Supplementary Material for 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.

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 [1] 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 [2]. 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 [2] 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 [3] 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 [2], 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

$$p(x|y, \Omega_y) \propto p(y|x)\ p(x|\Omega_y) \tag{1}$$

[*] equal contribution.

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 1: **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.

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,

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

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:

$$\mathcal{L}(\mu_x, \Sigma_x) = \frac{1}{2}[(y-\mu_x)^T(\Sigma_x + \sigma^2 I)^{-1}(y-\mu_x)] + \frac{1}{2}\log|\Sigma_x + \sigma^2 I|. \tag{3}$$

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 [2].

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

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 1. 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 [4] 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

	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 2: **Performance of UDVD**. Table shows the mean PSNR values of a state-of-the-art supervised video denoiser (FastDVDnet [4]) 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 $\sigma \in [5, 55]$.

for reconstruction with 4 and 96 intermediate output channels, as in [2]. 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 [5, 6, 2, 4, 7] (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 [8] 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 [2].

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 [2]. 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 1 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 2 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 1 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.

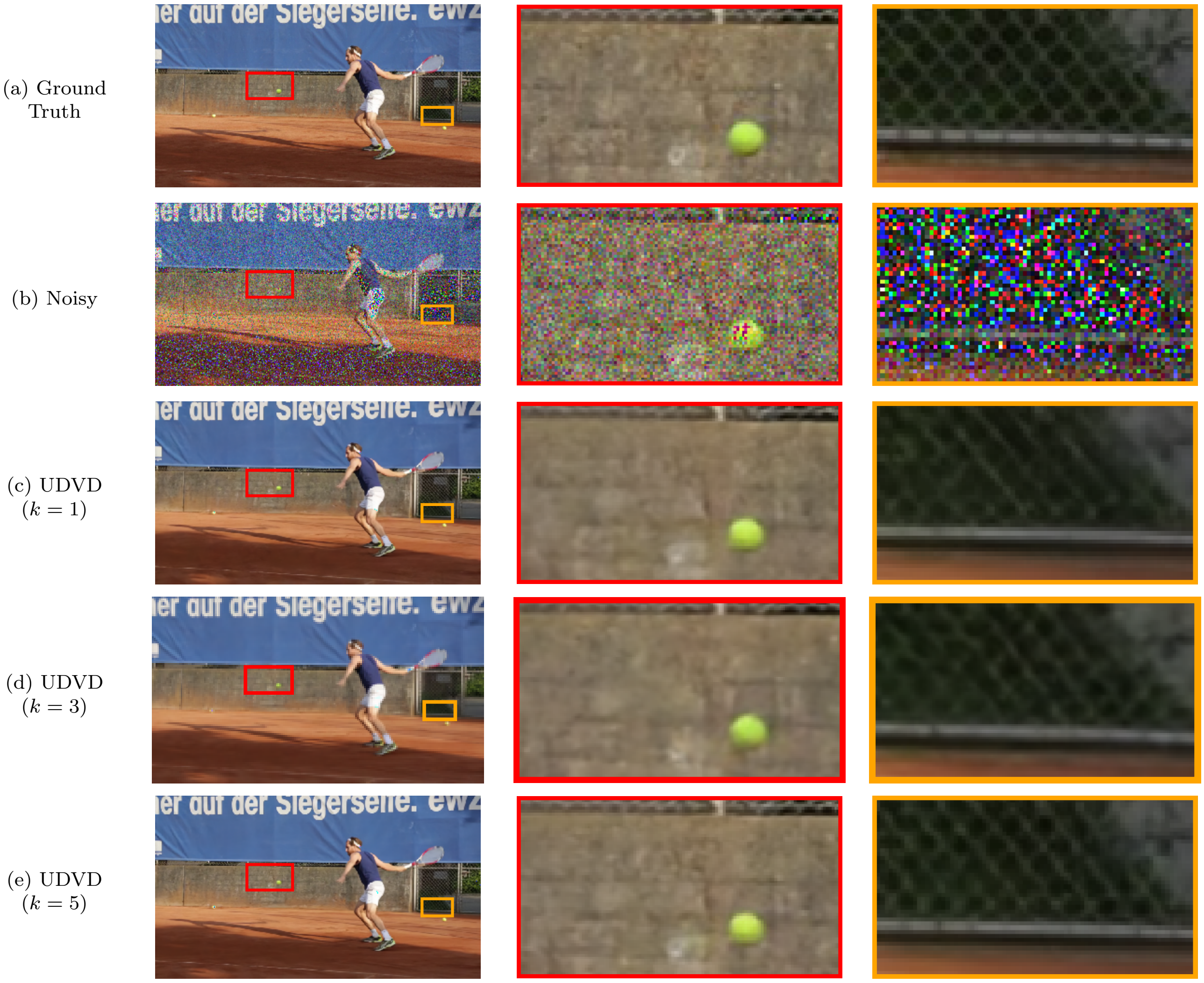


Figure 1: **Comparison of blind image and video denoising**. Example from the DAVIS dataset. (a) Ground truth frame. (b) Noisy frame. (c) Reconstruction using a single frame. The texture details of the brick wall and the fence are not recovered well. Reconstruction using (d) 3 and (e) 5 surrounding frames produces an improved texture estimate.

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 2 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[1] (`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 [9, 10, 11]. A popular burst denoising method, KPN [10] achieved a PSNR of 27.83 on the DAVIS dataset at $\sigma = 30$ [2], 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 [10], 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 [12].

