Unsupervised Deep Video Denoising

Dev Yashpal Sheth[1*], Sreyas Mohan[2*], Joshua L. Vincent[3], Ramon Manzorro[3], Peter A. Crozier[3], Mitesh M. Khapra[1,6], Eero P. Simoncelli[2,4,5], Carlos Fernandez-Granda[2,5]

[1]Indian Institute of Technology Madras, India, [2]Center for Data Science, New York University,
[3]School for Engineering of Matter, Transport, and Energy, ASU,
[4]Center for Neural Science, NYU and Flatiron Institute, Simons Foundation,
[5]Courant Institute of Mathematical Sciences, NYU, [6]Robert Bosch Center for Data Science and AI.

Abstract

Deep convolutional neural networks (CNNs) for video denoising are typically trained with supervision, assuming the availability of clean videos. However, in many applications, such as microscopy, noiseless videos are not available. To address this, we propose an Unsupervised Deep Video Denoiser (UDVD[1]), a CNN architecture designed to be trained exclusively with noisy data. The performance of UDVD is comparable to the supervised state-of-the-art, even when trained only on a single short noisy video. We demonstrate the promise of our approach in real-world imaging applications by denoising raw video, fluorescence-microscopy and electron-microscopy data. In contrast to many current approaches to video denoising, UDVD does not require explicit motion compensation. This is advantageous because motion compensation is computationally expensive, and can be unreliable when the input data are noisy. A gradient-based analysis reveals that UDVD automatically adapts to local motion in the input noisy videos. Thus, the network learns to perform implicit motion compensation, even though it is only trained for denoising.

1. Introduction

Video denoising is a fundamental problem in image processing, as well as an important preprocessing step for computer vision tasks. Convolutional neural networks (CNNs) [24] provide current state-of-the-art solutions for this problem [38, 39, 46, 48, 11, 9, 8, 6]. These networks are typically trained using a database of clean videos, which are corrupted with simulated noise. However, in applications such as microscopy, noiseless ground truth videos are often not available. To address this issue, we propose a method to train a video denoising CNN without access to supervised data, which we call Unsupervised Deep Video Denoising (UDVD). UDVD is inspired by the "blind-spot" technique, recently introduced for unsupervised still image denoising [25, 20, 2, 22], in which a CNN is trained to estimate each *noisy* pixel from the surrounding spatial neighborhood *without including the pixel itself*. Here, we propose a blind-spot architecture that processes the surrounding spatio-temporal neighborhood to denoise videos.

We show that UDVD is competitive with the current supervised state-of-the-art on standard benchmarks, despite not having access to ground-truth clean videos during training (see Figure 1). Moreover, when combined with aggressive data augmentation and early stopping, it can produce high-quality denoising even when trained exclusively on a single *brief* noisy video sequence (as few as 30 frames), outperforming unsupervised video denoising techniques (e.g. F2F[11] and MF2F [9]) which are pre-trained with supervision. Finally, methods based on pre-training are not suitable for imaging applications where clean data is unavailable. In contrast, we demonstrate that UDVD can effectively denoise three different real-world datasets: raw videos from surveillance cameras, fluorescence-microscopy videos of cells, and electron-microscopy videos of catalytic nanoparticles.

The state-of-the-art performance of UDVD is unexpected. Nearly all existing approaches to video denoising [27, 1, 3, 28], including those based on deep CNNs [38, 46, 11, 14, 47], use estimates of optical flow to adaptively compensate for the motion of objects in the video. Conventional wisdom suggest that ignoring such motion should lead to denoising results in which moving content is blurred. Contrary to this intuition, UDVD and some recent state-of-the-art supervised methods for video denoising [39, 8, 6] yield excellent empirical performance without explicit estimation of optical flow. *How can is this achieved?* We use a gradient-based analysis to show that both UDVD and supervised CNNs perform spatio-temporal *adaptive* filter-

*equal contribution.

[1]See https://sreyas-mohan.github.io/udvd/ for code and more results.

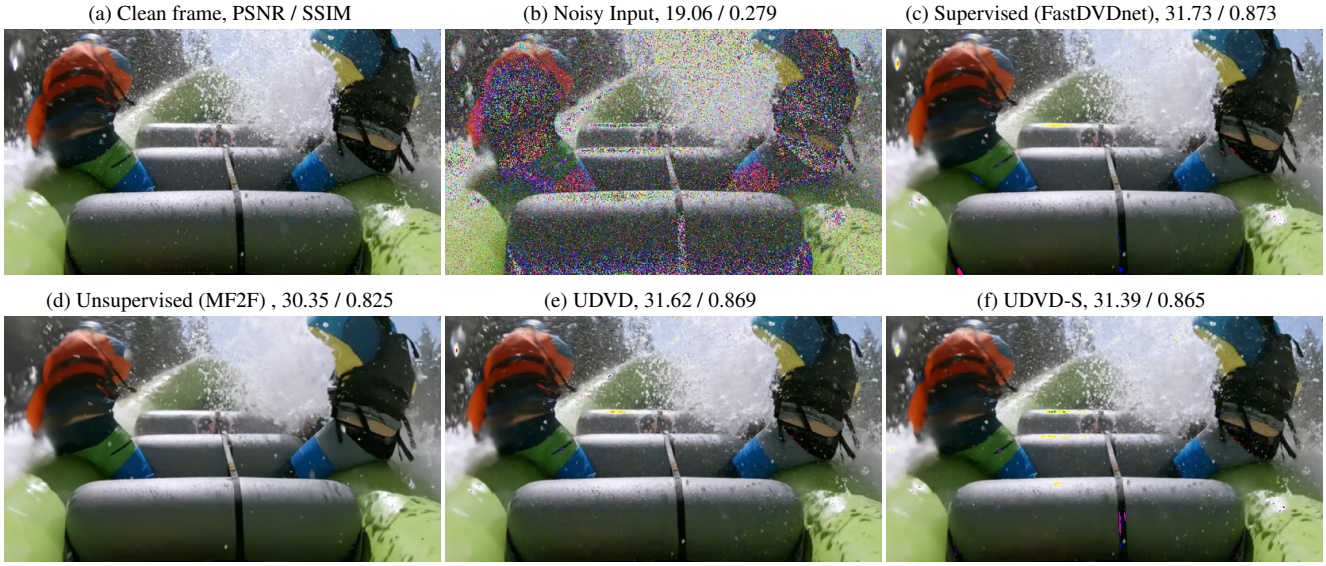


Figure 1. **Unsupervised denoising matches the performance of supervised denoising**. Frame from a video in the Set8 dataset denoised using different approaches. (a) Clean frame. (b) Frame corrupted with Gaussian noise of standard deviation 30 (relative to intensity range [0-255]). (c) FastDVDnet [39], a supervised method trained on the DAVIS dataset. (d) MF2F [9], an unsupervised method which fine-tunes a pre-trained FastDVDnet on the noisy video (e) Our proposed unsupervised method (UDVD), which uses five frames to denoise each frame, trained on the DAVIS dataset. (f) UDVD trained only on the noisy video itself. Performance is quantified using PSNR / SSIM [42], respectively. The corresponding videos, as well as additional examples, are included in Section C of the supplementary material.

ing, which is aligned with underlying motion. Thus, these CNNs are *automatically performing implicit motion compensation*. To quantify this, we demonstrate that it is possible to estimate optical flow accurately from the network gradients, even though the network architectures are not designed to account for optical flow, and the models receive no optical-flow information during training.

Our Contributions:

- A novel blind-spot architecture/objective for unsupervised video denoising, which achieves performance competitive with state-of-the-art supervised methods.
- A training paradigm using aggressive data augmentation (time and space reversal) and early stopping to achieve state-of-the-art performance from training *on a single brief noisy video*.
- A demonstration of our method's effectiveness in denoising real-world electron and fluorescence microscopy data, as well as raw videos. Unlike most existing methods for unsupervised video denoising, our proposed method does not require pre-training, which is key in real-world imaging applications.
- An analysis of the denoising mechanism learned by UDVD, demonstrating that it performs implicit motion compensation even though it is only trained for denoising. We apply the analysis to supervised networks, showing that the same conclusion holds.

2. Background and Related Work

Traditional and CNN-based video denoising. Traditional techniques for single image denoising include nonlinear filtering [40, 29], sparse prior methods [12, 10, 37, 4, 34, 7], and nonlocal means [23]; many of which have been extended to videos [27, 1, 28, 3]. In order to exploit the spatio-temporal structure of the video, these methods typically employ motion compensation based on estimates of optical flow.

In the last five years, data-driven methods based on deep CNNs [24] have outperformed all other techniques in image [50, 15, 5] and video denoising [38, 46, 39, 48]. The CNNs are trained to minimize the mean squared error between the network output and ground truth using large databases of natural images/videos. Many deep-learning techniques also perform explicit motion compensation. DVDnet [38] applies an image-denoising CNN to each input frame, estimates the optical flow from the denoised frames using DeepFlow [43] (a CNN pre-trained for this purpose), warps the frames using the flow estimate to align their content, and finally processes the registered frames with a CNN. Ref. [46] applies a similar pipeline, but jointly trains an optical-flow module with the denoising CNN.

Video denoising without motion compensation. Three recent methods perform video denoising without explicit motion estimation. VNLnet [8] uses a non-local search algorithm to find self-similar patches in the input video, and then uses a CNN to process the patches. ViDeNN [6] consists of

a first stage that denoises each frame using a CNN, and a second stage that exploits temporal structure by using the frames, $(t-1)$, t and $t+1$ to produce the denoised tth frame. FastDVDnet [39] uses UNet [36] blocks, trained end to end, to denoise each frame using five contiguous frames. These methods achieve state-of-the-art performance without any explicit motion compensation, similar to our proposed UDVD. In this work we show that such CNNs actually performs *implicit* motion estimation, which can be revealed through a gradient-based analysis.

Unsupervised denoising. Noise2Noise (N2N) is an unsupervised image-denoising technique where a CNN is trained on pairs of noisy images corresponding to the same clean image [25]. Frame2Frame (F2F) [11] exploits this approach to fine-tune a pretrained image-denoising CNN with noisy data. The idea is to register contiguous frames using the optical flow (obtained from TV-L1 [49]), and treat them as noisy realizations of the same clean image. This scheme is extended to have a trainable flow estimation module in [47], additional optical-flow consistency in [14] and to use multiple noisy frames as input in Multi-Frame2Frame (MF2F) [9].

Using the N2N framework to perform unsupervised video denoising requires warping adjoining frames, which in turn requires explicit motion compensation, and accurate occlusion estimation. In addition, the assumption that contiguous frames can be registered may not hold, particularly if the motion speeds in the video are large relative to the frame rate or local intensity changes are not due to translation. In order to bypass these issues, we develop a blind-spot network that trains denoising CNNs by fitting the noisy data directly. The CNN is trained to estimate each noisy pixel value using the surrounding spatio-temporal neighborhood, but without taking into account the noisy pixel itself in order to avoid the trivial identity solution. This "blind spot" can be enforced through architecture design [22], or by masking [2, 20]. For still images, several variations of this approach have been shown to provide effective denoising for natural images and noisy images from fluorescence microscopy [21, 35, 17].

3. Unsupervised Deep Video Denoising

In this section we describe our proposed architecture (see Figure 2 for a detailed diagram).

Multi-frame blind-spot architecture. Our CNN maps five contiguous noisy frames to a denoised estimate of the middle frame. Building on the "blind spot" idea proposed in [22] for single-image denoising, we design the architecture so that each output pixel is estimated from a spatio-temporal neighbourhood that does not include the pixel itself. We rotate the input frames by multiples of 90° and process them through four separate branches containing asymmetric convolutional filters that are *vertically causal*. As a result, the branches produce outputs that only depend on the pixels above (0° rotation), to the left (90°), below (180°) or to the right (270°) of the output pixel. These partial outputs are then *derotated* and combined using a three-layered cascade of 1×1 convolutions and nonlinearities to produce the final output. The resulting field of view does not include the pixel being denoised, as depicted at the bottom of Figure 2.

UDVD processes the video in two stages as shown in Figure 2, similar to previously proposed networks for supervised video denoising [38, 6, 39]. A first stage, consisting of three UNets [36] (D1 in the diagram) with shared parameters, maps each group of three contiguous frames (i.e. $(t-2, t-1, t)$, $(t-1, t, t+1)$ and $(t, t+1, t+2)$) to a separate feature map. These features are then mapped to a single output using another UNet (D2). See Suppl. A for a detailed description of the architecture.

Bias-free architecture. Inspired by [31], we remove all additive terms from the convolutional layers in UDVD. This provides automatic generalization to varying noise levels not encountered during training, and facilitates our proposed analysis to interpret the denoising mechanisms learned by the network (see Section 5 and 6).

Using the missing pixel. The denoised value generated by the proposed architecture at each pixel is computed without using the noisy observation at that location. This avoids overfitting – i.e. learning the trivial identity map that minimizes the mean-squared error cost function – but ignores important information provided by the noisy pixel. In the special case of Gaussian additive noise, we can use this information via a precision-weighted average between the network output and the noisy pixel value. Following [22, 21], the weights in the average are derived by assuming a Gaussian distribution for the error in the blind-spot estimates of the color pixel values. Specifically, we model the distribution of the three color channels of a pixel $x \in \mathcal{R}^3$ given the noisy neighbourhood Ω_y as $p(x|\Omega_y) = \mathcal{N}(\mu_x, \Sigma_x)$, where $\mu_x \in \mathcal{R}^3$ and $\Sigma_x \in \mathcal{R}^3$ represent the mean vector and covariance matrix. Let $y = x + \eta$, $\eta \sim \mathcal{N}(0, \sigma^2 I_3)$ be the observed noisy pixel. We integrate the information in the noisy pixel with the UDVD output by computing the mean of the posterior $p(x|y, \Omega_y)$, given by

$$E[x|y] = (\Sigma_x^{-1} + \sigma^{-2} I)^{-1} (\Sigma_x^{-1} \mu_x + \sigma^{-2} y). \quad (1)$$

See Suppl. A for more details. The CNN architecture is trained to estimate the mean and covariance of this distribution at each pixel by maximizing the log likelihood of the noisy data:

$$\mathcal{L}(\mu_x, \Sigma_x) = \frac{1}{2}[(y-\mu_x)^T (\Sigma_x + \sigma^2 I)^{-1} (y - \mu_x)] + \frac{1}{2} \log |\Sigma_x + \sigma^2 I|. \quad (2)$$

When the noise process is unknown, we simply minimize

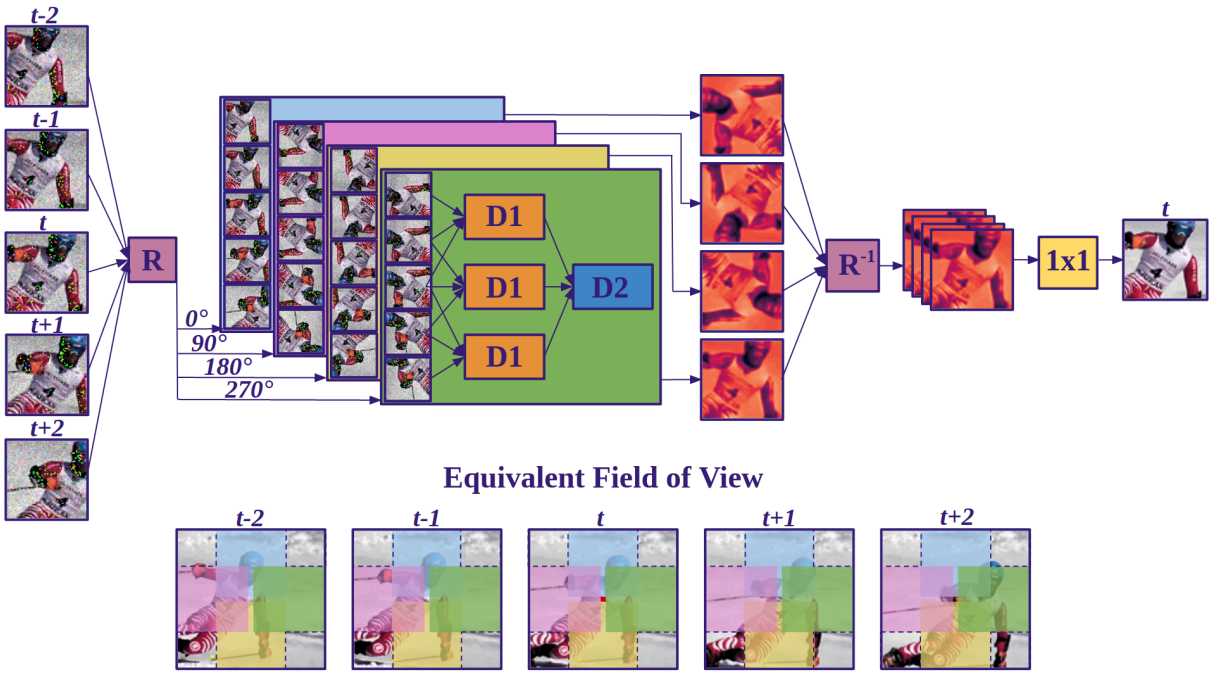


Figure 2. **Unsupervised Deep Video Denoising (UDVD) Network Architecture**. The network takes 5 consecutive noisy frames as input and produces a denoised central frame as output. We rotate the input frames by multiples of 90° and process them in four separate branches with shared parameters, each containing asymmetric convolutional filters that are *vertically causal*. As a result, the branches produce outputs that only depend on the pixels above (0° rotation, blue region), to the left (90°, pink region), below (180°, yellow region) or to the right (270°, green region) of the output pixel. Each branch consists of a cascade of 2 Unet-style blocks (D1 and D2) to combine information over frames. These outputs are then *derotated* and linearly combined (using a 1×1 convolutions) followed by a ReLU nonlinearity to produce the final output. The resulting "field of view" is depicted at the bottom with each color representing the contribution of the corresponding branch.

the MSE between the denoised output and noisy video, and ignore the center pixel (see Suppl. A for more details).

