reda Chalhoub*[†]
Peter Kalivas[†] *Feng Luo*[⋆] *Nianyi Li*[⋆]

[⋆] Clemson University, School of Computing
[†]Medical University of South Carolina, Department of Neuroscience

ABSTRACT

In this paper, we introduce a novel unsupervised video denoising deep learning approach that can help to mitigate data scarcity issues and shows robustness against different noise patterns, enhancing its broad applicability. Our method comprises three modules: a Feature generator creating features maps, a Denoise-Net generating denoised but slightly blurry reference frames, and a Refine-Net re-introducing high-frequency details. By leveraging the coordinate-based network, we can greatly simplify the network structure while preserving high-frequency details in the denoised video frames. Extensive experiments on both simulated and real-captured demonstrate that our method can effectively denoise real-world calcium imaging video sequences without prior knowledge of noise models and data augmentation during training.

***Index Terms*—** video denoising, unsupervised, implicit neural representation

1. INTRODUCTION

Video denoising is critical in computer vision, with various applications such as video surveillance, medical imaging, and autonomous driving. Supervised Convolutional Neural Networks (CNNs) have been successful in solving this problem by learning the mapping between noisy and clean images from large datasets [1, 2, 3, 4, 5]. However, accessing such large datasets can be challenging, particularly in domains like medical imaging, where noisy-clean pairs might be unrealistic. Likewise, in contexts like neural activity recordings, where crucial signals like neuron spikes are detectable in only a few frames, denoising short sequences becomes crucial.

To address the challenge of noisy-clean pair dataset, unsupervised video denoising methods [8, 9, 10, 7] have emerged in recent years, which do not require labeled data for training. Most of these self-supervised techniques necessitate using synthetic datasets for training, which involves introducing random noise to clean images during training, or they depend on prior knowledge of the noise characteristics [7, 10]. Also, many unsupervised methods utilize blind-spot strategies, such as omitting the central pixel [8, 7] or the central frame [6], to train their models. However, these strategies often require lengthy video sequences or extensive data augmentation to enlarge the training dataset, making them less effective for short video clips due to the potential inadequacy of data for the models to accurately learn the underlying noise model, as shown in Fig. 1.

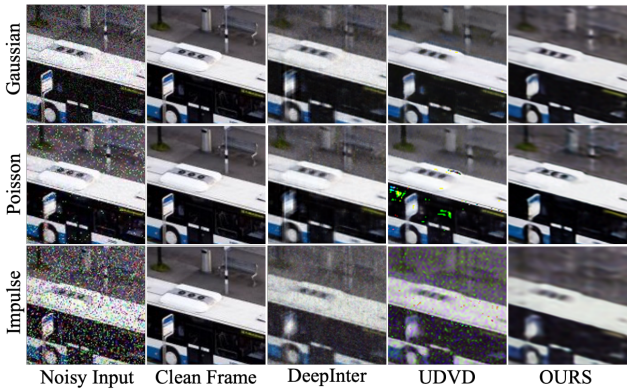

Fig. 1: **Denoising results of our method, DeepInterpolation (DeepInter) [6], and UDVD [7] (unsupervsied SOTA) on 10-frame videos.** Our method demonstrates robust video denoising performance across different noise types.

Our network architecture, shown in Fig. 2, comprises three key components, each designed to effectively denoise video data. First, the feature generator $\mathcal{F}$ renders feature maps that align with the coordinates of the input frames. Second, the Denoise-Net $\mathcal{D}$ utilizes these feature maps to produce denoised yet slightly blurry reference frames. Finally, the Refine-Net $\mathcal{R}$ restores high-frequency details to the denoised frames, enhancing overall image clarity. The network is trained by minimizing the discrepancy between the generated denoised frame, the input noisy frame, and the refined final output frame, ensuring both efficient denoising and preservation of the original video data's integrity and quality. To streamline the network architecture and enhance training efficiency, we incorporate coordinate-based networks, as referenced in [11, 12, 13, 14, 15], into both $\mathcal{D}$ and $\mathcal{R}$. We conduct comprehensive experiments on a diverse range of noisy videos, including both simulated and real-captured footage. Compared to the state-of-the-art unsupervised method, our approach demonstrates superior performance in effectively correcting noise.