D.1 Details of test sets.

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

1. **DAVIS** [**13**]: We take all the 30 sequences from the test set of the DAVIS Challenge 2017.

2. **Set8** [**4**]: Following FastDVDNet [4], 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 [12], 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 [4].

4. **Vid3oC** [**14**]: 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.

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

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

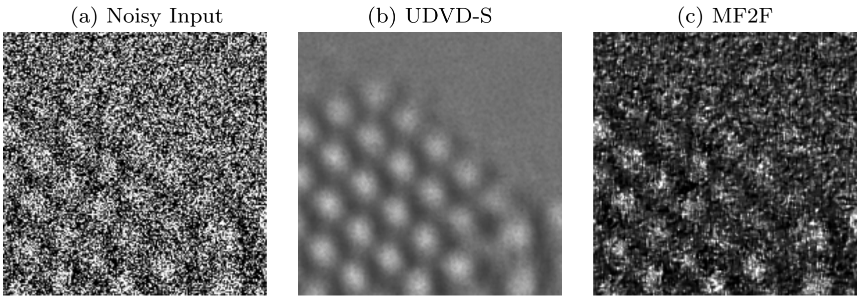
		$\sigma = 30$									
		ten-v	snow	hyper	raft	motor	trac	sunf	touch	park	**mean**
No. of frames		75	59	37	29	32	85	85	85	85	-
No Aug	(without ES)	33.37	29.10	29.72	27.26	27.28	32.52	35.07	32.65	30.20	30.80
No Aug	(with ES)	34.35	30.67	32.42	30.72	29.21	33.08	37.04	33.63	30.40	32.39
F	(without ES)	34.00	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 3: **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 [12] with all the 5 different fine-tuning scheme proposed in Ref. [12]. 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.

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 3 and 4 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 [12]. 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 [4], which is trained with supervised on a large dataset of natural videos(DAVIS [13]). 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 3 and 4 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 2), where it *fails*. We used the official implementation[3] for all the training schemes.


(a) Noisy Input

(b) UDVD-S

(c) MF2F

Figure 2: **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*.

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

		$\sigma = 90$									
		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 4: **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 [12] with all the 5 different fine-tuning scheme proposed in Ref. [12]. 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.

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 5.

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

Table 5: **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 ($\sigma = 90$).

E Denoising Results on Real-world Datasets

Raw videos: The estimated ground truth, noisy raw data [15], 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 [12], cannot be applied directly on this dataset as their pre-trained backbone expects an input in the RGB domain. In Ref. [12], the authors convert mosaiced videos into the RGB domain, apply MF2F [12] 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`).