Data augmentation and early stopping. In supervised denoising with simulated noise, training can rely on the generation of a virtually unlimited set of fresh noise realizations, which prevents overfitting. In the unsupervised setting, this is not possible, which makes it more challenging to train models that can denoise short video sequences. To address this, we (a) leverage data augmentation strategies: spatial flipping and time reversal, and (b) perform early stopping by monitoring the mean squared error between the network output and noisy frames on a held-out set of frames. These strategies make it possible to train UDVD with short video sequences (as few as 30 frames), while achieving denoising performance that is on par with or superior to both unsupervised and supervised networks trained on much larger datasets (see Figure 1, Table 2 and Suppl. D).

4. Datasets

We demonstrate the broad applicability of our approach by validating it on domains with different signal and noise structure: natural videos, raw videos, fluorescence microscopy, and electron microscopy.

Natural videos. We perform controlled experiments on natural videos by adding iid Gaussian noise to the DAVIS dataset [33]. The training/validation/test split is 60/30/30 videos, respectively. We use three additional datasets for testing - Set8 [39] composed of 4 videos from the Derfs Test Media collection and 4 videos captured with a GoPro camera, Derfs [9] with 7 videos, and the first 10 videos from Vid3oC [18] dataset (See Suppl. D for details).

Raw videos. We evaluate UDVD on a dataset of raw videos i.e with frame color channels interleaved according to the sensor mosaic containing real noise introduced in [48]. The dataset contains 11 unique videos, each containing 7 frames, captured at five different ISO levels using a surveillance camera. Each video has 10 different noise realizations per frame, which are averaged to obtain an estimated clean version of the video.

Fluorescence microscopy. We apply our approach to fluorescence-microscopy recordings of live cells in [41]. We use two videos: Fluo-C2DL-MSC (CTC-MSC) depicting mesenchymal stem cells, and Fluo-N2DH-GOWT1

		Traditional		Supervised CNN			Unsupervised CNN (UDVD)		
test set	σ	VNLB	VBM4D	VNLnet	DVDnet	FastDVDnet	1 frame	3 frames	5 frames
DAVIS	30	33.73	31.65	-	**34.08**	34.06	32.80	33.48	**33.92**
	40	32.32	30.05	32.32	**32.86**	32.80	31.48	32.20	**32.68**
	50	31.13	28.80	31.43	**31.85**	31.83	30.47	31.20	**31.70**
Set8	30	31.74	30.00	-	**31.79**	31.60	30.91	31.62	**32.01**
	40	30.39	28.48	**30.55**	**30.55**	30.37	29.63	30.42	**30.82**
	50	29.24	27.33	29.47	**29.56**	29.42	28.65	29.47	**29.89**

Table 1. **Denoising results on natural video datasets**. All networks are trained on the DAVIS train set. Performance values are PSNR of each trained network averaged over held-out test data. UDVD, operating on 5 frames, outperforms the supervised methods on Set8 and is competitive on the DAVIS test set. Unsupervised denoisers with more temporal frames show a consistent improvement in denoising performance. DVDnet and FastDVDnet are trained using varying noise levels ($\sigma \in [0, 55]$) and VNLnet is trained and evaluated on each specified noise level. All UDVD networks are trained *only* at $\sigma = 30$, showing that they generalize well on unseen noise levels. See Sections C and F in the supplementary material for additional results. The PSNR values for all methods except UDVD are taken from [39].

	$\sigma = 30$				$\sigma = 90$			
	DAVIS	Set8	Derfs	Vid3oC	DAVIS	Set8	Derfs	Vid3oC
UDVD-S	33.68 / 78.16	**32.90** / 81.85	**33.95** / 81.91	34.65 / 84.60	**29.05 / 53.53**	**28.07 / 55.35**	**29.42 / 59.25**	**29.94 / 63.79**
UDVD*	33.78 / 79.88	31.90 / **82.53**	32.58 / 81.44	34.24 / 83.96	28.87 / 51.22	27.25 / 51.84	28.26 / 52.44	29.23 / 60.08
FastDVDnet*	**33.91** / 76.99	31.81 / 80.21	32.45 / 81.64	35.05 / 84.44	28.01 / 47.53	26.54 / 50.16	27.36 / 52.87	28.42 / 55.99
MF2F	**33.91 / 80.01**	31.84 / 80.55	32.87 / **82.22**	**35.18 / 85.71**	28.81 / 51.24	27.25 / 52.78	28.29 / 55.06	29.67 / 61.28

Table 2. **Results for UDVD trained on individual noisy videos**. The top row shows **PSNR/VMAF**[26] values (averaged over the entire dataset) for UDVD trained on each individual video sequence with early stopping (labelled UDVD-S) using the last 5 frames of a video as a held-out set. We augmented the dataset with spatial flipping and time reversal (see Suppl. D for an ablation study). With the augmentations and early stopping, UDVD-S is comparable to (and often outperforms) UDVD or FastDVDnet trained on the full DAVIS dataset (indicated by *) and MF2F, which fine-tunes a pre-trained CNN on each individual video. See Suppl. D for results on individual video sequences.

(CTC-N2DH) depicting GOWT1 cells. This dataset illustrates the challenges of applying supervised approaches to real data: there is no ground-truth clean data.

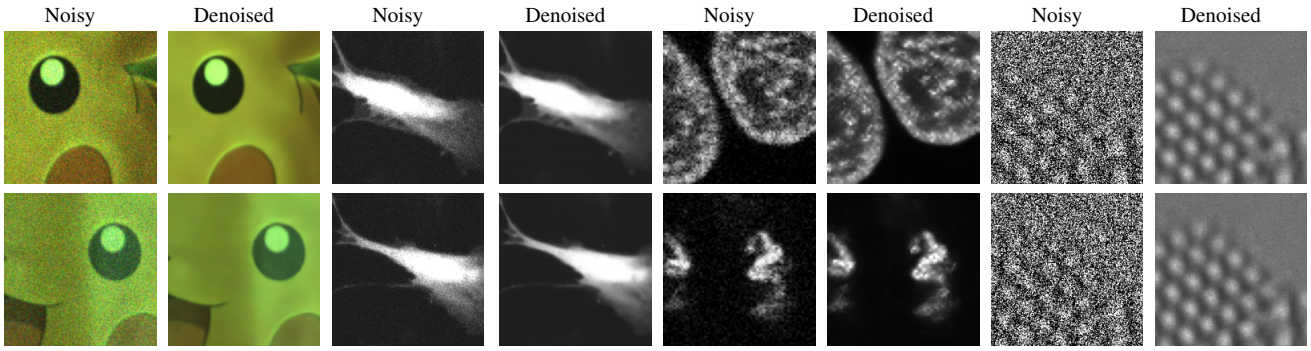
Electron microscopy. We also apply our methodology to a transmission electron microscopy dataset from [32]. The data consist of a 40-frame video depicting a platinum nanoparticle supported on a cerium oxide base. The average image intensity is 0.45 electrons/pixel, which results in an extremely low signal-to-noise ratio. As with the fluorescence-microscopy data, no ground-truth clean images are available.

5. Experiments and Results

Comparison with other approaches on natural videos. We train UDVD on the DAVIS training set (see Suppl. A for the training procedure). Following [50, 31, 39, 38, 22, 20, 2], we add iid Gaussian noise with standard deviation $\sigma = 30$ on the clean videos during training. UDVD is evaluated on the DAVIS test set and on Set8 by comparing to the clean ground-truth videos via PSNR. We compare UDVD with several popular methods: Bayesian processing of spatio-temporal patches (VNLB [23]), an extension of the popular image-denoising algorithm BM3D (VBM4D [28]) and supervised CNNs (VNLnet [8], DVDnet [38], FastDVDnet [39]). As shown in Table 1, UDVD achieves comparable performance to the supervised state-of-the-art on the DAVIS test set and slightly outperforms these methods on an independent test set (Set8) at multiple noise levels. It also outperforms traditional unsupervised techniques such as VNLB and VBM4D (see Figure 1 and Suppl. C for visual examples).

Unsupervised denoising from limited data. In order to validate our approach on a more challenging setting that is closer to the practical applications of unsupervised denoising, we trained and tested UDVD on individual videos from our test sets. As shown in Table 3 and 4 in Suppl. D, when combined with data augmentation and early stopping (using the last 5 frames of each video as a held-out validation set), this version of UDVD (called UDVD-S) achieves comparable results, or often outperforms supervised FastDVDnet and unsupervised UDVD trained on a large dataset (DAVIS) (see Table 2 for results on 4 different datasets).

To the best of our knowledge, all the existing unsupervised video denoising techniques are based on the F2F [11] framework, where a backbone CNN pre-trained with supervision is fine-tuned on the video to be denoised. We compared UDVD-S against the most recent such method – MF2F [9] which fine-tunes a FastDVDnet [39] trained



(a) Raw video (scene 11) (b) Fluor. micro. (Mes. stem cells) (c) Fluor. micro. (GOWT1 cells) (d) Electron micro. (nanoparticle)

Figure 3. **Denoising real-world data.** Results from applying UDVD to the raw video, fluorescence-microscopy and electron-microscopy datasets described in Section 4. Qualitatively, UDVD succeeds in removing noise while preserving the underlying signal structure, even for the highly noisy electron-microscopy data. Raw videos are converted to RGB for visualization. See Suppl. D and F for denoised videos.

with supervision on natural videos using an objective involving optical flow computed on consecutive noisy frames (see Section 2). Without any pre-training, UDVD-S outperforms MF2F in almost all videos in Table 3 and 4 in Suppl. D, and datasets in Table 2 (See Table 5 in Suppl. D.3 for measure of confidence). Note that (a) we trained MF2F using all the 5 training schemes provided in the paper and reported the best results in Table 2, and (b) the metric we used to measure performance in Table 2 is the average PSNR of all denoised frames, unlike in Ref. [9] where the first 10 frames of each video were excluded (see Suppl. D.3 for more details and results).

Use of temporal information. UDVD estimates each frame from k surrounding contiguous frames. To validate the effect of using more temporal information, we tested $k \in \{1, 3, 5\}$. As shown in Table 1, performance improves substantially and monotonically with k (see Suppl. B for more noise levels). This is in agreement with the literature on supervised learning [39]. The performance gains arising from a longer temporal context are more substantial at higher noise levels (see Table 1). This is consistent with our analysis in Section 6 which shows that, at low noise levels, UDVD($k = 5$) tends to ignore the distant frames, but relies on them more at higher noise levels (see Figure 4 & Suppl. G).

Generalization across noise levels. UDVD generalizes strongly across noise levels not encountered during training. The results in Table 1 are obtained with a network trained only at a fixed noise level of $\sigma = 30$. This generalization ability is consistent with bias-free networks for image denoising [31]. See Suppl. F for more discussion and results.

Raw videos with real noise. We train UDVD on the first 9 realizations of the 5 videos from the test set of the raw video dataset (see Section 4), holding out the last realization for early stopping. We compare our performance with RViDeNet [48] which is pre-trained on a simulated dataset and then fine-tuned with supervision on 6 training videos from the raw video dataset. UDVD outperforms RViDeNet at all noise levels (see Table 3 and Fig 3). Note that UDVD was directly trained on the mosaiced raw videos. Existing unsupervised video denoising methods, like MF2F, cannot be applied directly on this dataset as their pre-trained backbone expects an input in the RGB domain (more details in Suppl. E).

CNN \ ISO	1600	3200	6400	12800	25600	**mean**
UDVD	**48.04**	**46.24**	**44.70**	**42.19**	**42.11**	**44.69**
RViDeNet [48]	47.74	45.91	43.85	41.20	41.17	43.97

Table 3. **Raw video denoising**. PSNR values evaluated on the test set of the raw video dataset (Section 4) when denoised with (a) UDVD trained only the noisy test videos and (b) RViDeNet trained with supervision on a large dataset. The columns correspond to different ISO levels, with larger levels resulting in noisier data.

Real-world microscopy data. We train UDVD on the fluorescence-microscopy data described in Section 4 following the same procedure as for the natural videos, including data augmentation. For the electron-microscopy data, we trained on the first 35 frames of the video, and used the remaining 5 as a validation set to perform early stopping based on mean-squared error. UDVD is able to effectively denoise the fluorescence-microscopy and the electron-microscopy datasets described in Section 4. This can be appreciated qualitatively in Figure 3 and Suppl. E.