2. RELATED WORKS

Supervised Video Denoising approaches, such as those presented in [16, 17, 18, 3], have achieved state-of-the-art results in both single image and video denoising, requiring clean images for model training. Two-stage denoising without explicit motion estimation was explored by DVDNet [1], ViDeNN [18], and FastDVDnet [2]. The transformer-based approach was used by [4, 5] to achieve state-of-the-art video denoising. However, these models' reliance on unrealistic noisy/clean pairs is a drawback, especially in medical imaging, where clean ground truth images are seldom available.

Unsupervised Video Denoising methods leverage noisy videos as both inputs and targets [19, 20, 20, 21]. Blind spot techniques were used in [8, 10, 7] to estimate the underlying clean signal. While UDVD [7] achieves state-of-the-art results, their methodology necessitates noise addition at every model iteration and substantial data augmentation. Furthermore, the DeepInterpolation algorithm by Lecoq et al. [6] required over 200,000 data samples and showed limitations in generalizability. In contrast to these methods, our approach can effectively denoise short video sequences and generalize to different types of noisy videos without requiring excessive data augmentation or iterative noise addition.

Implicit Neural Representation also known as coordinate-based representations, utilize fully connected neural networks to associate input coordinates with their corresponding signal values. They have shown remarkable utility across a range of tasks, including view synthesis [22], image representation [14, 15], and 3D shape representation [23]. One significant development in this domain is the Sinusoidal Representation Networks (SIREN) [15], which leverage periodic activation functions to encode positional information and model complex natural signals with high precision.

3. UNSUPERVISED COORDINATE-BASED VIDEO DENOISING

Given a sequence of noisy video frames $\{I_t|t = 1, 2, \ldots, N\}$ and their corresponding coordinates $\{G_t|t = 1, 2, \ldots, N\}$, where $G_t(\mathbf{p}_t)$ represents the coordinates of pixel $\mathbf{p}_t = (x, y, t)$ in the noisy frame I_t, our objective is to recover the noise-free video frames $\{J_t|t = 1, 2, \ldots, N\}$. Our approach consists of three primary components: a feature generator $\mathcal{F}_\theta$, a Denoise-Net $\mathcal{D}_\phi$, and a Refine-Net $\mathcal{R}_\eta$, as illustrated in Fig. 2.

3.1. Feature Generator $\mathcal{F}_\theta$

Our Feature Generator processes a batch of uniformly sampled coordinate grids $\{G_t|t = 1, ..., B\}$, to generate a corresponding batch of feature maps $\{F_t|t = 1, ..., B\}$, where $F_t \in \mathbb{R}^{H\times W\times C}$. Here, B is the batch size, while C is the number of feature channels. Positional encoding [22, 24, 25] is applied to each coordinate $\mathbf{p}_t = (x, y, t)$ in G_t before they are passed into $\mathcal{F}_\theta$. This encoding step transforms low-dimensional input coordinates into a higher-dimensional space, enabling the model to better learn and represent high-frequency details inherent in the image data. Equation 1 gives the positional encoding function we adopt.

$$\gamma(\mathbf{p}_t) = [\sin(2^0\pi\mathbf{p}_t), \cos(2^0\pi\mathbf{p}_t), ..., \sin(2^{L-1}\pi\mathbf{p}_t), \cos(2^{L-1}\pi\mathbf{p}_t)]. \quad (1)$$

L is a hyperparameter that controls the level of detail or high-frequency information in the output. By selecting a smaller value for L, we can effectively reduce the level of high-frequency noise in the image data, as noise often manifests as high-frequency information. For our experiments, we set $L = 30$. The input coordinates, normalized to the range $[-1, 1]$ using a mesh grid, are passed through the encoding function $\gamma(.)$. As depicted in Equation 2, the resulting high-dimensional output $\gamma(G_t) \in \mathbb{R}^{H\times W\times 6L}$ is subsequently fed into the feature generator $\mathcal{F}_\theta$ to generate feature maps $F_t \in \mathbb{R}^{H\times W\times C}$, corresponding to each noisy frame.

$$F_t = \mathcal{F}_\theta(\gamma(G_t)), \quad (2)$$

where C denotes the channel size of the output features.