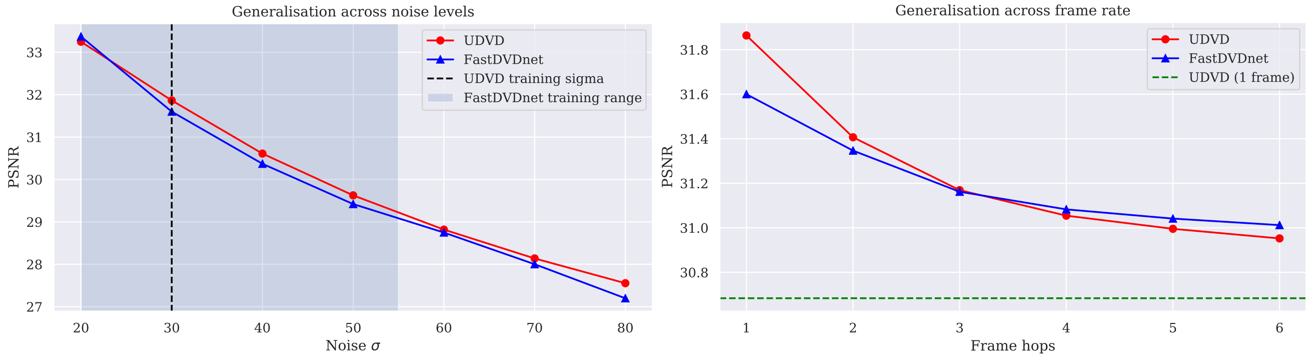


Figure 3: **Generalization across noise levels and frame rates.** (left) UDVD trained at only $\sigma = 30$ generalizes well to noise levels not seen during training. The plotted points represent mean PSNR values evaluated on Set8. (right) UDVD generalizes well to faster videos (created by skipping frames) and consistently outperforms a baseline image denoiser (UDVD with a single input frame, shown as a green dashed line).

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 [5, 4, 7]. 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 [6] 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 [3]) 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 3).

Generalization across frame rates: To test generalization across frame rates, we simulated faster videos by skipping frames of videos in Set8. Fig 3 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 5, 6, 7 and 8 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 9)

Contribution of neighbouring frames for denoising: UDVD tends to ignore temporally distant frames at lower noise levels as shown in Fig 5, 6, 7 and 8. This phenomenon is quantified in Fig 4 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 9.

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 5, 6, 7 and 8) and they approximately sum up to one (see Fig 4).

G.2 Real-world Data

Equivalent filters for the raw video, the fluorescence-microscopy and the electron-microscopy data are shown in Fig 10. 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 4). In the electron-microscopy dataset the weighting functions shows that the network relies on adjacent frames to estimate the denoised (as quantified in Fig 4).

G.3 Motion Estimation

Figures 5, 6, 7 and 8 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 11, 12, 13 and 14 show additional examples for UDVD and FastDVDnet. The estimated optical flow is mostly consistent with the estimated obtained by DeepFlow [16] 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.