6. Automatic Motion Compensation

Most previous approaches for video denoising rely on explicit motion compensation [27, 1, 3, 28]. This requires estimating the optical flow, which is the local translational motion of features in the image arising from the motion of objects and surfaces in a visual scene relative to the cam-

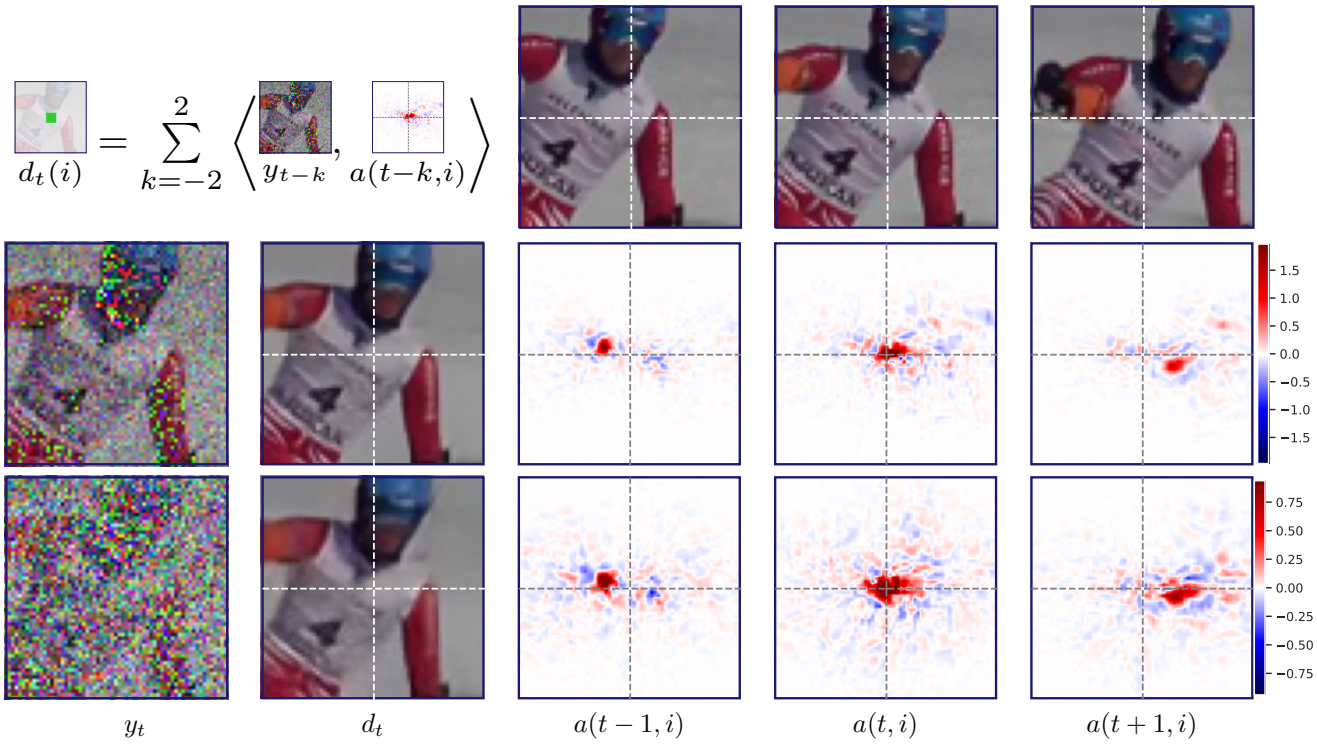


Figure 4. **Video denoising as spatiotemporal adaptive filtering**. Visualization of the equivalent linear weights ($a(k,i)$, Eq. 4) used to compute two example denoised pixels using UDVD. The left two columns show noisy frames y_t at two noise levels, and the corresponding denoised frames, d_t. Three successive clean frames $\{x_{t-1}, x_t, x_{t+1}\}$ are shown in top row, for reference. Corresponding weights $a(k,i)$ for pixel i (intersection of the dashed white lines) in these three frames, are shown in the last three columns. The weights are seen to adapt to underlying video content, with their mode shifting to track the motion of the skier. As the noise level σ increases (bottom row), their spatial extent grows, averaging out more of the noise while respecting object boundaries. For each denoised pixel, the sum of weights (over all pixel locations and frames) is approximately one, and thus can be interpreted as computing a local average (but note that some weights are negative, depicted in blue).

era. Several CNN-based denoisers build motion estimation into the architecture [38, 46]. In particular, motion compensation is critical to the F2F and MF2F frameworks for unsupervised denoising, which use motion compensation to register contiguous images [11, 14, 9]. In contrast, recent supervised video denoising networks like FastDVDnet [39] and ViDeNN [6], as well as our unsupervised UDVD, do not perform any explicit motion compensation. Despite this, they achieve state-of-the-art results. The empirical performance of these approaches suggests that the networks must somehow be exploiting temporal information successfully. Here, we study this phenomenon through an analysis of the denoising mapping, which reveals that these networks perform an implicit form of motion compensation.

Gradient-based analysis. We use the approach of [31] to analyze CNNs trained for image denoising. Let $y \in \mathbb{R}^{nT}$ be a flattened video sequence containing T noisy frames with n pixels each, processed by a CNN. We define the denoising function $f_i : \mathbb{R}^{nT} \to \mathbb{R}$ as the map between the noisy video and the denoised value $d_i := f_i(y)$ of the CNN output at the ith pixel. A first-order Taylor decomposition of the denoising function may be written as:

$$d_i := f_i(y) = \langle \nabla f_i(y), y \rangle + b, \tag{3}$$

where $\nabla f_i(y) \in \mathbb{R}^{nT}$ denotes the gradient of f_i at y. The constant $b := f_i(y) - \langle \nabla f_i(y), y \rangle$ is the net bias of the network, a combined function of all additive constants in the convolutional and batch-normalization layers of the CNN.

Our proposed architecture is bias-free (i.e., all additive constants are removed from the architecture, as proposed in [31]), and thus $b = 0$. As a result, the denoised value at the ith pixel may be written as:

$$d(i) = \langle \nabla f_i(y), y \rangle = \sum_{k=1}^{T} \langle a(k,i), y_k \rangle, \tag{4}$$

where y_k denotes each of the T flattened frames that compose the noisy video, and the weights $a(k,i)$ correspond to the gradient of f_i with respect to y. Each vector $a(k,i)$ can be interpreted as an *equivalent filter* that produces an estimate of the denoised video at pixel i via a weighted average of the noisy observations over space and time.

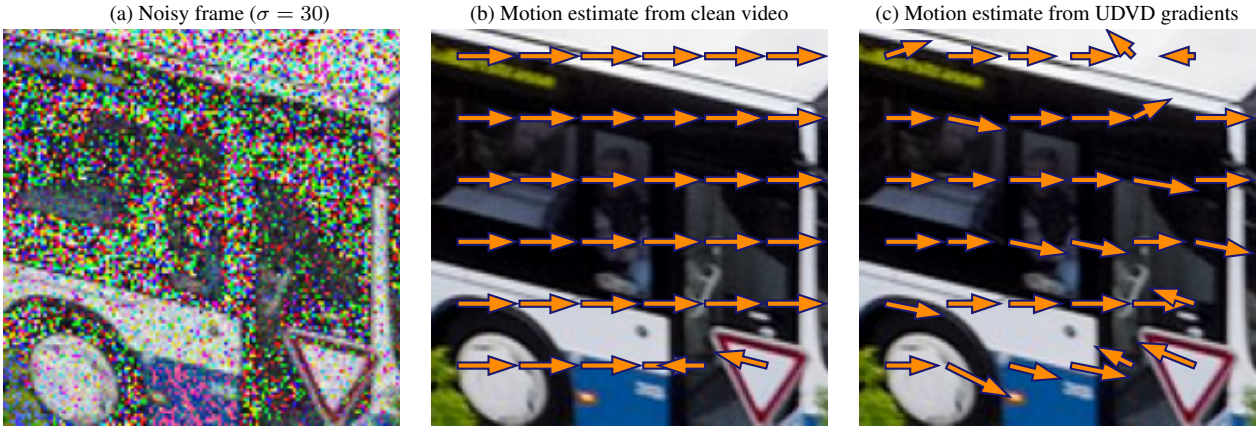


Figure 5. **CNNs trained for denoising automatically learn to perform motion estimation**. (a) Noisy frame from a video in the DAVIS dataset. (b) Optical flow direction at multiple locations of the image obtained using a state-of-the-art algorithm applied *to the clean video*. (c) Optical flow direction estimated from the shift of the adaptive filter obtained by differentiating the network, which is trained exclusively with noisy videos and no optical flow information. Optical flow estimates are well-matched to those in (b), but deviate according to the aperture problem at oriented features (see black vertical edge of bus door), and in homogeneous regions (see bus roof, top right).

Interpreting equivalent filters. Visualizing these equivalent filters reveals that UDVD learns to denoise by performing averaging over an adaptive spatiotemporal neighborhood of each pixel. As illustrated in Figure 4 (and Suppl. G), when the noise level increases, the averaging is carried out over larger regions. This intuitive behavior is also seen in classical linear Wiener filters [44], where the filters are larger for higher levels of noise. The crucial difference is that in the case of CNNs, the equivalent filters are *adapted* to the local video content: they respect object boundaries in space and time, taking into account their motion. This is apparent in Figure 4: equivalent filters in adjoining frames are automatically shifted spatially to compensate for the movement of the skier (additional examples in Suppl. G). We find that this implicit motion compensation is not unique to UDVD: CNNs trained in a supervised fashion have the same property (see also Suppl. G).

Optical-flow estimation. In order to validate our observation that CNNs exclusively trained for denoising implicitly detect and exploit video motion, we use the equivalent filters of the networks to estimate the optical flow. To estimate the optical flow from the t^{th} frame to the $(t+1)^{th}$ frame at the ith pixel, we compute the difference between the position of the centroid of the equivalent filter corresponding to the pixel at times t, $a(t,i)$, and time $t+1$, $a(t+1,i)$. To increase the stability of the estimated flow, we compute the filter centroid through a robust weighted average that only includes entries with relatively large values (within 20% of maximum value in the filter).

The optical-flow estimates obtained from the gradients of the trained UDVD network are surprisingly precise, even at very high noise levels. Figure 5, and additional figures in Suppl. G, show that the results are similar to those obtained by applying an algorithm for optical-flow estimation (DeepFlow [43]) on the corresponding *clean* video. This demonstrates that the CNNs are able to implicitly estimate motion from data, despite the fact that they were not trained on that problem, and *even in the presence of substantial noise corruption*, a setting that is quite challenging for optical-flow estimation techniques. We also observe that the optical-flow estimates obtained from UDVD gradients tend to be less accurate for pixels near strongly oriented features where local motion is only partially constrained (known as the *aperture problem*) or in homogeneous regions, where the local motion is unconstrained (the *blank wall problem*).

7. Conclusion

In this work we propose a method for unsupervised deep video denoising that achieves comparable performance to state-of-the-art supervised approaches. Combined with data-augmentation techniques and early stopping, the method achieves effective denoising even when trained exclusively on short individual noisy sequences, which enables its application to real-world noisy data. In addition, we perform a gradient-based analysis of denoising CNNs, which reveals that they learn to perform implicit adaptive motion compensation. This suggests several interesting research directions. For example, denoising may be a useful pretraining task for optical-flow estimation or other computer-vision tasks requiring motion estimation.

Acknowledgements. This work was supported by HHMI, NSF NRT HDR Award 1922658, CBET 1604971, OAC-1940263 and OAC-1940097. We thank the HPC staff at NYU, ASU and RBCDSAI, IIT Madras for their support.

References

[1] Pablo Arias and Jean-Michel Morel. Video denoising via empirical Bayesian estimation of space-time patches. *Journal of Mathematical Imaging and Vision*, 60(1):70–93, 2018. 1, 2, 6

[2] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *Proceedings of the 36th International Conference on Machine Learning*, pages 524–533, 2019. 1, 3, 5

[3] Antoni Buades, Jose-Luis Lisani, and Marko Miladinović. Patch-based video denoising with optical flow estimation. *IEEE Transactions on Image Processing*, 25(6):2573–2586, 2016. 1, 2, 6

[4] S Grace Chang, Bin Yu, and Martin Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Processing*, 9(9):1532–1546, 2000. 2

[5] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1256–1272, 2016. 2

[6] Michele Claus and Jan van Gemert. Videnn: Deep blind video denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, pages 1843–1852, 2019. 1, 2, 3, 7

[7] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, pages 2080–2095, 2017. 2

[8] Axel Davy, Thibaud Ehret, Jean-Michel Morel, Pablo Arias, and Gabriele Facciolo. A non-local CNN for video denoising. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2409–2413. IEEE, 2019. 1, 2, 5

[9] Valery Dewil, Jeremy 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*, pages 2724–2734, 2021. 1, 2, 3, 4, 5, 6, 7, 14, 15, 16, 17

[10] D Donoho. Denoising by soft-thresholding. *IEEE Trans Info Theory*, 43:613–627, 1995. 2

[11] 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*, pages 11361–11370, 2019. 1, 3, 5, 7

[12] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006. 2

[13] Clement Godard, Kevin Matzen, and Matt Uyttendaele. Deep burst denoising. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 13

[14] Bichuan Guo, Jiangtao Wen, Zhen Xia, Shan Liu, and Yuxing Han. Learning model-blind temporal denoisers without ground truths. *arXiv preprint arXiv:2007.03241*, 2020. 1, 3, 7

[15] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1712–1722, 2019. 2

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 11, 17

[17] Wesley Khademi, Sonia Rao, Clare Minnerath, Guy Hagen, and Jonathan Ventura. Self-supervised Poisson-Gaussian denoising. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2131–2139, 2021. 3

[18] Sohyeong Kim, Guanju Li, Dario Fuoli, Martin Danelljan, Zhiwu Huang, Shuhang Gu, and Radu Timofte. The vid3oc and intvid datasets for video super resolution and quality mapping. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3609–3616, 2019. 4, 15

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 12

[20] 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*, pages 2124–2132, 2019. 1, 3, 5

[21] Alexander Krull, Tomas Vicar, and Florian Jug. Probabilistic noise2void: Unsupervised content-aware denoising. *arXiv preprint arXiv:1906.00651*, 2019. 3

[22] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *Advances in Neural Information Processing Systems 32*, pages 6970–6980, 2019. 1, 3, 5, 11, 12, 13

[23] Marc Lebrun, Antoni Buades, and Jean-Michel Morel. A nonlocal Bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013. 2, 5

[24] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015. 1, 2

[25] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2965–2974, 2018. 1, 3

[26] Zhi Li, Anne Aaron, Ioannis Katsavounidis, Anush Moorthy, and Megha Manohara. Toward a practical perceptual video quality metric. *Netflix Technology Blog*, 2016. 5

[27] Ce Liu and William T Freeman. A high-quality video denoising algorithm based on reliable motion estimation. In *European conference on computer vision*, pages 706–719. Springer, 2010. 1, 2, 6

[28] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian. Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. *IEEE Transactions on image processing*, 21(9):3952–3966, 2012. 1, 2, 5, 6

[29] Peyman Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE signal processing magazine*, 30(1):106–128, 2012. 2

[30] Ben Mildenhall, Jonathan T. Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 13, 14

[31] Sreyas Mohan, Zahra Kadkhodaie, Eero P. Simoncelli, and Carlos Fernandez-Granda. Robust and interpretable blind image denoising via bias-free convolutional neural networks. In *Proceedings of the International Conference on Learning Representations*, 2020. 3, 5, 6, 7, 12, 17

[32] Sreyas Mohan, Ramon Manzorro, Joshua L Vincent, Binh Tang, Dev Yashpal Sheth, Eero P Simoncelli, David S Matteson, Peter A Crozier, and Carlos Fernandez-Granda. Deep denoising for scientific discovery: A case study in electron microscopy. *arXiv preprint arXiv:2010.12970*, 2020. 5

[33] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 4, 14, 16

[34] Javier Portilla, Vasily Strela, Martin J Wainwright, and Eero P Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. Image Processing*, 12(11), 2003. 2

[35] Mangal Prakash, Manan Lalit, Pavel Tomancak, Alexander Krull, and Florian Jug. Fully unsupervised probabilistic noise2void. *arXiv preprint arXiv:1911.12291*, 2019. 3

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3, 11

[37] E P Simoncelli and E H Adelson. Noise removal via Bayesian wavelet coring. In *Proc 3rd IEEE Int'l Conf on Image Proc*, volume I, pages 379–382, Lausanne, Sep 16-19 1996. IEEE Sig Proc Society. 2

[38] Matias Tassano, Julie Delon, and Thomas Veit. Dvdnet: A fast network for deep video denoising. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1805–1809, 2020. 1, 2, 3, 5, 7, 12, 17

[39] 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*, pages 1351–1360, 2020. 1, 2, 3, 4, 5, 6, 7, 12, 13, 15, 16, 17

[40] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *ICCV*, volume 98, 1998. 2

[41] Vladimír Ulman et al. An objective comparison of cell-tracking algorithms. *Nature Methods*, 14:1141–1152, 2017. 4

[42] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Processing*, 13(4):600–612, 2004. 2

[43] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE international conference on computer vision*, pages 1385–1392, 2013. 2, 8, 18

[44] Norbert Wiener. *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. Technology Press, 1950. 8

[45] Zhihao Xia, Federico Perazzi, Michael Gharbi, Kalyan Sunkavalli, and Ayan Chakrabarti. Basis prediction networks for effective burst denoising with large kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 13

[46] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019. 1, 2, 7

[47] Songhyun Yu, Bumjun Park, Junwoo Park, and Jechang Jeong. Joint learning of blind video denoising and optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 1, 3

[48] H. Yue, C. Cao, L. Liao, R. Chu, and J. Yang. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2298–2307, 2020. 1, 2, 4, 6, 17

[49] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007. 3

[50] 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*, pages 3142–3155, 2017. 2, 5, 12, 17

A. Implementation Details of Unsupervised Deep Video Denoising

A.1. Restricting field of view

In UDVD, we rotate the input frames by multiples of 90° and process them through four separate branches (with shared parameters) containing asymmetric convolutional filters that are *vertically causal*. As a result, the branches produce outputs that only depend on the pixels above (0° rotation), to the left (90°), below (180°) or to the right (270°) of the output pixel. We use a UNet [36] style architecture for each branch of UDVD. The field of view of the UNet is constrained by restricting the field of view of the convolutional, downsampling and upsampling layers that are used to build the UNet.