Our feature generator comprises six convolution layers, and each layer has 256 feature channels with batch normalization (BN) applied solely to the first two layers. ReLU activation is employed for all layers except for the last one.

3.2. Denoiser $\mathcal{D}_\phi$

The denoiser network takes the concatenated feature maps output from the feature generator as input, and generates a denoised central frame $\hat{I}_B$:

$$\hat{I}_B = \mathcal{D}_\phi([F_1, F_2, ...F_B]), \quad (3)$$

where the concatenation is applied along the feature channel dimension. This allows $\mathcal{D}_\phi$ to learn spatial-temporal patterns along the neighboring time frames. Note that the output of $\mathcal{D}_\phi$ in Equation 3 may be somewhat blurred because we've set a low L in the feature generator.

The architecture of $\mathcal{D}_\phi$ includes 6 convolutional layers. ReLU activation is used in the first five layers, and a sigmoid activation function is used in the last layer. Unlike the feature generator, batch normalization is not applied at this stage. Each layer uses 256 filters with a kernel size of 3, except for the last two layers, which have 96 filters and the number of color channel filters, respectively, with kernel sizes of 1.

3.3. Refine-Net $\mathcal{R}_\eta$

The refine-net is built upon the backbone of the Sinusoidal Representation Networks (SIREN) [15], which is a type of coordinate-based network that uses periodic activation functions, particularly sine functions, in place of traditional activation functions like ReLU. SIREN's unique characteristic lies

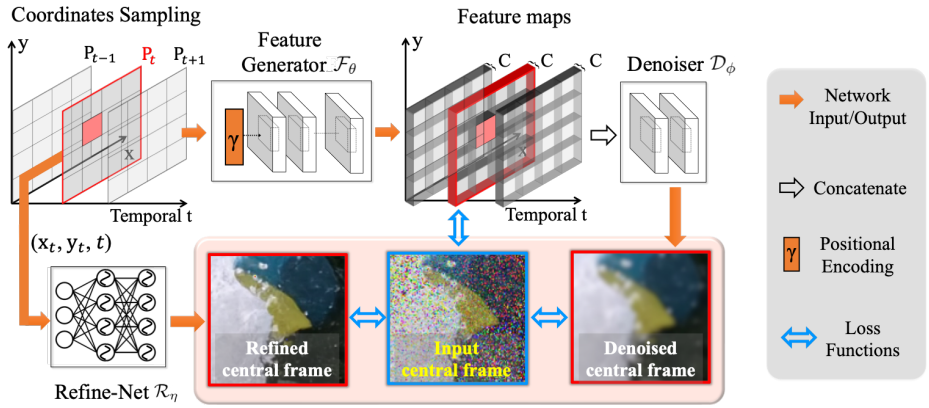

Fig. 2: The pipeline consists of three main components: $\mathcal{F}$ renders feature maps matching the input frame coordinates; $\mathcal{D}$ utilizes these feature maps to generate denoised yet slightly blurry reference frames; and $\mathcal{R}$ reintroduces high-frequency details to enhance the clarity of the denoised frames. The entire network is trained by optimizing the difference between the generated denoised frame, the input noisy frame, and the final refined output frame, ensuring efficient and accurate video denoising.

in its ability to naturally model the high-frequency details of complex patterns by leveraging its intrinsic periodic activation functions. In our context, $\mathcal{R}_\eta$ uses the SIREN network to take the coordinates grid of the central frame as input and generate the refined image $\hat{I}_R$ as shown inn Equation 4.

$$\hat{I}_R = \mathcal{R}_\eta(G_t^c), \tag{4}$$

where G_t^c is the coordinates of the central frames in the input batch. The use of a SIREN-based refine-net is especially advantageous in our scenario because it aids in further enhancing the denoised output from $\mathcal{D}_\phi$. This enhancement includes improving the finer details and fixing any blurring that occurred during the denoising stage. The output from $\mathcal{R}_\eta$ represents the final denoised and refined video frames.