References

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

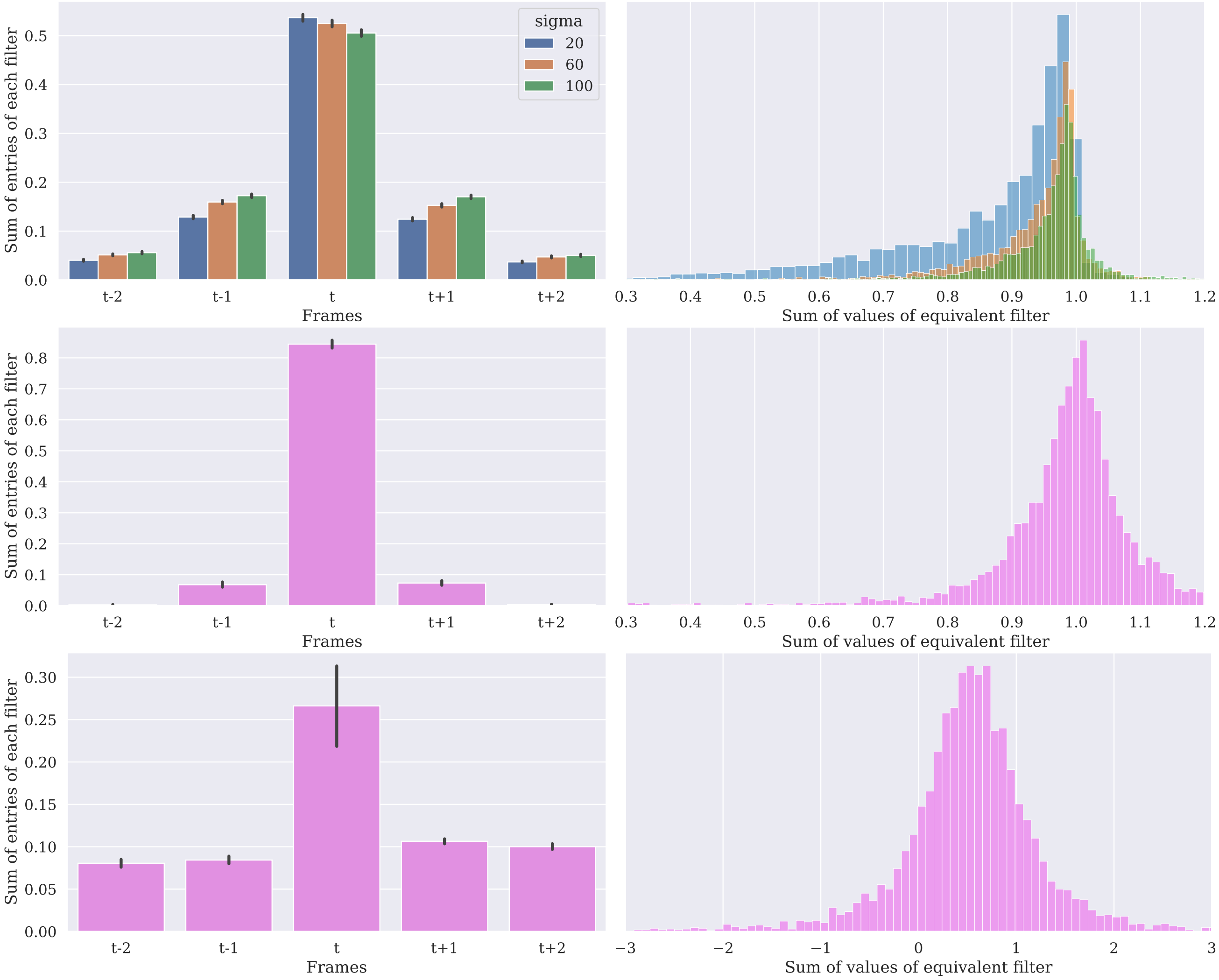


Figure 4: **Quantitative analysis of equivalent filters**. *Left column:* The graphs show the sum of the entries of the equivalent filters in each frame, averaged over 5000 pixels from 250 random patches of size 128 × 128. For all datasets, the central frame dominates. For the DAVIS dataset (top), the 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).