Convolutional Layers: We restrict the receptive field of each convolutional layer to extend only upwards following the strategy proposed in [22]. Let the filter size be $h \times w$. We zero-pad the top region of the input tensor with $k = \lfloor h/2 \rfloor$ zero rows before convolution and remove the bottom k rows after convolution. This is equivalent to convolving with a filter, where all weights below the center row are zero, so that the field of view only extends upwards.

Downsampling and Upsampling Layers: Following [22] we restrict the receptive field of the downsampling layer by creating an offset of one pixel (zero-pad with a row of zeros on the top and remove a row of pixels from below) before performing max-pooling using a 2×2 kernel. This operation restricts the field of view of the downsampling and upsampling operation pair.
Note that we do not use BatchNorm [16] layers in UDVD as computing the spatial mean and variance would modify the field of view to include the center pixel.

A.2. Adding the Noisy Pixel Back

The denoised generated by the proposed architecture at each pixel is computed without using the noisy observation at that location. This avoids overfitting – i.e. learning the trivial identity map that minimizes the mean-squared error cost function – but ignores important information provided by the noisy pixel. In the case of Gaussian additive noise, we can use this information via a precision-weighted average between the network output and the noisy pixel value. Following [22], the weights in the average are derived by assuming a Gaussian distribution for the error in the blind-spot estimates of the (color) pixel values. The CNN architecture is trained to estimate the mean and covariance of this distribution at each pixel by maximizing the log likelihood of the noisy data. We explain this in detail in the following paragraphs.

UDVD estimates the value of a pixel based on the noisy pixels in its neighbourhood. We model the distribution of the three color channels of a pixel $x \in \mathcal{R}^3$ given the noisy neighbourhood Ω_y as $p(x|\Omega_y) = \mathcal{N}(\mu_x, \Sigma_x)$, where $\mu_x \in \mathcal{R}^3$ and $\Sigma_x \in \mathcal{R}^3$ represent the mean vector and covariance matrix. Let $y = x + \eta$, $\eta \sim \mathcal{N}(0, \sigma^2 I_3)$ be the observed noisy pixel. We integrate the information in the noisy pixel with the UDVD output by computing the mean of the posterior $p(x|y, \Omega_y)$, given by

$$p(x|y, \Omega_y) \propto p(y|x)\, p(x|\Omega_y) \tag{5}$$

where $p(x|\Omega_y)$ is the prior and $p(y|x)$ is the noise model. Since both the prior and the noise model are Gaussian, we can write the optimal posterior estimate as,

$$E[x|y] = (\Sigma_x^{-1} + \sigma^{-2} I)^{-1} (\Sigma_x^{-1} \mu_x + \sigma^{-2} y). \tag{6}$$

Note that the posterior mean has a very intuitive interpretation. When the signal variance is high compared to noise variance (i.e. the uncertainty in our estimation is high) the posterior mean favours noisy pixel value. We estimate μ_x and Σ_x as a function of the neighbourhood Ω_y using the network architecture discussed earlier. If x is a grayscale image, then the output of the network consists of two channels - one for μ_x and one for σ_x. When the input image has k channels, the output consists of k channels for μ_x and $k(k-1)/2$ for the upper-triangular entries of Σ_x

One can estimate μ_x and Σ_x directly from the noisy data by maximizing the likelihood. Using our distributional assumptions, the noisy pixels y follows a Gaussian distribution, $y \sim \mathcal{N}(\mu_y, \Sigma_y)$, where $\mu_y = \mu_x$ and $\Sigma_y = \Sigma_x + \sigma^2 I$. Therefore, the loss function or the negative log likelihood is:

$$\mathcal{L}(\mu_x, \Sigma_x) = \frac{1}{2}[(y - \mu_x)^T (\Sigma_x + \sigma^2 I)^{-1} (y - \mu_x)] + \frac{1}{2} \log |\Sigma_x + \sigma^2 I|. \tag{7}$$

If σ is unknown during training and has to be estimated, we use a separate neural network with the same architecture to do so. In such cases, we add a small regularization term equal to -0.1σ for numerical stability, following [22].

For the experiments with real data, the noise distribution is unknown, so we simply ignore the central pixel.

Name	N_{out}	Function
Input	k_1	
enc_conv_0	48	Convolution 3×3
enc_conv_1	48	Convolution 3×3
enc_conv_2	48	Convolution 3×3
pool_1	48	MaxPool 2×2
enc_conv_3	48	Convolution 3×3
enc_conv_4	48	Convolution 3×3
enc_conv_5	48	Convolution 3×3
pool_2	48	MaxPool 2×2
enc_conv_6	96	Convolution 3×3
enc_conv_7	96	Convolution 3×3
enc_conv_8	48	Convolution 3×3
upsample_1	48	NearestUpsample 2×2
concat_1	96	Concatenate output of pool_1
dec_conv_0	96	Convolution 3×3
dec_conv_1	96	Convolution 3×3
dec_conv_2	96	Convolution 3×3
dec_conv_3	96	Convolution 3×3
upsample_2	96	NearestUpsample 2×2
concat_2	96+k_1	Concatenate output of Input
dec_conv_4	96	Convolution 3×3
dec_conv_5	96	Convolution 3×3
dec_conv_6	96	Convolution 3×3
dec_conv_7	k_2	Convolution 3×3

Table 4. **Network architecture used for UDVD**. The convolution and pooling layers are the blind-spot variants described in Section A.1. k_1 and k_2 represent the number of input and output channels of the base network respectively.

A.3. Architecture and Training

Architecture: The overall architecture is explained in Section 3 of the paper. The network architecture for the D1 and D2 blocks is described in Table 4. D1 has $k_1 = 9$ input channels and $k_2 = 32$ output channels. D2 has $k_1 = 96$ input channels and $k_2 = 96$ output channels. The architecture of D1 and D2 are analogous to the blocks in FastDVDnet [39] to facilitate fair comparison with the supervised models. As described in Fig. 2 of the paper, D2 is followed by a derotation and the output is passed to a series of three cascaded 1×1 convolutions and non-linearity for reconstruction with 4 and 96 intermediate output channels, as in [22]. The final convolutional layer is linear and has 9 output channels, 3 representing the RGB value of the denoised image and 6 representing its covariance matrix. We use the same architecture for fluorescence microscopy and electron microscopy with the number of input channels to UDVD modified to 5 and number of output channels modified to 1.

Training Details: Following the convention in image and video denoising, we train UDVD on 128×128 patches extracted from our dataset [50, 31, 22, 39, 38] (this is also consistent with the training methodology of the supervised baselines). For the natural video and fluorescence microscopy datasets, no data augmentation was applied. For electron microscopy dataset, we applied spatial flipping, time reversal and time subsampling (i.e. skipping frames).

Optimization Details: All networks were trained using Adam [19] optimizer with a starting learning of 10^{-4}. The learning rate was decreased by a factor of 2 at checkpoints $[20, 25, 30]$ during a total training of 40 epochs. We did not experiment with other learning rate schedules such as cosine scheduling, which is a popular choice in unsupervised image denoising [22].

	DAVIS				Set8			
	Supervised CNN	Unsupervised CNN (UDVD)			Supervised CNN	Unsupervised CNN (UDVD)		
σ	5 frames	1 frame	3 frames	5 frames	5 frames	1 frame	3 frames	5 frames
20	35.86	34.13	34.91	35.16	33.37	32.39	33.09	33.36
30	34.06	32.80	33.48	33.92	31.60	30.91	31.62	32.01
40	32.80	31.48	32.20	32.68	30.37	29.63	30.42	30.82
50	31.83	30.47	31.20	31.70	29.42	28.65	29.47	29.89
60	31.01	29.65	30.39	30.90	29.08	27.86	28.70	29.13
70	30.21	28.96	29.70	30.22	28.37	27.20	28.06	28.49
80	29.28	28.37	29.10	29.63	27.60	26.65	27.50	27.94

Table 5. **Performance of UDVD**. Table shows the mean PSNR values of a state-of-the-art supervised video denoiser (FastDVDnet [39]) and UDVD with the denoised frame being predicted from $k \in \{1, 3, 5\}$ surrounding frames. The performance of UDVD monotonically increases with k and is comparable for supervised denoising across all noise levels. All the three UDVD networks reported here are trained for only $\sigma = 30$. FastDVDnet is trained for $\sigma \in [5, 55]$.

B. Ablation Study on Number of Input Frames

We perform an ablation study on the number of frames k UDVD uses as input, $k \in \{1, 3, 5\}$. UDVD with $k = 1$ is equivalent to the blind-spot network proposed for image denoising in [22]. In this section we describe the architectural and training details for UDVD with $k \in \{1, 3, 5\}$ and present some additional results.

Architectural Details: UDVD with $k = 1$ contains only one UNet style network in each branch with architecture described in Table 4 and Section A.3. There are 3 input channels and 9 output channels (3 for the RGB channels in each denoised pixel and 6 for the corresponding covariance matrix). UDVD with $k = 3$ has a similar architecture as for $k = 1$ but has 9 input channels instead (3 channels for each frame). The architecture for $k = 5$ is described in Section A.3.

Training Details: UDVD with $k \in \{1, 3, 5\}$ was trained on the DAVIS dataset with $\sigma = 30$. The training details were as described in Section A.3.

Results: As shown in Table 1 of the paper and Table 5 performance improves substantially and monotonically with k (the number of surrounding frames used to denoise each frame) across a wide range of noise levels. This difference in performance can also be observed visually. Fig 6 shows an example where the texture details of the brick wall and the fence are not well recovered when using only a single noisy frame. The texture is estimated better when using 5 noisy frames to predict the denoised output.

C. Denoising Results on Natural Video Datasets

C.1. Comparison to Supervised Video Denoising

In this section we provide additional comparisons between UDVD and supervised CNN-based methods.

1. Table 5 shows the performance of UDVD trained at $\sigma = 30$, and FastDVDnet trained for $\sigma \in [5, 55]$ when evaluated on the DAVIS test set and Set8 corrupted with $\sigma \in \{20, 40, \ldots, 80\}$. UDVD achieves comparable performance to FastDVDnet on DAVIS test set and slightly outperforms it on Set8 at all noise levels.
2. Examples of noisy videos, and denoised counterparts obtained using UDVD are included in the official github repository[2] (`hypermooth.mp4`, `rafting.mp4`, `motorbike.mp4` and `snowboard.mp4`).

C.2. Comparison to Burst Denoising

When several photographs are captured in quick succession to each other, the resulting set of images are often blurry or noisy (particularly when the object is in motion). Burst denoising aims to recover estimate the original scene from the set of burst photographs. Recent methods have solved burst denoising by applying deep neural network to map a stack of burst images to a single clean frame [13, 30, 45]. A popular burst denoising method, KPN [30] achieved a PSNR of 27.83 on

[2] https://github.com/sreyas-mohan/udvd

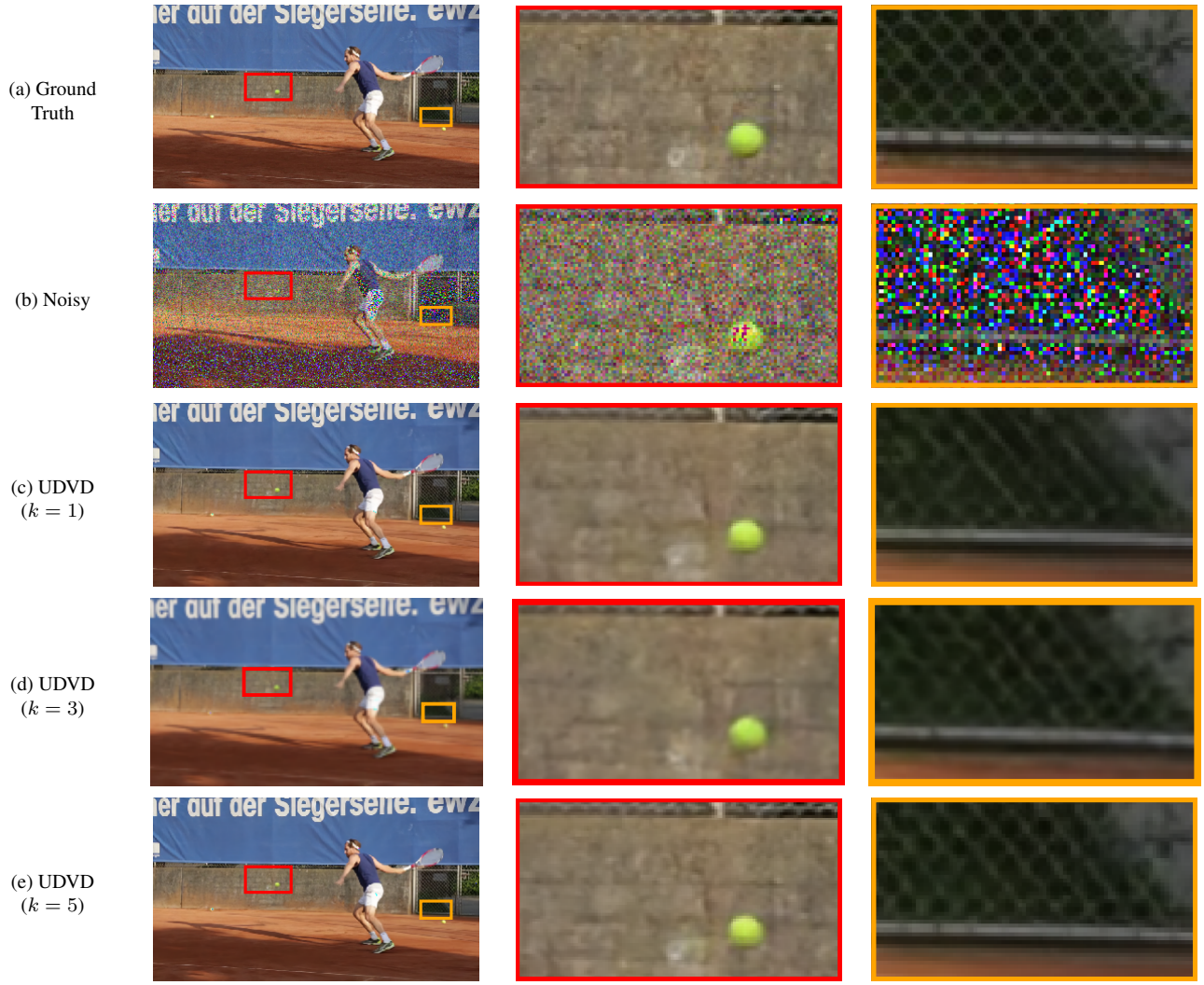


Figure 6. **Comparison of blind image and video denoising**. Example from the DAVIS dataset. (a) Ground truth frame. (b) Noisy frame. (c) Reconstruction using a single frame. The texture details of the brick wall and the fence are not recovered well. Reconstruction using (d) 3 and (e) 5 surrounding frames produces an improved texture estimate.

the DAVIS dataset at $\sigma = 30$ [3], while UDVD achieves a PSNR of 33.92. UDVD is expected to outperform burst denoising methods as these methods (1) are trained for jittered motions, and cannot exploit systematic motion in natural videos like video denoising methods, and (2) often do not expect a motion change of more than 2 pixels from one frame to another [30], while the motion in natural videos is usually much larger (see Section 6 in main paper).