4. NETWORK OPTIMIZATION

Given the unsupervised nature of our architecture, the network optimization problem is highly non-convex with a vast parameter search space. To navigate this challenge, we propose a two-step network optimization strategy that exploits the structural similarity between neighboring frames to reconstruct the central frame.

4.1. First Stage: Joint Training of the Feature Generator $\mathcal{F}_\theta$ and Denoiser $\mathcal{D}_\phi$

In the first stage, we jointly train the feature generator $\mathcal{F}_\theta$ and the denoiser $\mathcal{D}_\phi$ in an end-to-end fashion. The objective function for this stage is composed of two parts. The first part is the l_1 loss in Equation 5, which measures the difference between the denoised central frame $\hat{I}_B$ and the central frame I^c of the input batch of noisy frames $\{I_t | t = 1, 2, \ldots, B\}$:

$$\mathcal{L}_\mathcal{D} = ||\hat{I}_B - I^c||. \tag{5}$$

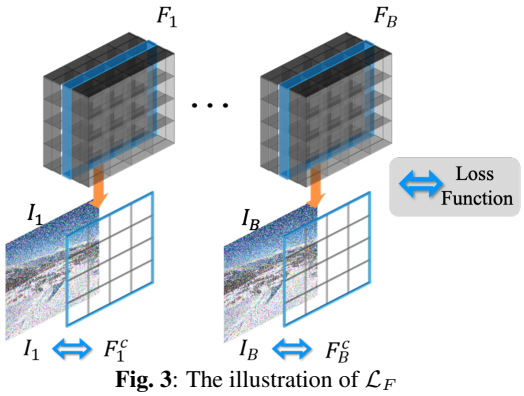

Fig. 3: The illustration of $\mathcal{L}_F$

The second part of the objective function, as depicted in Equation 6, ensures that the feature maps generated by $\mathcal{F}_\theta$ capture the image information. This is achieved by enforcing similarity between the central channels of each feature map and the corresponding noisy frame:

$$\mathcal{L}_\mathcal{F} = \frac{1}{B}\sum_{t=1}^{B}||F_t^c - I_t||, \tag{6}$$

where F_t^c is the central channels of each feature map $\mathcal{F}_\theta(\gamma(G_t))$ and I_t is the corresponding noisy frame, as shown in Fig. 3. The final loss function for the first stage is the sum of these two losses as shown in Equation 7.

$$\mathcal{L}_1 = \mathcal{L}_\mathcal{D} + \lambda_1 \mathcal{L}_\mathcal{F}, \tag{7}$$

where λ_1 is a weight parameter that controls the trade-off between the two terms. To ensure our model doesn't simply learn to reproduce the noise present in the input, we train the network for approximately 2000 epochs.

4.2. Second Stage: Training the Refine-Net $\mathcal{R}_\eta$

In the second stage, we focus on training the Refine-Net $\mathcal{R}_\eta$. Unlike in the first stage, where $\mathcal{F}_\theta$ and $\mathcal{D}_\phi$ were trained together, here we fix $\mathcal{F}_\theta$ and $\mathcal{D}_\phi$ and solely train $\mathcal{R}_\eta$.

We use the coordinates of the central frame G_t^c as input to $\mathcal{R}_\eta$, which outputs a refined denoised image $\hat{I}_R$. As this network is designed to further improve the quality of the denoised frames, the loss function for this stage in Equation 8 is defined to measure the difference between the output of $\mathcal{R}_\eta$ and both the noisy central frame I_t^c and the denoised central frame $\hat{I}_B$ from the first stage.

$$L_2 = \lambda_2||\mathcal{R}\eta(G_t^c) - I^c||+\lambda_3||\mathcal{R}\eta(G_t^c) - \hat{I}_B||, \quad (8)$$

where λ_2 and λ_3 are weight parameters controlling the contribution of each term. These parameters help balance the network's objectives of reducing noise (by making the output similar to I_B^c) and preserving details (by making the output similar to I_t^c).