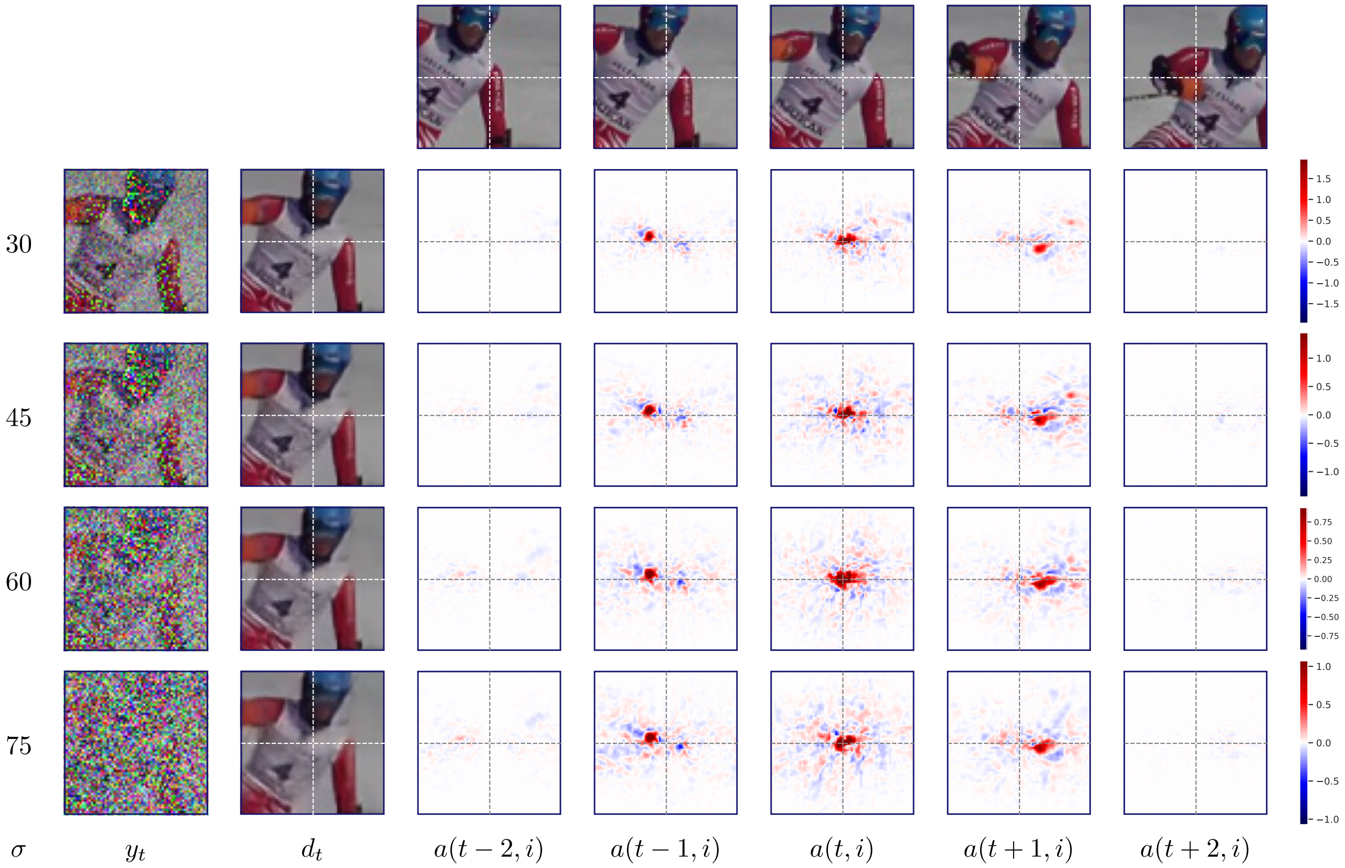


Figure 5: **Video denoising as spatiotemporal adaptive filtering; `giant-slalom` video from the DAVIS dataset**. Visualization of the linear weighting functions ($a(k,i)$, Section 6 of paper) of UDVD. The left two columns show the noisy frame y_t at four levels of noise, and the corresponding denoised 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.