D. UDVD-S: Denoising Using Only a Single Video

UDVD, combined with aggressive data augmentation and early stopping, achieves state-of-the-art performance even when trained on only a single short video. In this section, we analyze the contribution of each of the data augmentation and early stopping scheme to the performance of UDVD-S through an ablation study. We also provide more details about our comparison to MF2F [9].

D.1. Details of test sets.

We evaluate UDVD-S and baselines on the following four datasets:

1. **DAVIS [33]**: We take all the 30 sequences from the test set of the DAVIS Challenge 2017.

[3] Evaluated using the pre-trained model provided here: https://github.com/z-bingo/kernel-prediction-networks-PyTorch

		$\sigma = 30$									
		ten-v	snow	hyper	raft	motor	trac	sunf	touch	park	**mean**
No. of frames		75	59	37	29	32	85	85	85	85	-
No Aug	(without ES)	33.37	29.10	29.72	27.26	27.28	32.52	35.07	32.65	30.20	30.80
No Aug	(with ES)	34.35	30.67	32.42	30.72	29.21	33.08	37.04	33.63	30.40	32.39
F	(without ES)	34.00	30.60	30.15	30.16	28.44	33.09	36.86	33.56	30.37	31.91
F	(with ES)	34.68	30.76	32.41	30.76	29.33	33.35	37.13	33.74	30.53	32.52
F+TR	(without ES)	34.18	30.73	31.06	30.31	28.98	33.53	**37.29**	33.51	**30.56**	32.24
F+TR	(with ES)	*34.70*	*30.78*	**32.60**	*30.80*	**29.36**	**33.54**	**37.29**	**33.88**	**30.56**	**32.61**
UDVD*		**34.82**	**30.83**	32.34	30.82	29.24	31.73	35.33	33.48	28.98	31.95
FastDVDnet*		34.58	30.78	32.48	**30.94**	29.35	31.39	35.06	33.71	28.73	31.89
MF2F - 8 sigmas		34.45	30.44	30.93	29.70	28.81	31.61	34.43	33.41	28.79	31.40
MF2F - online no teacher		34.50	30.42	30.54	29.45	28.40	32.11	35.19	33.47	28.89	31.44
MF2F - online with teacher		34.48	30.44	31.13	*29.91*	*28.92*	32.08	35.20	33.44	28.91	31.61
MF2F - offline no teacher		*34.66*	30.49	30.20	29.38	28.36	*32.19*	35.50	33.58	28.98	31.48
MF2F - offline with teacher		34.63	*30.52*	*31.16*	29.55	*28.92*	31.93	*35.52*	*33.61*	*29.04*	*31.65*

Table 6. **Results for UDVD and MF2F trained on individual noisy videos for** $\sigma = 30$. The top block show PSNR values for UDVD trained on each individual video sequence with and without data augmentation (spatial flipping(F) and time-reversal(TR)) and early stopping (ES). Early stopping was performed using the last 5 frames of each video as the held-out set. The last block shows the result of running MF2F [9] with all the 5 different fine-tuning scheme proposed in Ref. [9]. With the augmentations and early stopping, UDVD-S, on average outperforms UDVD and FastDVDnet trained on the full DAVIS dataset (indicated by *) and MF2F which fine-tunes a pre-trained FastDVDNet on each individual video. The best performing method for each video is highlighted in bold and the best performing method in each block is highlighted in italics. The tennis-vest video is from DAVIS and the rest of the 8 videos are from Set8.

		$\sigma = 90$									
		ten-v	snow	hyper	raft	motor	trac	sunf	touch	park	**mean**
No. of frames		75	59	37	29	32	85	85	85	85	-
No Aug	(without ES)	24.13	22.89	22.04	20.99	20.06	24.84	25.98	25.67	23.35	23.33
No Aug	(with ES)	30.15	25.49	27.48	26.05	23.79	28.18	31.91	29.87	25.46	27.60
F	(without ES)	27.21	24.42	24.05	23.32	21.84	27.42	29.53	28.01	25.03	25.65
F	(with ES)	30.35	**25.60**	27.72	*26.16*	23.89	**28.71**	32.17	29.93	25.59	27.79
F+TR	(without ES)	27.11	24.77	24.25	23.55	21.98	27.80	30.22	28.56	25.44	25.96
F+TR	(with ES)	**30.40**	25.59	**27.75**	*26.16*	**23.92**	28.63	**32.18**	**29.96**	**25.62**	**27.80**
UDVD*		28.78	25.16	26.78	25.81	23.57	26.42	29.04	28.71	24.23	26.50
FastDVDnet*		29.44	25.25	27.30	**26.35**	23.68	27.42	30.29	29.61	24.72	27.12
MF2F - 8 sigmas		28.79	25.04	27.14	26.21	23.56	26.89	29.19	29.04	24.35	26.69
MF2F - online no teacher		28.35	25.12	26.67	26.07	23.39	27.28	30.01	29.49	24.64	26.78
MF2F - online with teacher		*29.44*	*25.25*	*27.30*	**26.35**	*23.68*	*27.42*	30.09	29.53	24.71	*27.08*
MF2F - offline no teacher		28.70	25.17	26.64	26.02	23.41	27.42	*30.29*	29.60	*24.72*	26.89
MF2F - offline with teacher		28.79	*25.25*	27.22	26.31	23.62	27.34	*30.29*	*29.61*	24.69	27.01

Table 7. **Results for UDVD and MF2F trained on individual noisy videos for** $\sigma = 90$. The top block show PSNR values for UDVD trained on each individual video sequence with and without data augmentation (spatial flipping(F) and time-reversal(TR)) and early stopping (ES). Early stopping was performed using the last 5 frames of each video as the held-out set. The last block shows the result of running MF2F [9] with all the 5 different fine-tuning scheme proposed in Ref. [9]. With the augmentations and early stopping, UDVD-S, on average outperforms, UDVD or FastDVDnet trained on the full DAVIS dataset (indicated by *) and MF2F which fine-tunes on a pre-trained FastDVDNet on each individual video. The best performing method for each video is highlighted in bold and the best performing method in each block is highlighted in italics. The tennis-vest video is from DAVIS and the rest of the 8 videos are from Set8.

2. **Set8 [39]**: Following FastDVDNet [39], we use 4 sequences from the GoPro set (*hypersmooth, motorbike, rafting, snowboard*) and 4 sequences from the Derfs Test Media Collection (*park_joy, sunflower, touchdown, tractor*).
3. **Derfs**: Following [9], we use 7 sequences from the Derfs Test Media Collection, which are *park_joy, sunflower, touchdown, tractor* (shared with Set8), and *blue_sky, old_town_cross, pedestrian_area*. We use the first 85 frames from each sequences with a spatial-resolution of 960×540 [39].
4. **Vid3oC [18]**: We use the first 10 sequences (*000 to 009*) out of the 50 available sequences.

D.2. Ablation study

We train UDVD-S on 128×128 patches extracted from the noisy video. (see Section A.3) for more details). For each patch, we apply each of the following data augmentations at random:

1. **Spatial flipping**: We flip all the 5 input patches vertically or horizontally. This operation only rearranges the pixel location and does not combine the pixel together in anyway, making sure that the noise statistics is still preserved after the augmentation.
2. **Time reversal**: We reverse the order of frames in the input to generate a new video. Like spatial flipping, this operation also preserves the noise statistics.

In addition to data augmentation, we employ early stopping by choosing the model parameters which produced the best error on a a held-out set of frames. We pick the last 5 frames of each video as our held out set. Tables 6 and 7 show an ablation study over data augmentations and early stopping for 9 different videos at two different noise levels, $\sigma = 30$ and $\sigma = 90$. Across videos and noise levels, data augmentation and early stopping significantly increase the performance of our method.

D.3. Comparison with MF2F

We compare the performance of UDVD-S to an unsupervised denoising method MF2F [9]. MF2F fine-tunes a pre-trained CNN on the noisy video using an objective function involving optical flow. The pre-trained CNN used in MF2F is FastDVDNet [39], which is trained with supervised on a large dataset of natural videos(DAVIS [33]). The authors of MF2F provide five different schemes for fine-tuning: 8 sigmas, online no teacher, online with teacher, offline no teacher and offline with teacher. Tables 6 and 7 show the denoising results using each of these five training schemes. The best result (on 4 different dataets) is reported in Table 2 of the main paper. In addition to this, we also apply MF2F on real electron microscopy data (see Figure 7), where it *fails*. We used the official implementation[4] for all the training schemes.

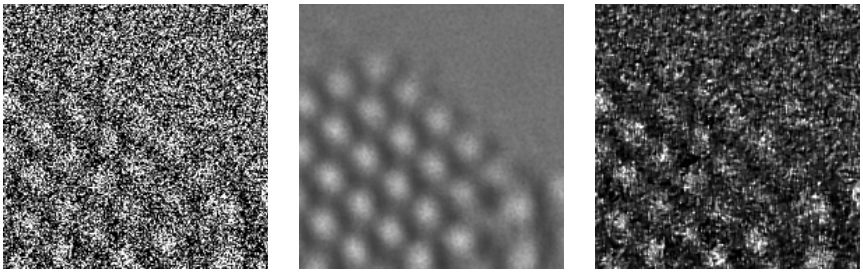


Figure 7. **UDVD-S outperforms MF2F on electron microscopy data**. UDVD-S is able to effectively denoise real-world data acquired from an electron-microscopy, but MF2F *fails*.

D.4. Measure of Confidence on Improvements

We compute the mean and standard deviation of the improvement of UDVD-S with respect to MF2F in Table 8.

	DAVIS	Set8	Derfs	Vid3oC
$\sigma = 30$	-0.33 $\pm$ 0.18	0.99 $\pm$ 0.21	1.09 $\pm$ 0.50	-0.54 $\pm$ 0.52
$\sigma = 90$	0.15 $\pm$ 0.09	0.65 $\pm$ 0.23	1.13 $\pm$ 0.39	0.26 $\pm$ 0.28

Table 8. **Measure of confidence on improvement of UDVD-S with respect to MF2F**. We compute the mean and standard deviation of the difference between performance on UDVD-S and MF2F (in PSNR) on four different datasets and two different noise levels ($\sigma = 30, 90$). UDVD-S outperforms MF2F with high certainty on two datasets at low noise level ($\sigma = 30$) and all the four datasets at high noise level ($\sigma = 90$).

[4] https://github.com/cmla/mf2f

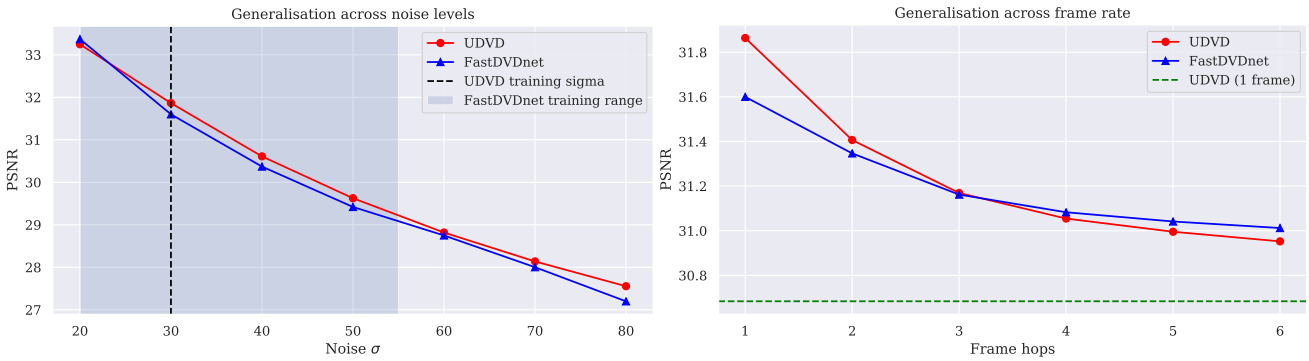


Figure 8. **Generalization across noise levels and frame rates.** (left) UDVD trained at only $\sigma = 30$ generalizes well to noise levels not seen during training. The plotted points represent mean PSNR values evaluated on Set8. (right) UDVD generalizes well to faster videos (created by skipping frames) and consistently outperforms a baseline image denoiser (UDVD with a single input frame, shown as a green dashed line).

E. Denoising Results on Real-world Datasets

Raw videos: The estimated ground truth, noisy raw data [48], and the denoised videos obtained with UDVD can be found on the official github repository (`raw_video.mp4`). The videos were converted to RGB for illustration.

As discussed in the main paper, UDVD was directly trained on the mosaiced raw videos. Existing unsupervised video denoising methods, like MF2F [9], cannot be applied directly on this dataset as their pre-trained backbone expects an input in the RGB domain. In Ref. [9], the authors convert mosaiced videos into the RGB domain, apply MF2F [9] and transform the denoised RGB videos back.

Fluorescence and electron microscopy data: The noisy fluorescence microscopy and electron microscopy data, and the denoised videos obtained with UDVD can be found on the official github repository (`fluoro_1.mp4`, `fluoro_2.mp4` and `electron.mp4`).

F. Generalization Across Noise and Frame Rate

Ideally, a denoiser should be able to denoise videos corrupted at a wide range of noise levels. This is usually achieved by training the CNN on examples corrupted with a range of noise strength [50, 39, 38]. The range of noise levels on which the network is trained is called the *training range* of the network.

Generalization outside the training range: The authors of [31] showed that CNNs trained for image denoising generalize well on test images corrupted with noise in the training range, but fails catastrophically when corrupted with noise strength outside the training range. The authors provided evidence that the overfitting is due to additive terms in the convolutional layers (and BatchNorm [16]) and showed that a CNN with no additive terms, called a *bias-free* CNN generalizes well outside the training range. UDVD uses a bias-free architecture and generalizes well to noise levels outside its training range (Fig 8).

Generalization across frame rates: To test generalization across frame rates, we simulated faster videos by skipping frames of videos in Set8. Fig 8 shows that UDVD generalizes robustly to faster videos and maintains a significant gain in performance over single-image denoising even when tested on videos where a large number of frames have been skipped (i.e. at a very low frame rate).

G. Analysis of CNN-based Video Denoising

G.1. Natural Videos

In Section 7 and Fig 4 of the paper we examined the equivalent filters and illustrated that UDVD learns to denoise by performing an average over a spatiotemporal neighbourhood of each pixel. Here we examine equivalent filters for more videos and a supervised CNN (FastDVDnet) and show that similar observations hold.

Adaptive filtering: Fig 10, 11, 12 and 13 shows filters computed at a pixel for 4 different videos at 4 different noise levels. The filters adapt to the underlying signal content. They span larger areas as the noise level increases. These observations also holds for FastDVDnet, which is trained with supervision (Fig 14)

Contribution of neighbouring frames for denoising: UDVD tends to ignore temporally distant frames at lower noise levels as shown in Fig 10, 11, 12 and 13. This phenomenon is quantified in Fig 9 by plotting the contribution of each frame to the denoised pixel by averaging over **5000** pixels from **250** random patches of size 128×128. At higher noise levels, UDVD seems to use distant frames more. This is consistent with the ablation study, which shows that for higher noise levels using more surrounding frames improves the denoising performance. Similar results hold for supervised CNN FastDVDnet, as shown in Fig 14.

Local Averaging: The weighting functions or equivalent filters perform an approximate averaging operation. They are mostly non-negative (although they do have some negative entries as depicted in blue in Fig 10, 11, 12 and 13) and they approximately sum up to one (see Fig 9).