This strategy allows the Refine-Net to leverage the advantages of both the noisy and denoised frames, by enhancing details and suppressing noise. The training of $\mathcal{R}\eta$ is performed until satisfactory results are obtained, typically for about 2000 epochs. It's important to note that the second stage training does not affect the training of $\mathcal{F}_\theta$ and $\mathcal{D}_\phi$, which is crucial for preserving the generalization capability of the whole framework.

Our two-stage optimization can effectively remove the noise in the video frames. The first image on the left represents the initial state, which is typically a noisy video frame. The joint training of the feature generator $\mathcal{F}_\theta$ and the denoiser $\mathcal{D}_\phi$ shows a noticeable reduction in noise (middle frame), but some blurriness might still be present due to the low L we used in our feature generator. The right image showcases the result of the refining stage $\mathcal{R}_\eta$. At this stage, the high-frequency details that might have been lost in the denoising process are recovered, resulting in a crisp, clean frame that retains the original structure and details of the scene, demonstrating the step-by-step improvement of our method.

5. EXPERIMENTS

In this section, we evaluate the performance of our proposed method through extensive experiments. Our method is tested on a variety of video sequences with different types of noise, and the results are compared with state-of-the-art denoising methods to demonstrate its effectiveness.

5.1. Datasets and Setup

Our approach is tested on various datasets, encompassing both synthetic and real-world scenarios, to provide a comprehensive evaluation of its performance. Synthetic data are derived from established benchmarks and intentionally corrupted with diverse types of noise, while real-world data are sourced from calcium imaging experiments. Moreover, we outline the specifics of our computational setup and training parameters, detailing the choices made to optimize our model's performance. This thorough experimental setup aims to provide a robust assessment of our proposed video denoising method and its potential applicability to different types of video data.
Synthetic. To provide a comprehensive evaluation of our algorithm, we employed a variety of benchmark datasets, using DAVIS [26] and SET8[27] datasets. These datasets include diverse video sequences, each with unique content and characteristics, thus allowing us to test our method under numerous conditions. To evaluate the effectiveness of our model in denoising short frames, denoise short videos, we clipped each video sequence to 10 frames. To challenge our algorithm's robustness, we deliberately introduced various types of noise to the clean video sequences. We evaluated with Gaussian, Poisson and Impulse noise types at different noise intensities.
Real-World. In addition to the synthetic datasets, we also applied our algorithm to real-world, highly noisy calcium imaging data, as shown in Fig. 5. These were locally sourced recordings from freely behaving transgenic mice engaged in cocaine/sucrose self-administration experiments. The recordings were captured using single-channel epifluorescent miniscopes and were subsequently processed using a motion correction algorithm to adjust for translational motion artifacts. This dataset represents the practical complexity and noise levels often present in real-world scenarios, further challenging our algorithm's ability to effectively denoise video sequences.

Our proposed method was implemented using the PyTorch framework and trained on an NVIDIA A100 GPU. We employed the Adam optimizer during the training process, with an initial learning rate of $1e-4$ set for the first stage and $1e-5$ for the second stage. Both learning rates were reduced by a factor of 10 every 100 epochs. In our loss function, we set $\lambda_1 = 0.1$ and $\lambda_2 = 1.0$ as the balancing factors for our dual-term loss.