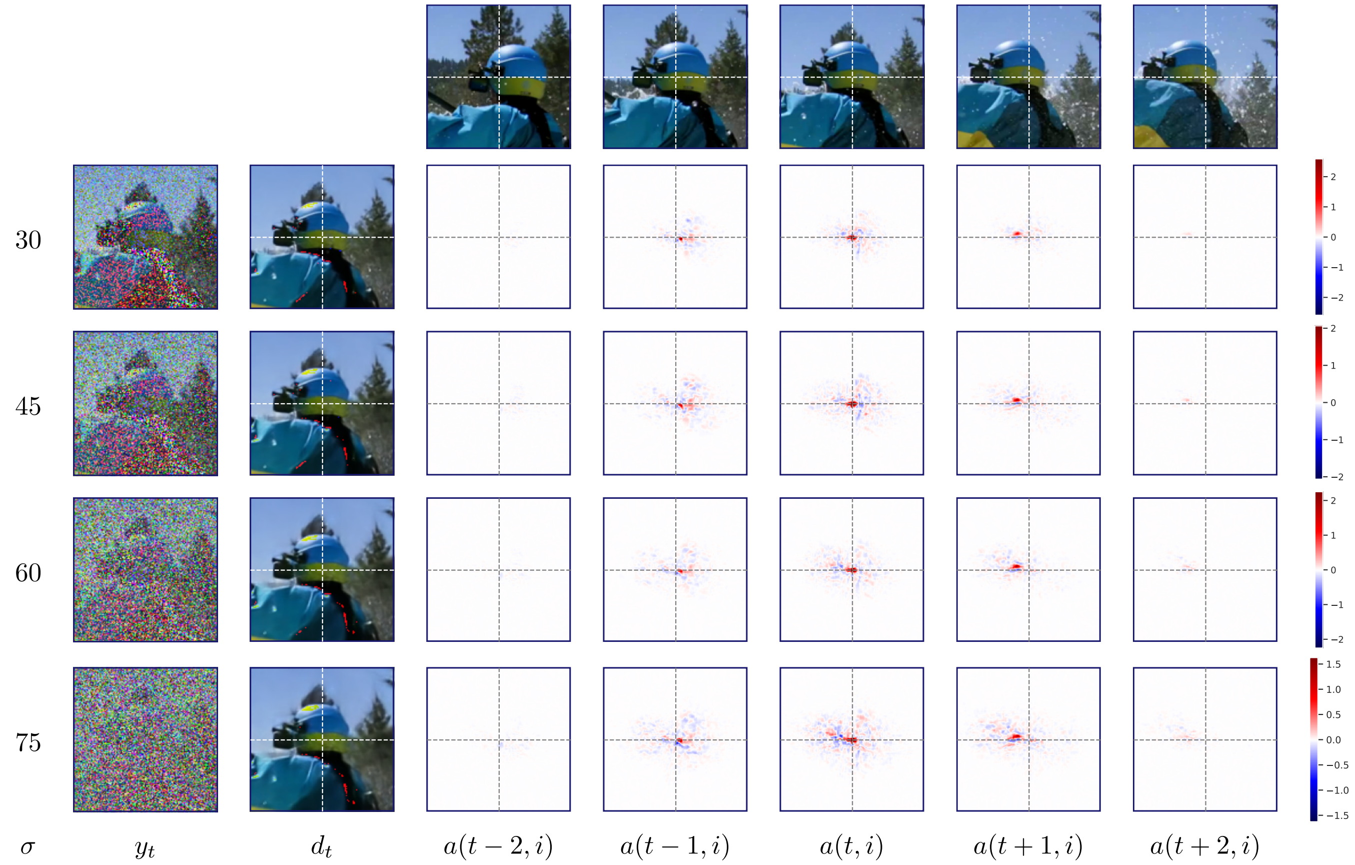


Figure 6: **Video denoising as spatiotemporal adaptive filtering; `rafting` video from the GoPro dataset**. Visualization of the equivalent filters, as described in Fig 5.

[15] 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.

[16] 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.

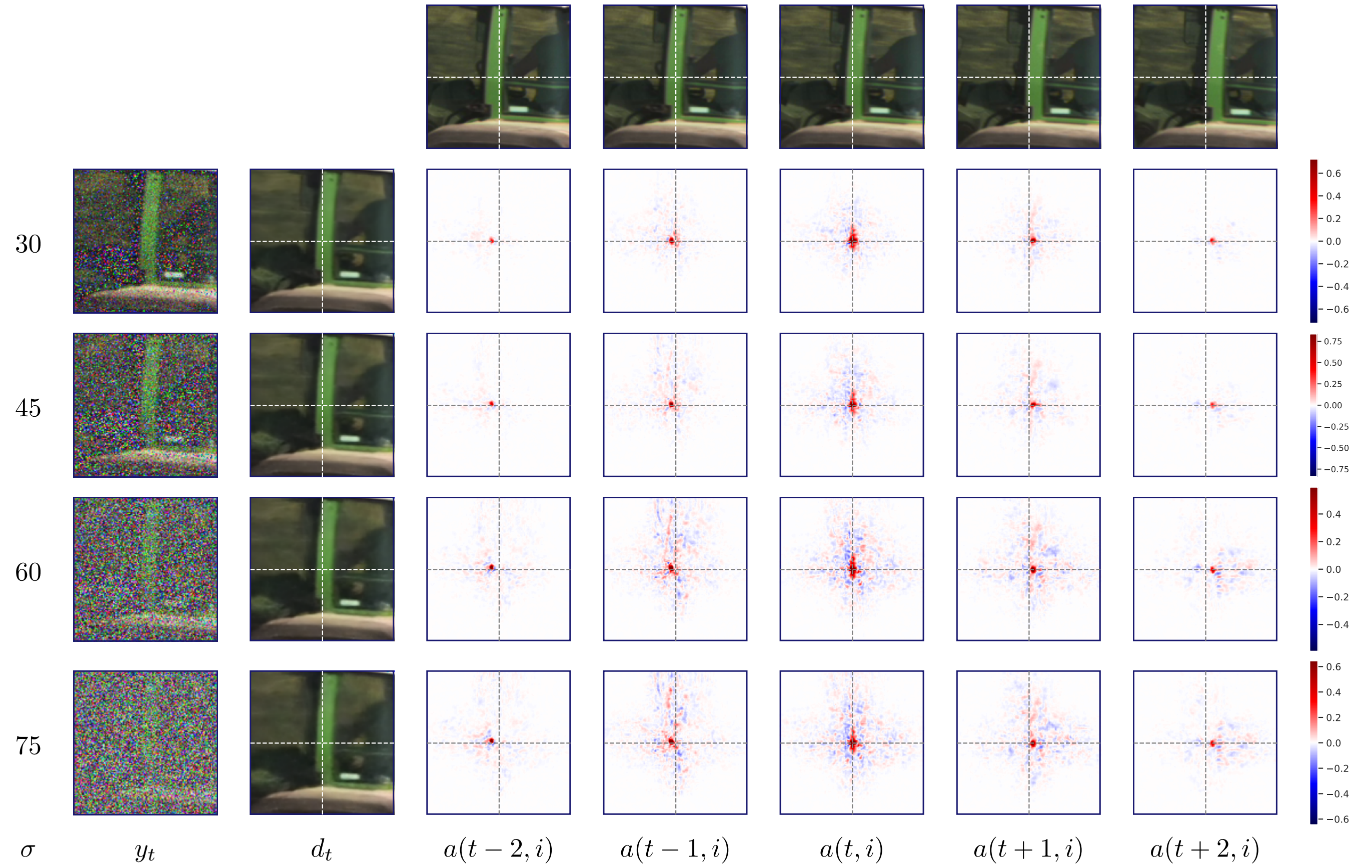


Figure 7: **Video denoising as spatiotemporal adaptive filtering; `tractor` video from Set8**. Visualization of the equivalent filters, as described in Fig 5.