G.2. Real-world Data

Equivalent filters for the raw video, the fluorescence-microscopy and the electron-microscopy data are shown in Fig 15. The fluorescence -microscopy data have a low noise level. As expected from the results on natural videos (see Section C), the weighting functions are mostly confined to the middle frame (as quantified in Fig 9). In the electron-microscopy dataset the weighting functions shows that the network relies on adjacent frames to estimate the denoised (as quantified in Fig 9).

G.3. Motion Estimation

Figures 10, 11, 12 and 13 show that the equivalent filters in adjoining frames are automatically shifted spatially to account for the movement of objects in the videos. We extracted motion information using the shift as explained in Section 6. Figures 16, 17, 18 and 19 show additional examples for UDVD and FastDVDnet. The estimated optical flow is mostly consistent with the estimated obtained by DeepFlow [43] applied on the clean videos. The motion estimates obtained from the equivalent filters tends to be less accurate for pixels near strongly correlated features or highly homogeneous regions where the local motion is ambiguous.

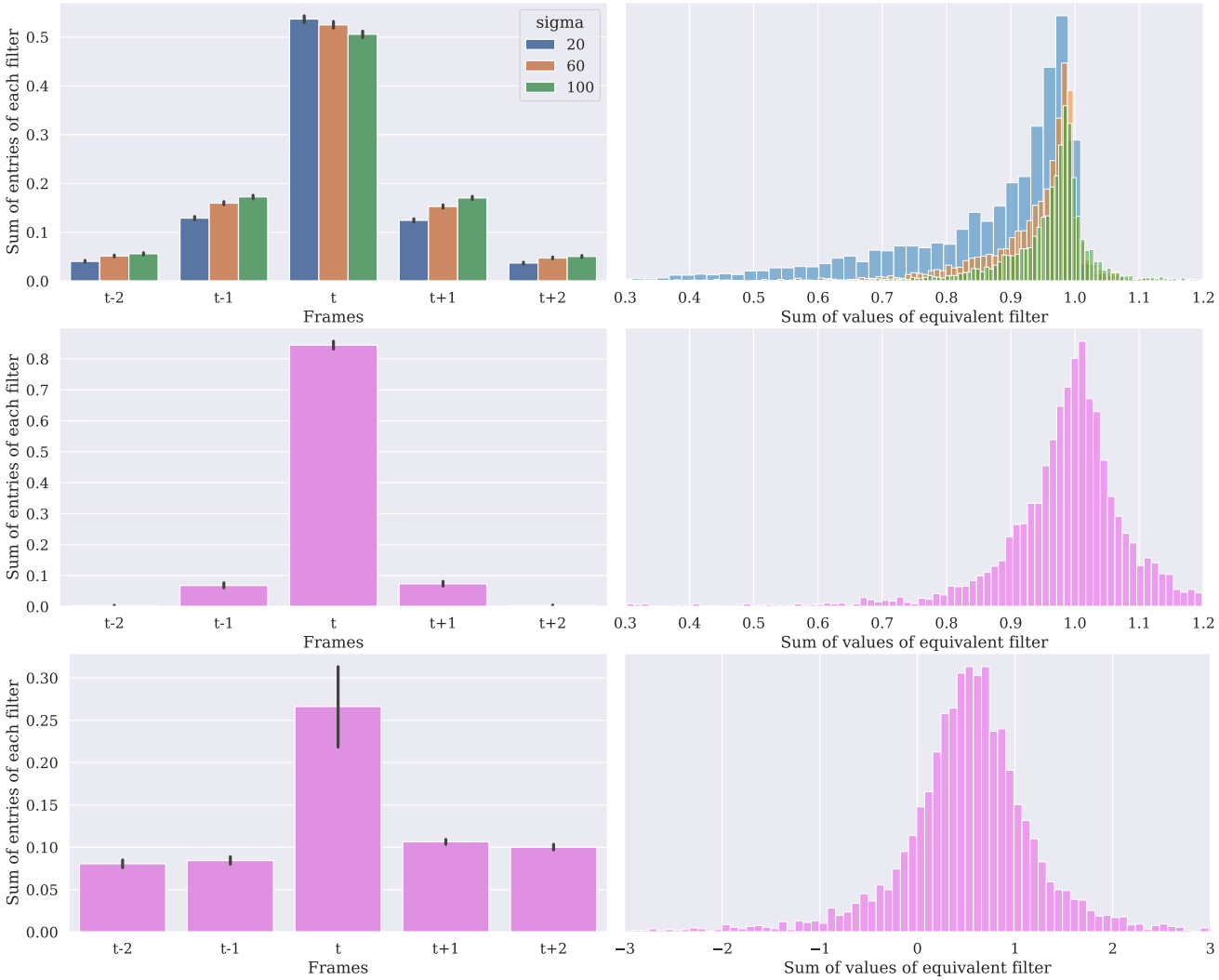


Figure 9. **Quantitative analysis of equivalent filters**. *Left column:* The graphs show the sum of the entries of the equivalent filters in each frame, averaged over 5000 pixels from 250 random patches of size 128×128. For all datasets, the central frame dominates. For the DAVIS dataset (top), the contribution from the other frames increases with the noise level. For the fluorescence-microscopy data (mid) the contribution of the other frames is rather low, due to the high signal-to-noise ratio. For the electron-microscopy dataset the contribution of the other frames is larger (bottom). *Right column:* Histogram of the sum of all entries in the equivalent filters (over all 5 frames) for 5000 pixels from 250 random patches of size 128×128 from the DAVIS test set (top), the fluorescence-microscopy dataset (mid) and the electron-microscopy dataset (bottom). For the DAVIS and fluorescence-microscopy datasets, the filters sum to 1 in most cases. The peak of electron microscopy deviates significantly from 1. This could be due to the noise model, which has non-Gaussian characteristics (it is Poisson with low counts).

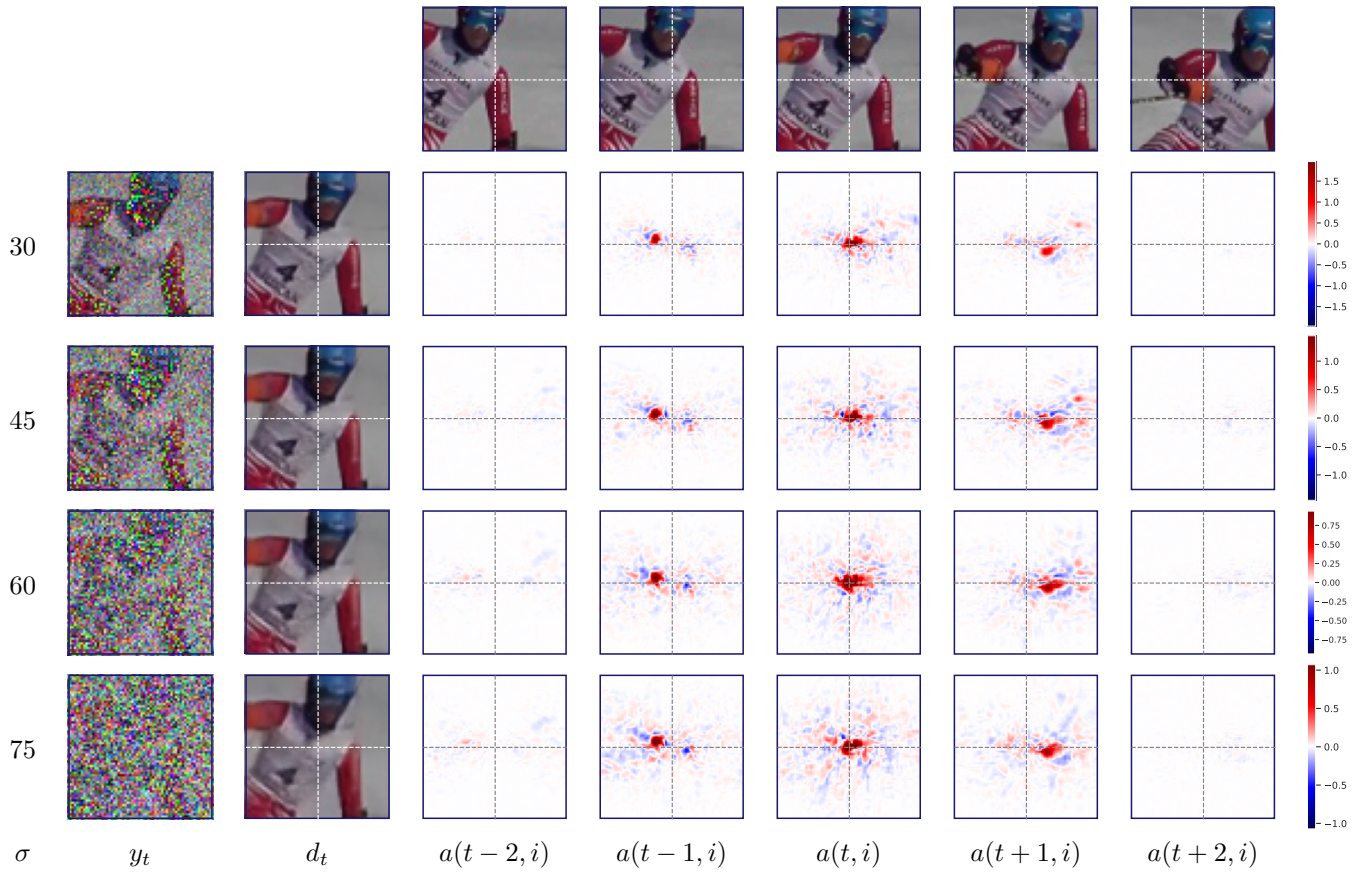


Figure 10. **Video denoising as spatiotemporal adaptive filtering; `giant-slalom` video from the DAVIS dataset**. Visualization of the linear weighting functions ($a(k,i)$, Section 6 of paper) of UDVD. The left two columns show the noisy frame y_t at four levels of noise, and the corresponding denoised frame, d_t. Weighting functions $a(k,i)$ corresponding to the pixel i (at the intersection of the dashed white lines), for five successive frames, are shown in the last five columns. The weighting functions adapt to underlying image content, and are shifted to track the motion of the skier. As the noise level σ increases, their spatial extent grows, averaging out more of the noise while respecting object boundaries. The weighting functions corresponding to the five frames approximately sum to one, and thus compute a local average (although some weights are negative, depicted in blue) as explained in Section G.1.

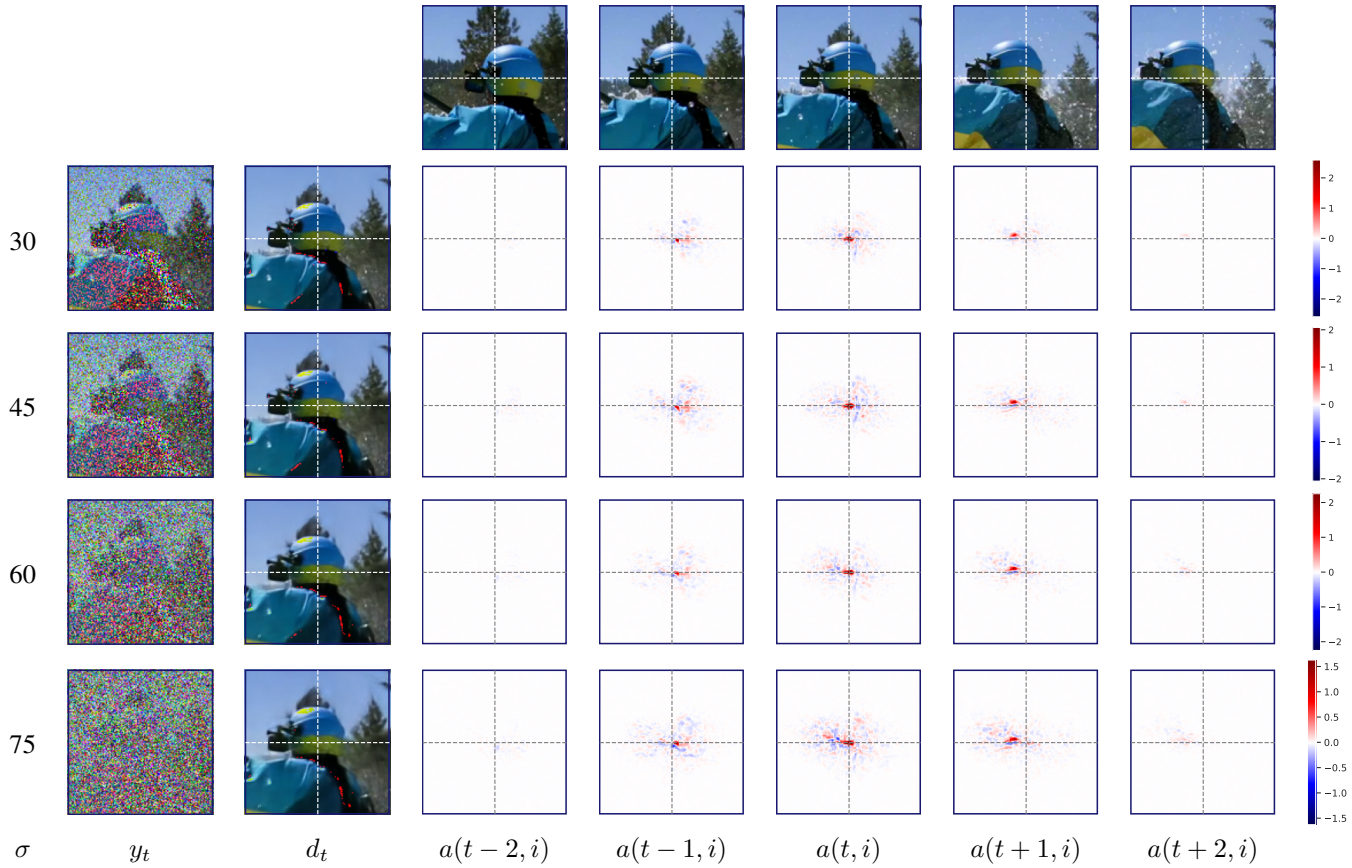


Figure 11. **Video denoising as spatiotemporal adaptive filtering; `rafting` video from the GoPro dataset**. Visualization of the equivalent filters, as described in Fig 10.