5.2. Quantitative Evaluation

We employ two widely accepted metrics in image and video processing to facilitate a quantitative evaluation of our approach: the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). These metrics allow us to objectively compare our results with those of the current state-of-the-art unsupervised video denoising algorithm, UDVD [7] and DeepInterpolation [6] and the state-of-the-art supervised network RVRT [5]

The results, presented in Table 1, show that our approach surpasses unsupervised approaches on all evaluated datasets and noise types. While RVRT is effective at removing Gaussian noise, our model demonstrates superior performance across a broader range of noise types and intensities compared to the supervised model, which is mainly optimized for

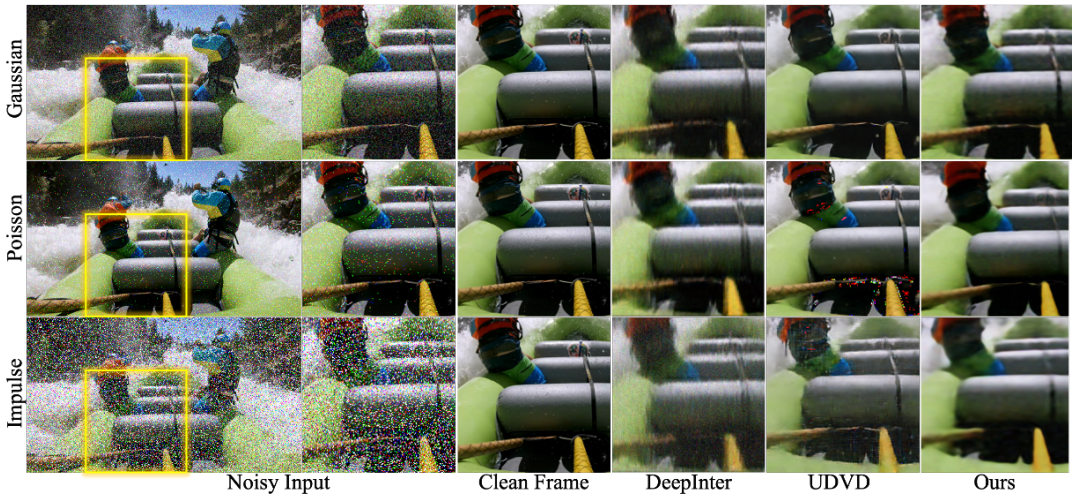

Fig. 4: **Visual comparison of denoising results on the DAVIS dataset:** The figure illustrates denoising results from three types of noise: Gaussian, Poisson, and Impulse.

			Gaussian		Poisson		Impulse	
			$\sigma = 30$	$\sigma = 50$	$\lambda = 30$	$\lambda = 50$	$\alpha = 0.2$	$\alpha = 0.3$
DAVIS	Supervised	RVRT	**30.25**/0.71	**29.24/ 0.77**	27.60/0.78	26.57/ **0.86**	**24.50**/0.61	17.82/0.20
	Unsupervised	DeepInter	24.84/0.56	23.56/0.46	24.41/0.62	23.10/0.56	21.52/0.40	19.95/0.38
		UDVD	27.90/ **0.80**	24.30/0.63	27.70/ **0.82**	24.30/0.77	19.10/0.23	16.90/0.14
		Ours	28.49/0.78	28.86/0.74	**29.81/0.82**	**27.44**/0.79	22.61/ **0.66**	**20.66/0.62**
SET8	Supervised	RVRT	**30.80/ 0.84**	27.13/ **0.78**	29.39/ 0.85	26.77/0.82	22.38/0.54	17.28/0.20
	Unsupervised	DeepInter	21.90/0.50	20.75/0.40	21.04/0.53	20.71/0.50	18.93/0.33	17.36/0.23
		UDVD	27.40/0.79	25.27/0.73	27.84/ **0.87**	26.30/ **0.85**	22.06/0.67	19.02/0.49
		Ours	29.01/0.80	**27.36**/0.76	**29.45**/0.81	**29.05**/0.81	**28.57/ 0.81**	**28.51/0.78**

Table 1: **Denoising results on synthetic noise.** This table presents the PSNR and SSIM values on the DAVIS and SET8 video datasets.

Gaussian noise, and its performance diminishes with different noise types. This superior performance is primarily attributed to our approach's effective utilization of spatial-temporal information from the video sequence. Our algorithm leverages this information to efficiently eliminate noise while simultaneously preserving high-frequency details within the frames.