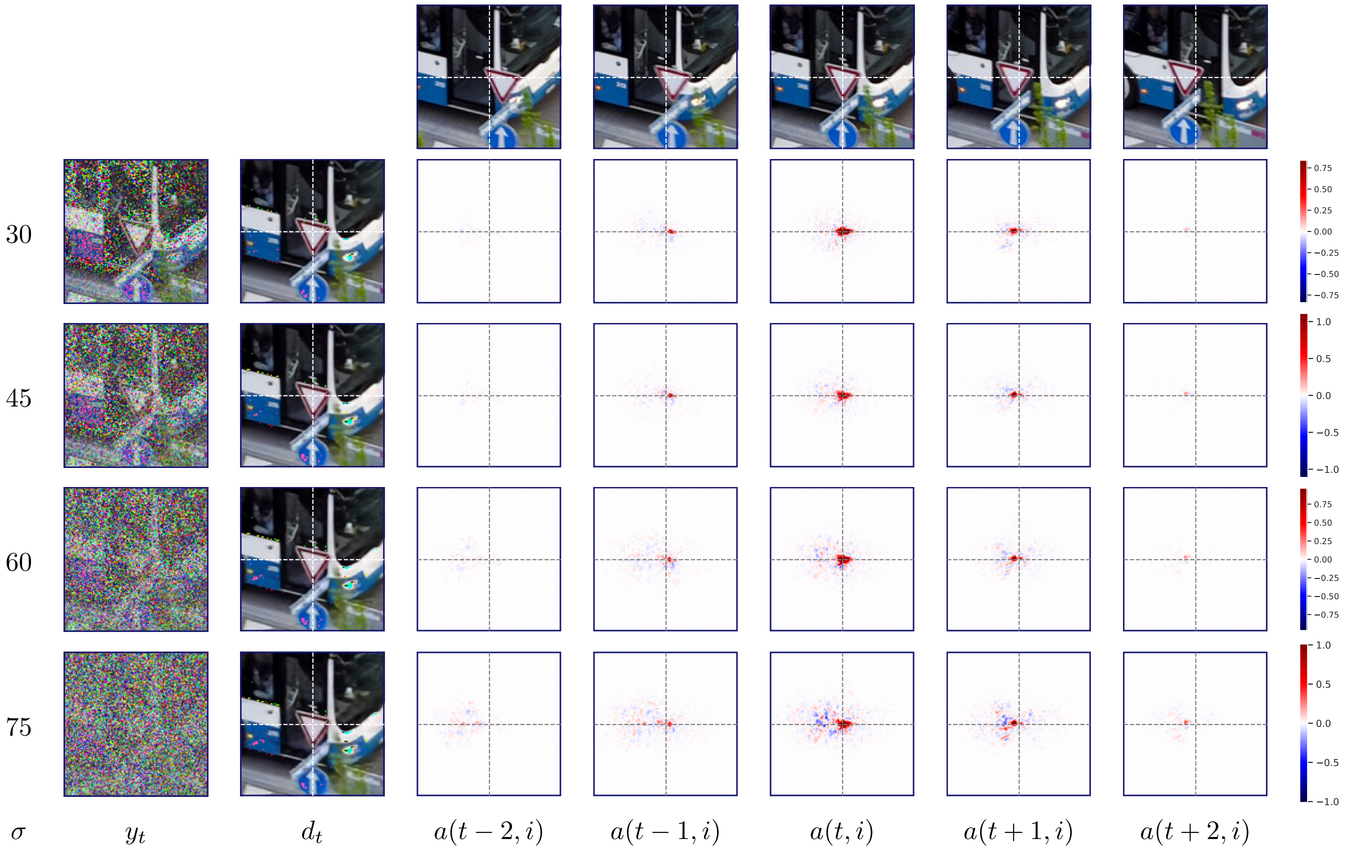


Figure 8: **Video denoising as spatiotemporal adaptive filtering;** `bus` **video from the DAVIS dataset**. Visualization of the equivalent filters, as described in Fig 5.