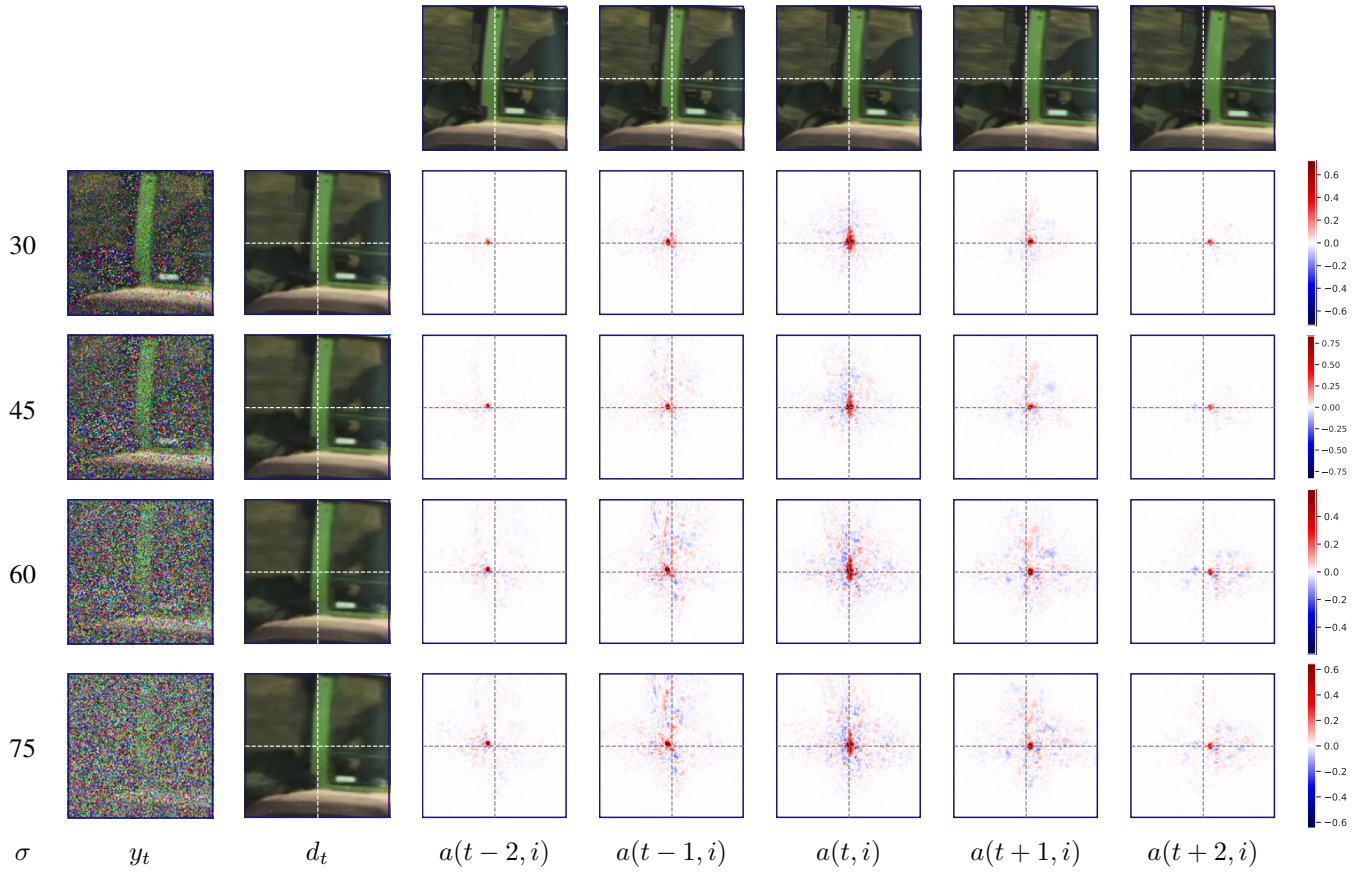


Figure 12. **Video denoising as spatiotemporal adaptive filtering; `tractor` video from Set8**. Visualization of the equivalent filters, as described in Fig 10.

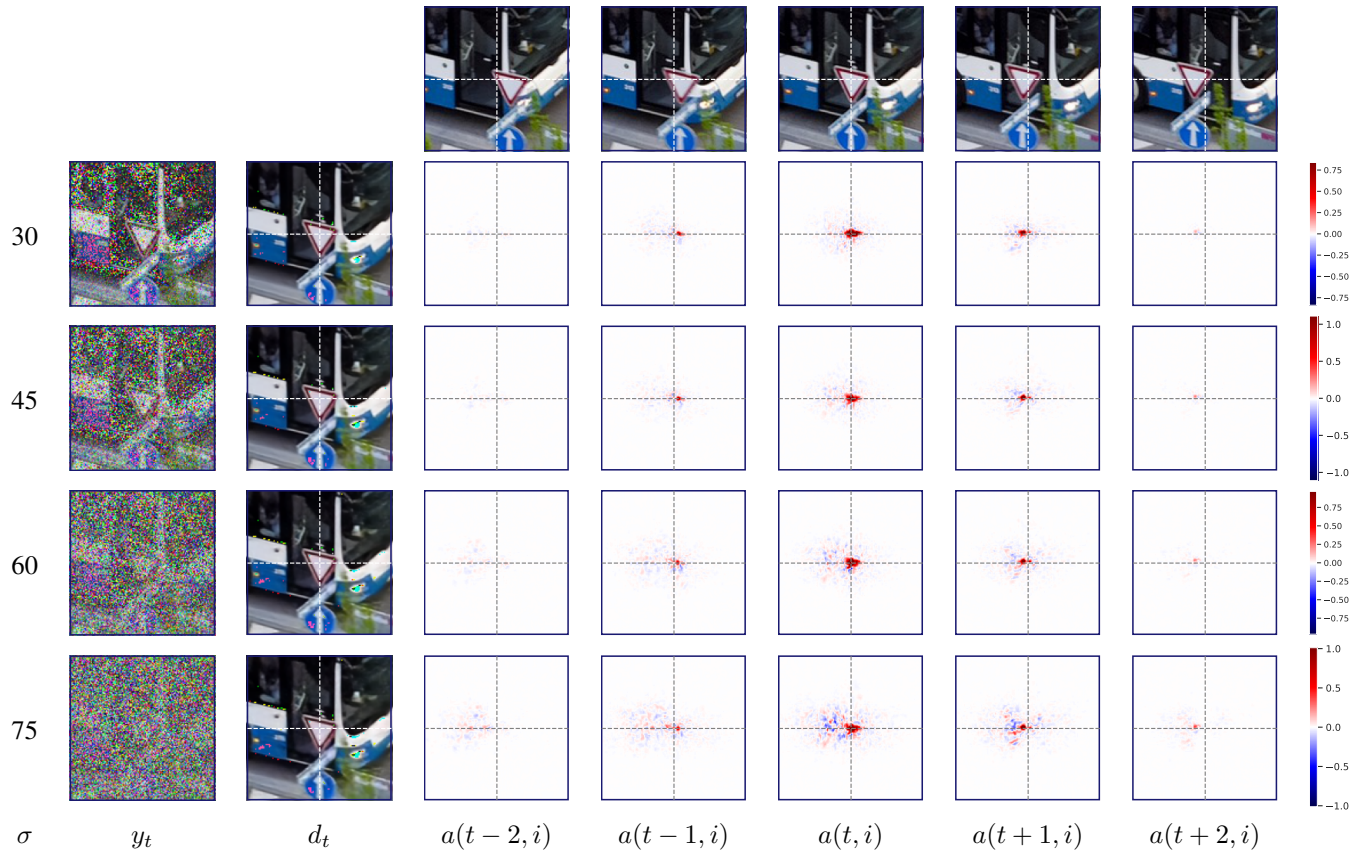


Figure 13. **Video denoising as spatiotemporal adaptive filtering; `bus` video from the DAVIS dataset**. Visualization of the equivalent filters, as described in Fig 10.

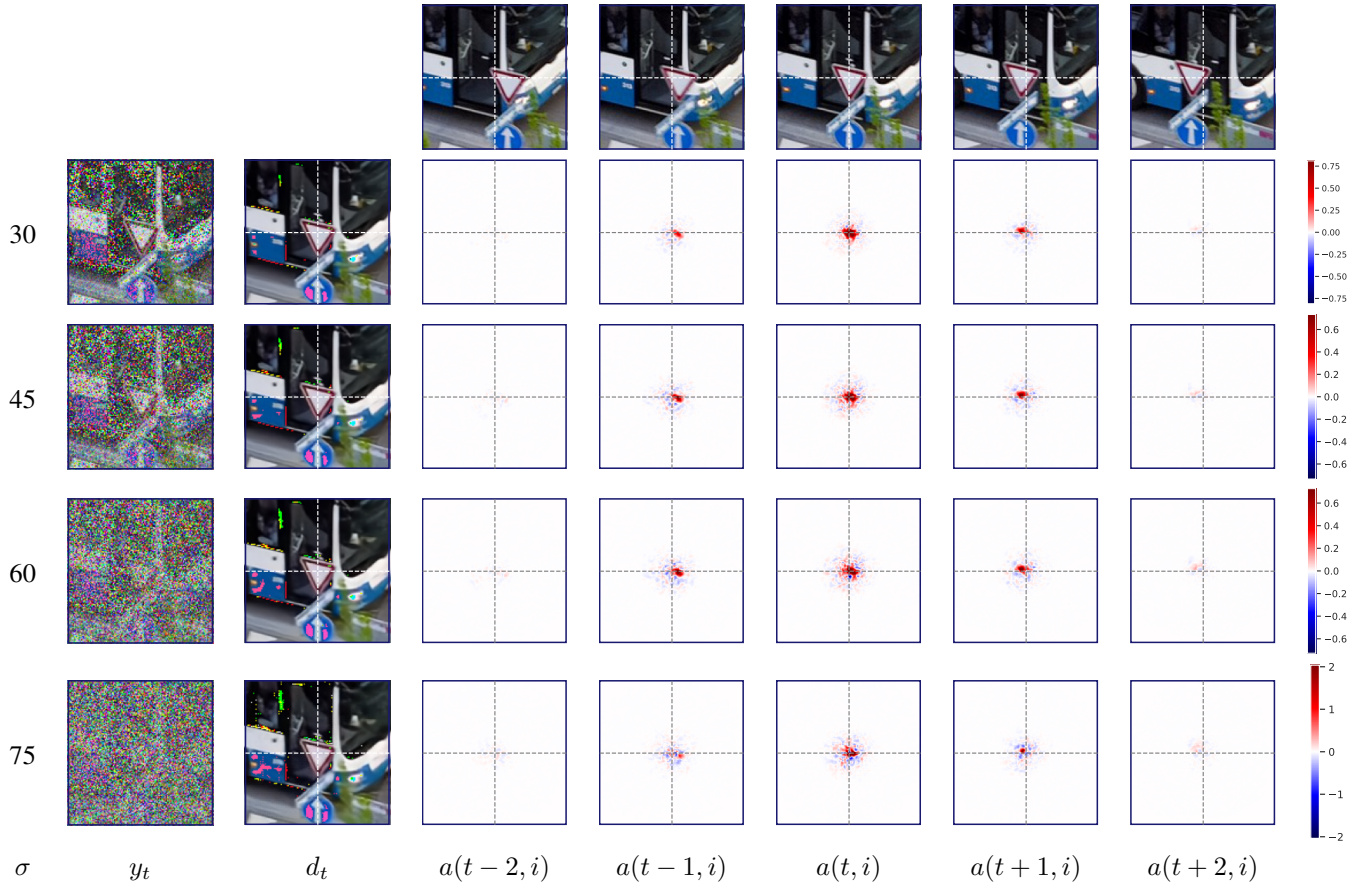


Figure 14. **Video denoising using FastDVDnet as spatiotemporal adaptive filtering; `bus` video from the DAVIS dataset**. Visualization of the linear weighting functions ($a(k,i)$, Section 6 of paper) of FastDVDnet which is trained with supervision. The left two columns show the noisy frame y_t at four levels of noise, and the corresponding denoised frame, d_t. Weighting functions $a(k,i)$ corresponding to the pixel i (at the intersection of the dashed white lines), for five successive frames, are shown in the last five columns. The weighting functions adapt to underlying image content, and are shifted to track the motion of the stop sign. As the noise level σ increases, their spatial extent grows, averaging out more of the noise while respecting object boundaries. The behavior is very similar to the corresponding filters of UDVD as shown in Fig 13.

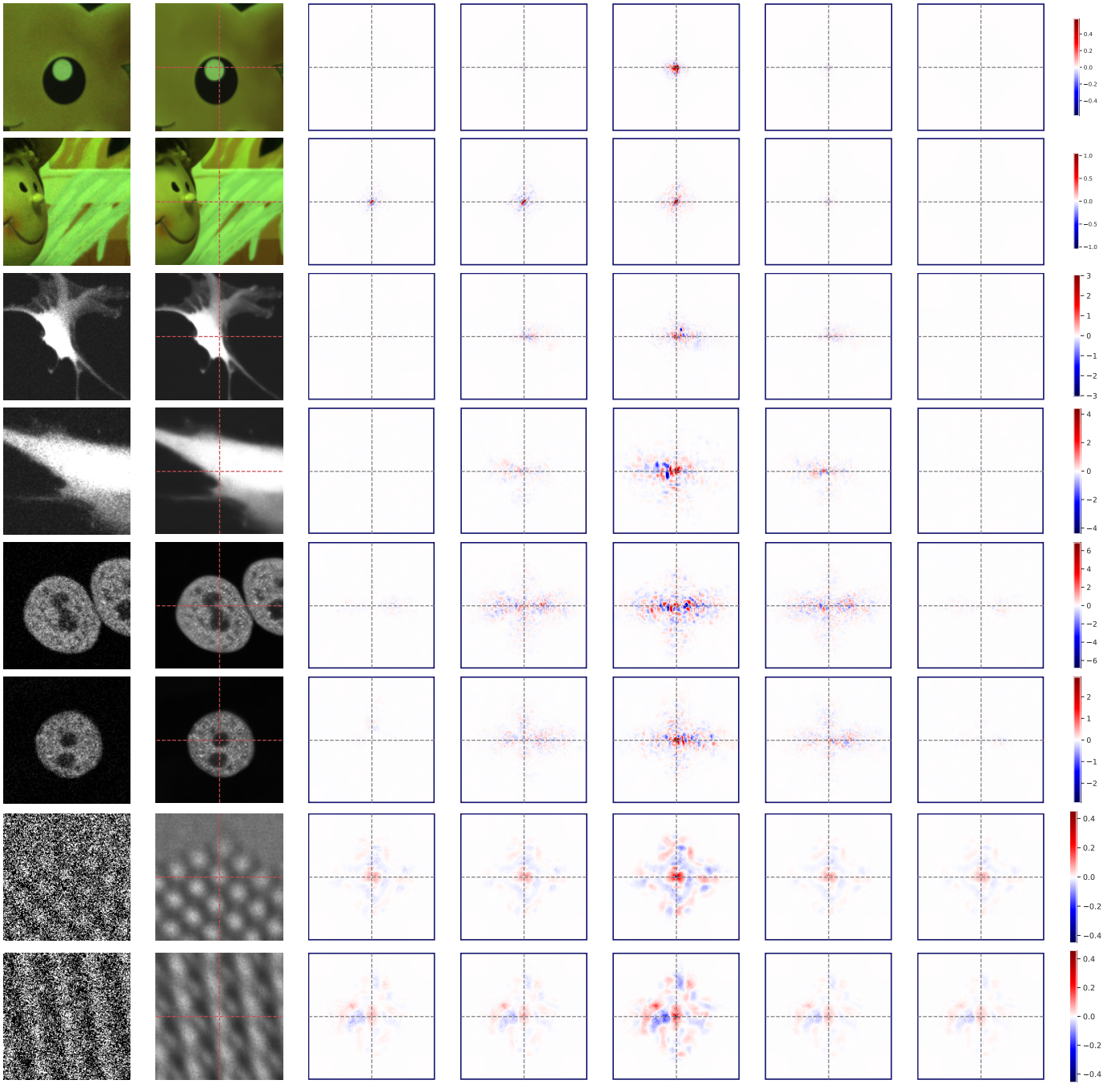


Figure 15. **Equivalent filters of UDVD when applied to real-world data**. Visualization of the linear weighting functions ($a(k,i)$, Section 6 of paper) of UDVD trained to denoise raw video, fluorescence and electron microscopy data. The left two columns show the noisy frame y_t and the corresponding denoised frame, d_t. Weighting functions $a(k,i)$ corresponding to the pixel i (at the intersection of the dashed white lines), for five successive frames, are shown in the last five columns. In raw video data and fluorescence-microscopy data, the contributions from neighbouring frames are smaller. For electron-microscopy data they are larger (see also Fig 9).