5.3. Qualitative Evaluation

The visual comparison results presented in Fig. 4 show that UDVD often struggles to effectively remove noise when dealing with short input sequences. This challenge is particularly evident when confronted with Poisson and Impulse noise types, where UDVD tends to produce noticeable artifacts. Also, the performance of DeepInterpolation is notably poorer, especially on Impulse noise. Conversely, our method shows remarkable resilience even in these demanding situations. We tested our approach on several video sequences, each consisting of ten frames, and the results consistently demonstrated our method's superior noise removal capability and robustness.

A visual comparison between our method and UDVD on the highly noisy calcium imaging sequences further underscores our superior performance, as shown in Fig. 5. In the noisy frames, distinguishing individual cells can be challenging due to high noise levels. UDVD, while reducing some noise, often blurs the intricate cellular structures, making it difficult to identify individual cells. DeepInterpolation also introduced some line artifacts to the denoised frames. In contrast, our approach not only removes the noise effectively but also preserves the intricate cellular structures, allowing for better visualization and identification of individual cells. This difference is particularly notable in regions with a high density of cells, where our method is able to maintain the distinct boundaries between cells, whereas UDVD tends to blur them together. This visual comparison highlights our method's ability to handle real-world data with significant noise, offering promising potential for applications in biological and medical imaging.

5.4. Ablation Study

In this section, we conducted ablation studies to assess the impact of various components of our network and to validate the choice of hyperparameters. All experiments were conducted using the DAVIS dataset.

Network Components. We conduct an ablation study to un-

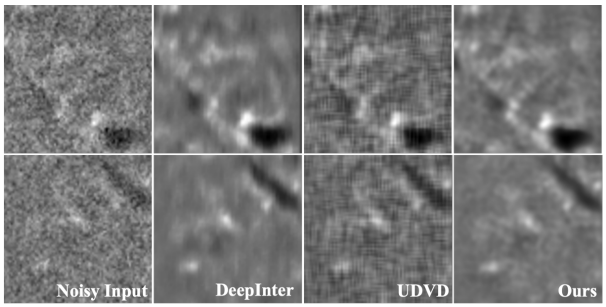

Fig. 5: **Visual comparison on one-photon calcium imaging**: Our method is superior at noise reduction while also maintaining the detailed cellular structures.

derstand the contribution of each component in our method. Notably, when we omit the refining stage $\mathcal{R}_\eta$, the denoised frames tend to be slightly blurry due to the low L in the feature generator $\mathcal{F}_\theta$. However, with the incorporation of the refining stage, our method is able to effectively recover high-frequency details, thereby underscoring its crucial role in enhancing the overall quality of the denoised frames. The result of the ablation study results is provided in Table 2.

Method	PSNR	SSIM
$\mathcal{F}_\theta + \mathcal{D}_\phi$	30.50	0.81
$\mathcal{F}_\theta + \mathcal{D}_\phi + \mathcal{R}_\eta$	**30.90**	**0.84**

Table 2: **Ablation study on network variants.** The table presents the PSNR/SSIM values of each model variant when trained on video corrupted with Gaussian noise of level σ = 30.

Positional encoding. We provide ablation studies on the positional encoding hyperparameters L in Equation 1. We experimented with different frequency levels to encode the input coordinates. Table 3 shows the average PSNR and SSIM value using different L:

	Metrics	$L = 10$	$L = 20$	$L = 30$	$L = 50$
Poisson	PSNR	27.54	27.46	**29.01**	27.46
$\lambda = 30$	SSIM	0.77	0.77	**0.81**	0.77
Gaussian	PSNR	26.18	26.12	**26.74**	25.81
$\sigma = 30$	SSIM	0.72	0.72	**0.75**	0.71

Table 3: **Postional encoding ablation** L means frequency level used to encode the low dimensional coordinate to high dimension. The texts in bold indicate the highest value.