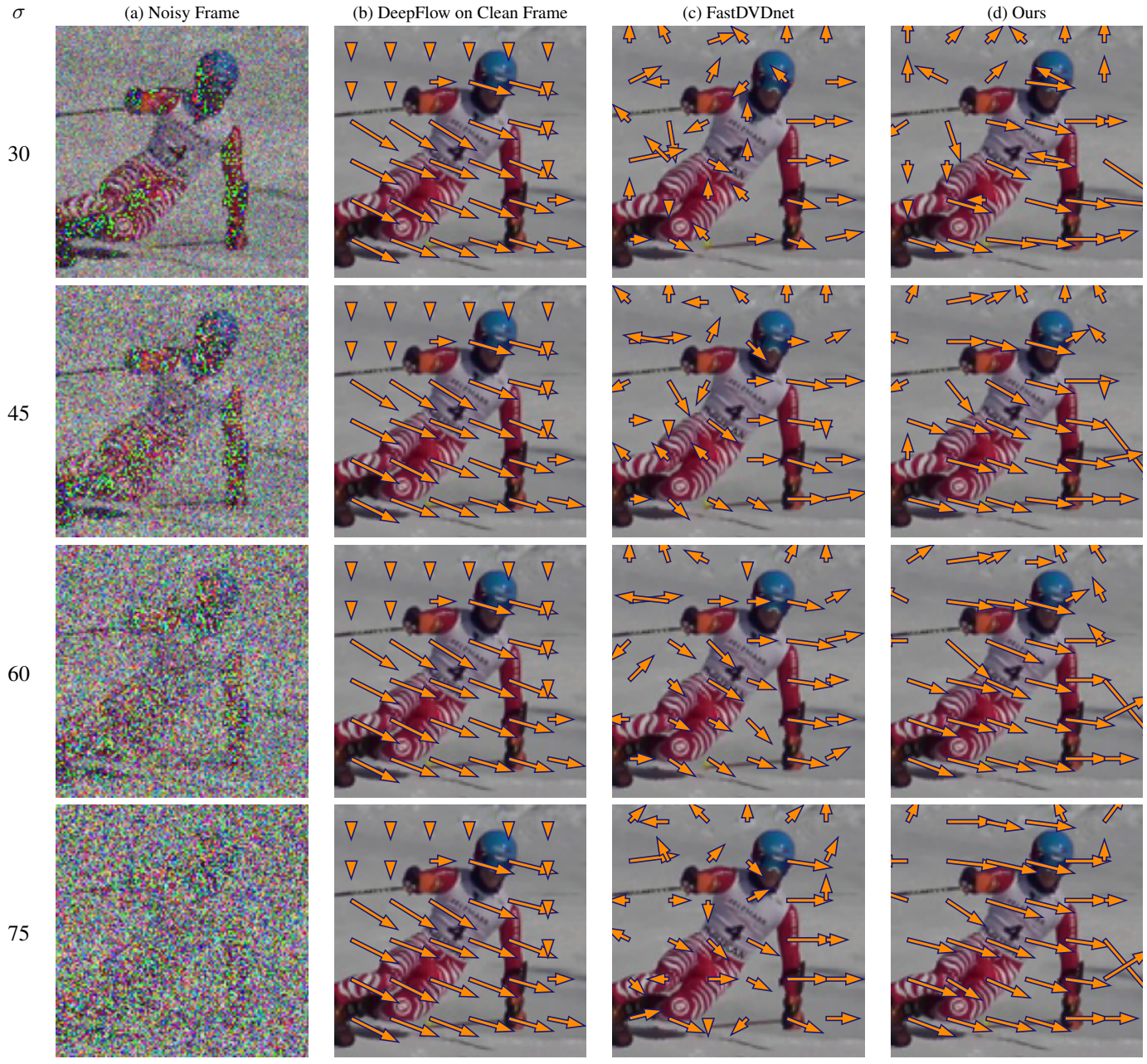


Figure 16. **CNNs trained for denoising automatically learn to perform motion estimation**. (a) Noisy frame from `giant-slalom` video in the DAVIS dataset. (b) Optical flow direction at multiple locations of the image obtained using a state-of-the-art algorithm applied *to the clean video*. Optical flow direction estimated from the shift of the adaptive filter obtained from the gradients of (c) FastDVDnet and (d) UDVD, both of which are trained with no optical flow information. FastDVDnet is trained with supervision. Optical flow estimates are well-matched to those in (b), but are not as accurated at oriented features, and in homogeneous regions where local motion is not well defined (e.g. in the background). Each row corresponds to a different noise levels. At higher noise levels, the networks perform averages over more frames, improving the motion estimation results.

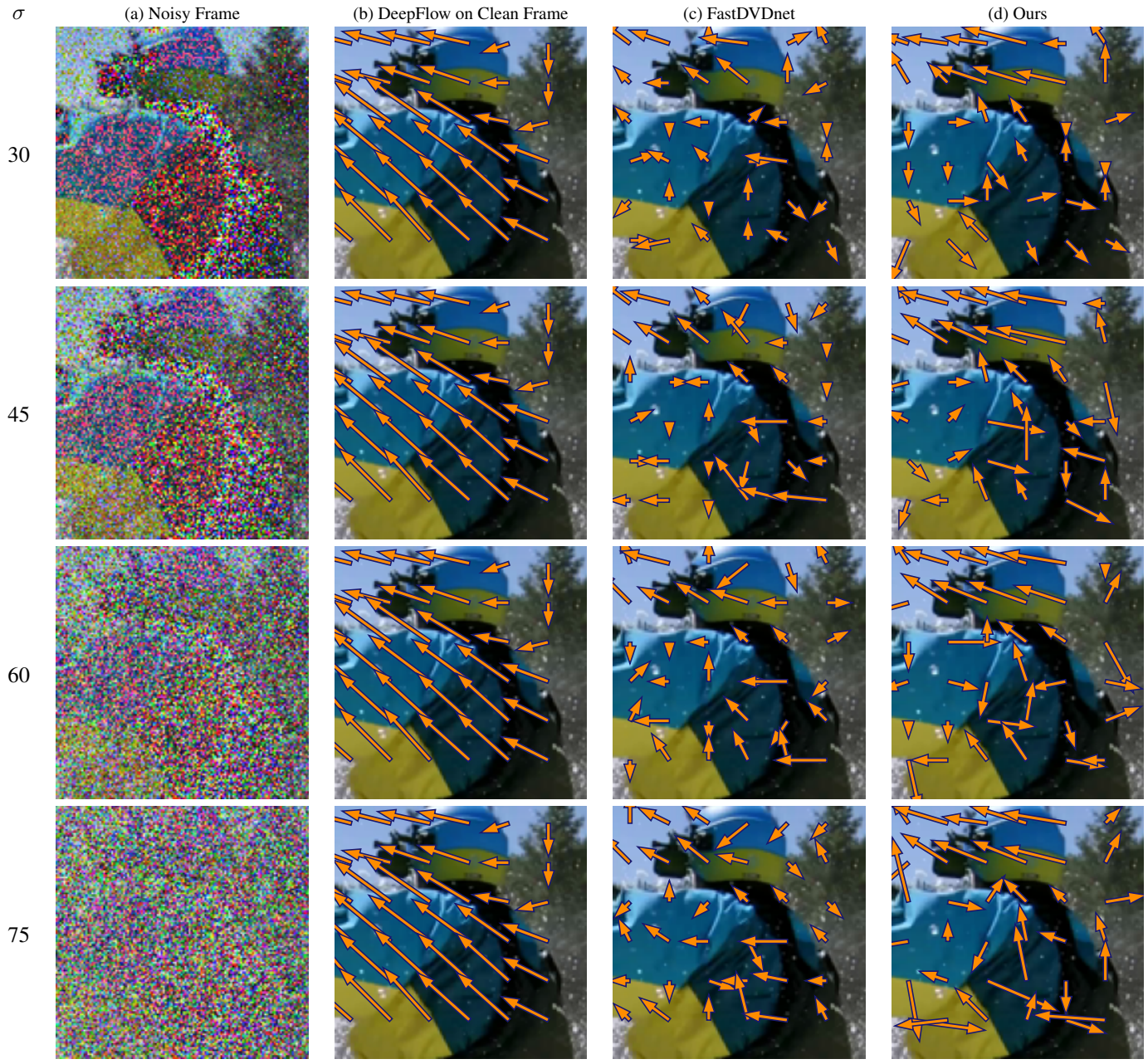


Figure 17. **CNNs trained for denoising automatically learn to perform motion estimation; `rafting` video from Set8**. Motion estimated from the gradients of UDVD and FastDVDnet. See description of Figure 16.

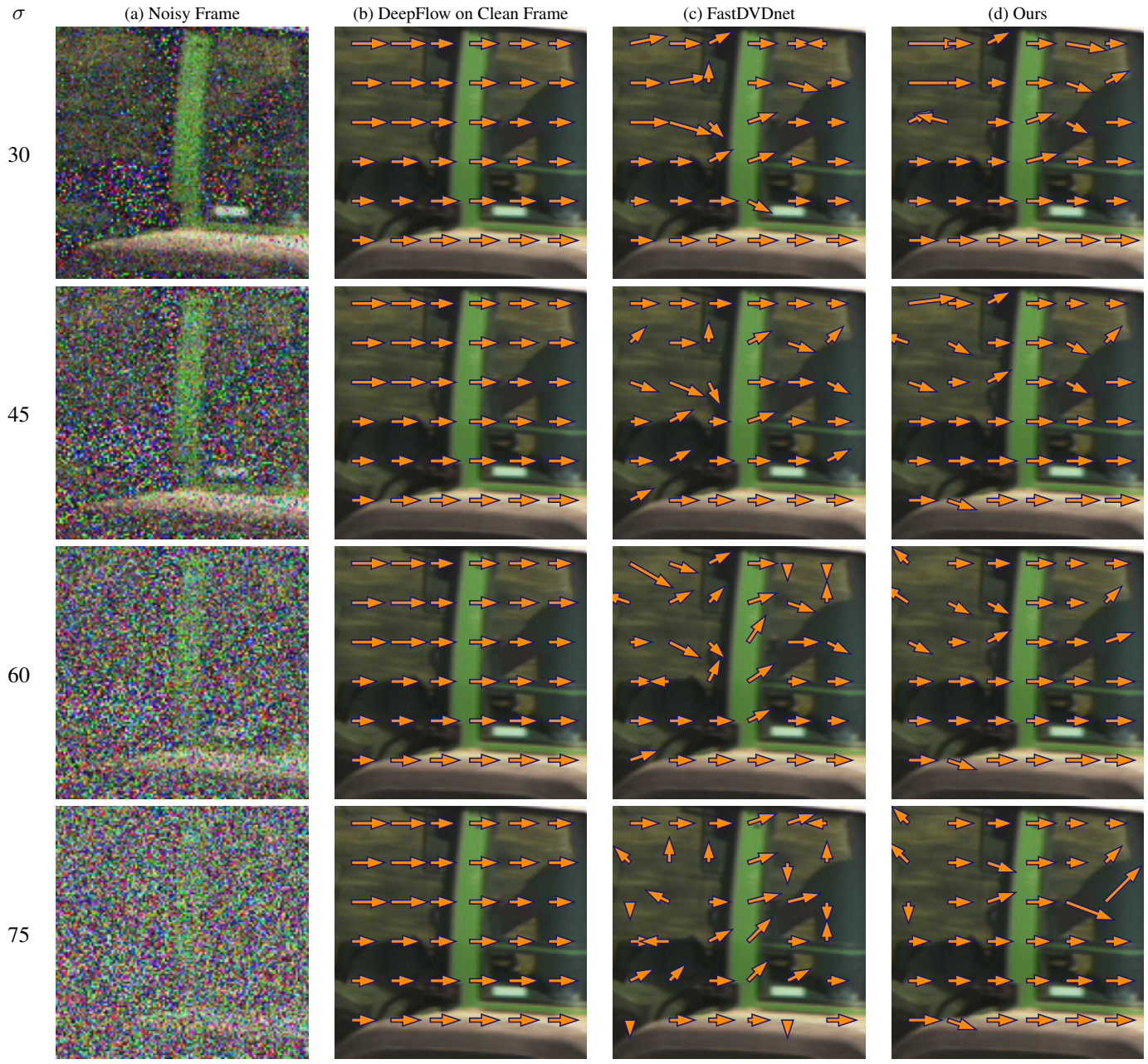


Figure 18. **CNNs trained for denoising automatically learn to perform motion estimation; `tractor` video from Set8**. Motion estimated from the gradients of UDVD and FastDVDnet. See description of Figure 16.

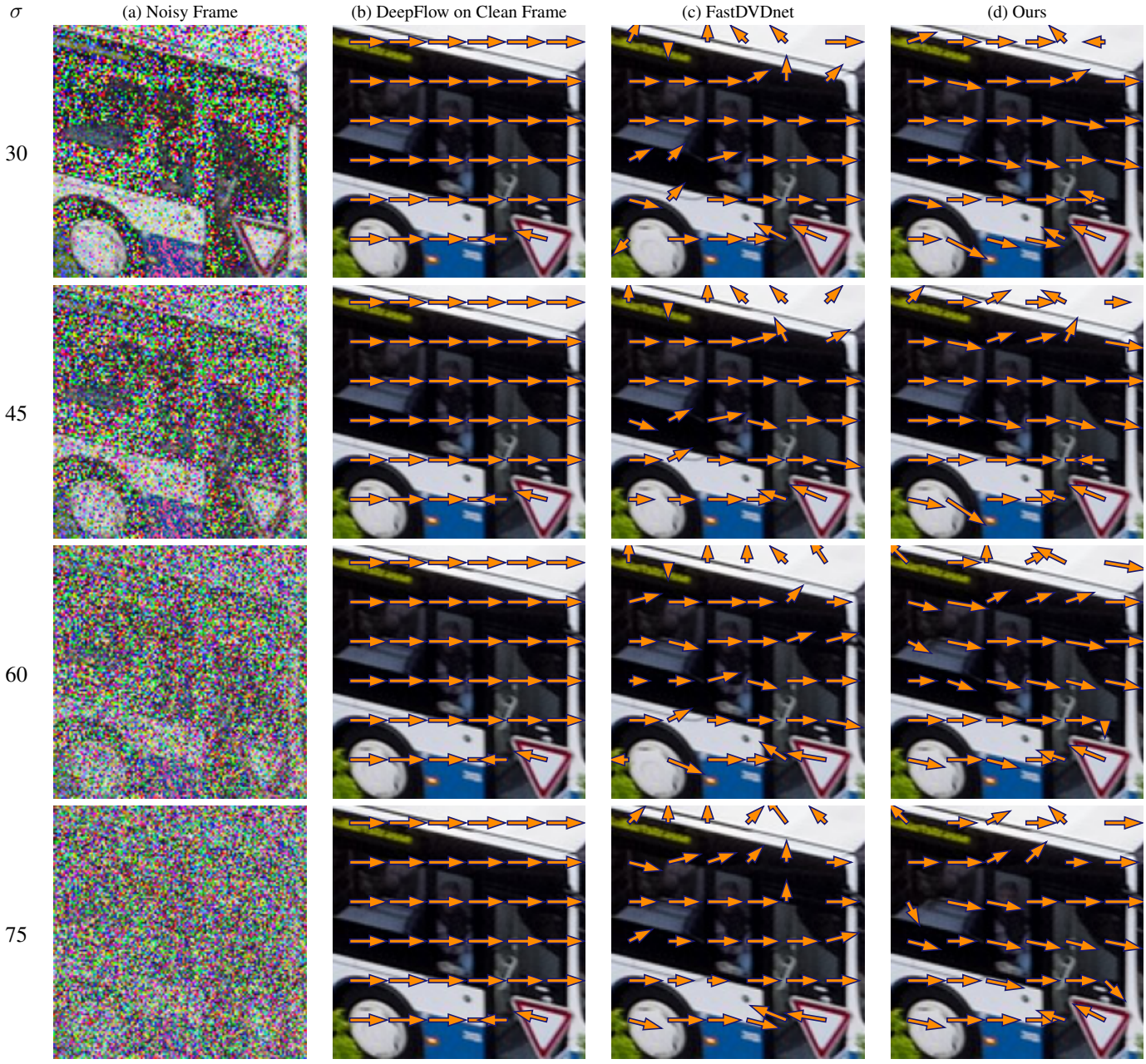


Figure 19. **CNNs trained for denoising automatically learn to perform motion estimation; `bus` video from the DAVIS dataset**. Motion estimated from the gradients of UDVD and FastDVDnet. See description of Figure 16.