6. DISCUSSION

In this study, we proposed an innovative unsupervised video denoising framework that leverages a two-stage optimization strategy and a novel refine-net that employs the SIREN architecture as its backbone. One key strength of our approach is its adaptability to various noise types and levels without needing prior knowledge about the noise characteristics. However, like all models, ours also has certain limitations. The denoising quality could be affected by the value of L in the feature generator. Setting a lower L might result in blurry output frames, whereas a higher L could potentially lead to overfitting. Therefore, the selection of L requires careful tuning based on the specific characteristics of the dataset.

7. REFERENCES

[1] Matias Tassano, Julie Delon, and Thomas Veit, "Dvdnet: A fast network for deep video denoising," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1805–1809.

[2] Matias Tassano, Julie Delon, and Thomas Veit, "Fastdvdnet: Towards real-time deep video denoising without flow estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1354–1363.

[3] Huanjing Yue, Cong Cao, Lei Liao, Ronghe Chu, and Jingyu Yang, "Supervised raw video denoising with a benchmark dataset on dynamic scenes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2301–2310.

[4] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool, "Vrt: A video restoration transformer," *arXiv preprint arXiv:2201.12288*, 2022.

[5] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc V Gool, "Recurrent video restoration transformer with guided deformable attention," *Advances in Neural Information Processing Systems*, vol. 35, pp. 378–393, 2022.

[6] Jérôme Lecoq, Michael Oliver, Joshua H Siegle, Natalia Orlova, Peter Ledochowitsch, and Christof Koch, "Removing independent noise in systems neuroscience data using deepinterpolation," *Nature methods*, vol. 18, no. 11, pp. 1401–1408, 2021.

[7] Dev Yashpal Sheth, Sreyas Mohan, Joshua L Vincent, Ramon Manzorro, Peter A Crozier, Mitesh M Khapra, Eero P Simoncelli, and Carlos Fernandez-Granda, "Unsupervised deep video denoising," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1759–1768.

[8] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug, "Noise2void-learning denoising from single noisy images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2129–2137.

[9] Joshua Batson and Loic Royer, "Noise2self: Blind denoising by self-supervision," in *International Conference on Machine Learning*. PMLR, 2019, pp. 524–533.

[10] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila, "High-quality self-supervised deep image denoising," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[11] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll, "Implicit functions in feature space for 3d shape reconstruction and completion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6970–6981.

[12] Emilien Dupont, Miguel Bautista Martin, Alex Colburn, Aditya Sankar, Josh Susskind, and Qi Shan, "Equivariant neural rendering," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2761–2770.

[13] Gernot Riegler and Vladlen Koltun, "Free view synthesis," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*. Springer, 2020, pp. 623–640.

[14] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[15] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7462–7473, 2020.

[16] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE transactions on image processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[17] Kai Zhang, Wangmeng Zuo, and Lei Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.

[18] Michele Claus and Jan Van Gemert, "Videnn: Deep blind video denoising," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2019, pp. 0–0.

[19] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila, "Noise2noise: Learning image restoration without clean data," *arXiv preprint arXiv:1803.04189*, 2018.

[20] Thibaud Ehret, Axel Davy, Jean-Michel Morel, Gabriele Facciolo, and Pablo Arias, "Model-blind video denoising via frame-to-frame training," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11369–11378.

[21] Valéry Dewil, Jérémy Anger, Axel Davy, Thibaud Ehret, Gabriele Facciolo, and Pablo Arias, "Self-supervised training for blind multi-frame video denoising," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 2724–2734.

[22] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[23] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler, "Neural geometric level of detail: Real-time rendering with implicit 3d shapes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11358–11367.

[24] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.

[25] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron, "Nerf in the dark: High dynamic range view synthesis from noisy raw images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16190–16199.

[26] Jordi Pont-Tuset, Sergi Caelles, Federico Perazzi, Laura Leal-Taixe, Daniel Cremers, and Luc Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv preprint arXiv:1704.00675*, 2017.

[27] Valéry Dewil, Jérémy Anger, Axel Davy, Thibaud Ehret, Gabriele Facciolo, and Pablo Arias, "Self-supervised training for blind multi-frame video denoising," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 2724–2734.