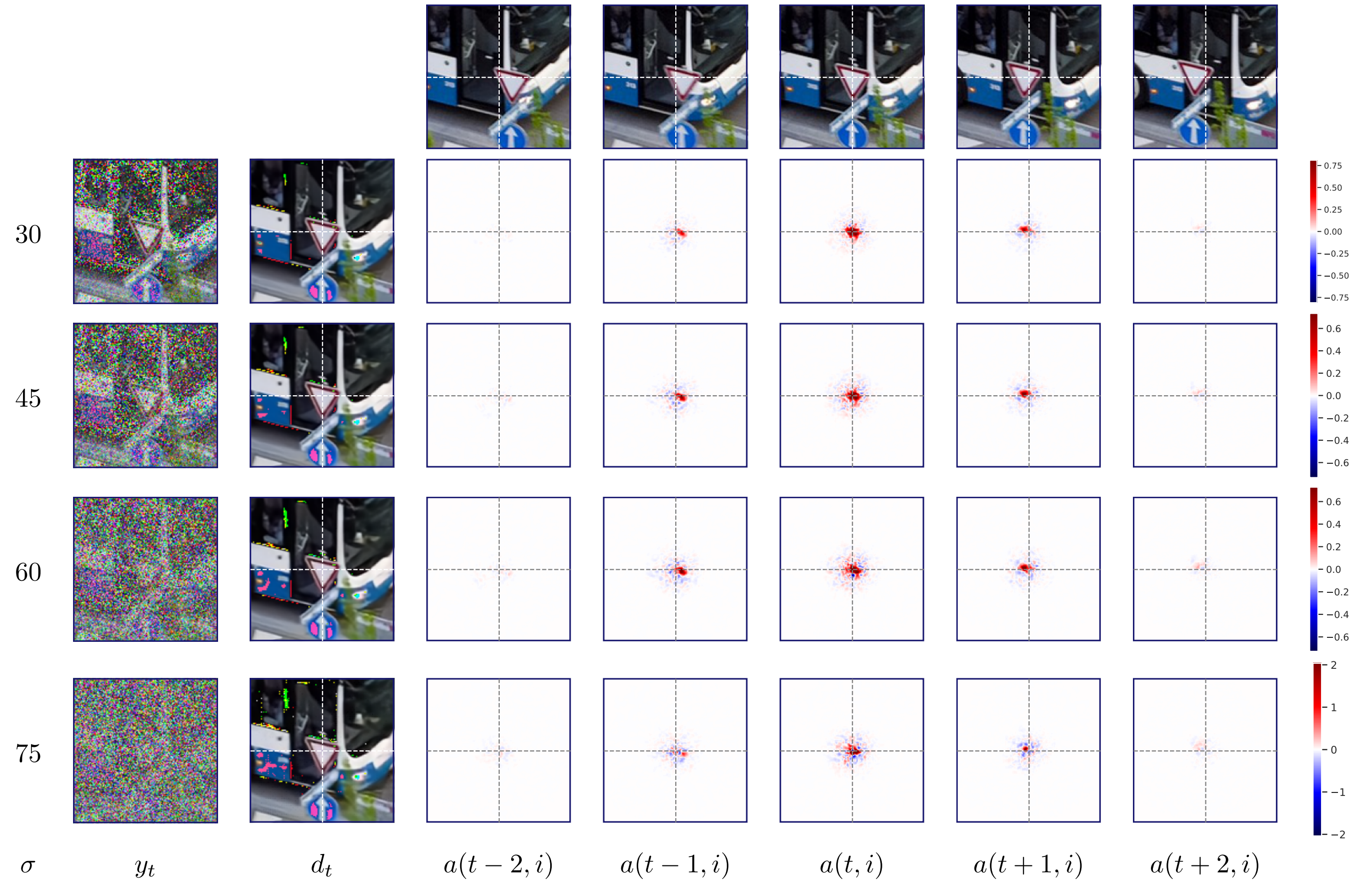


Figure 9: **Video denoising using FastDVDnet as spatiotemporal adaptive filtering; bus video from the DAVIS dataset**. Visualization of the linear weighting functions ($a(k, i)$, Section 6 of paper) of FastDVDnet which is trained with supervision. The left two columns show the noisy frame y_t at four 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 8.

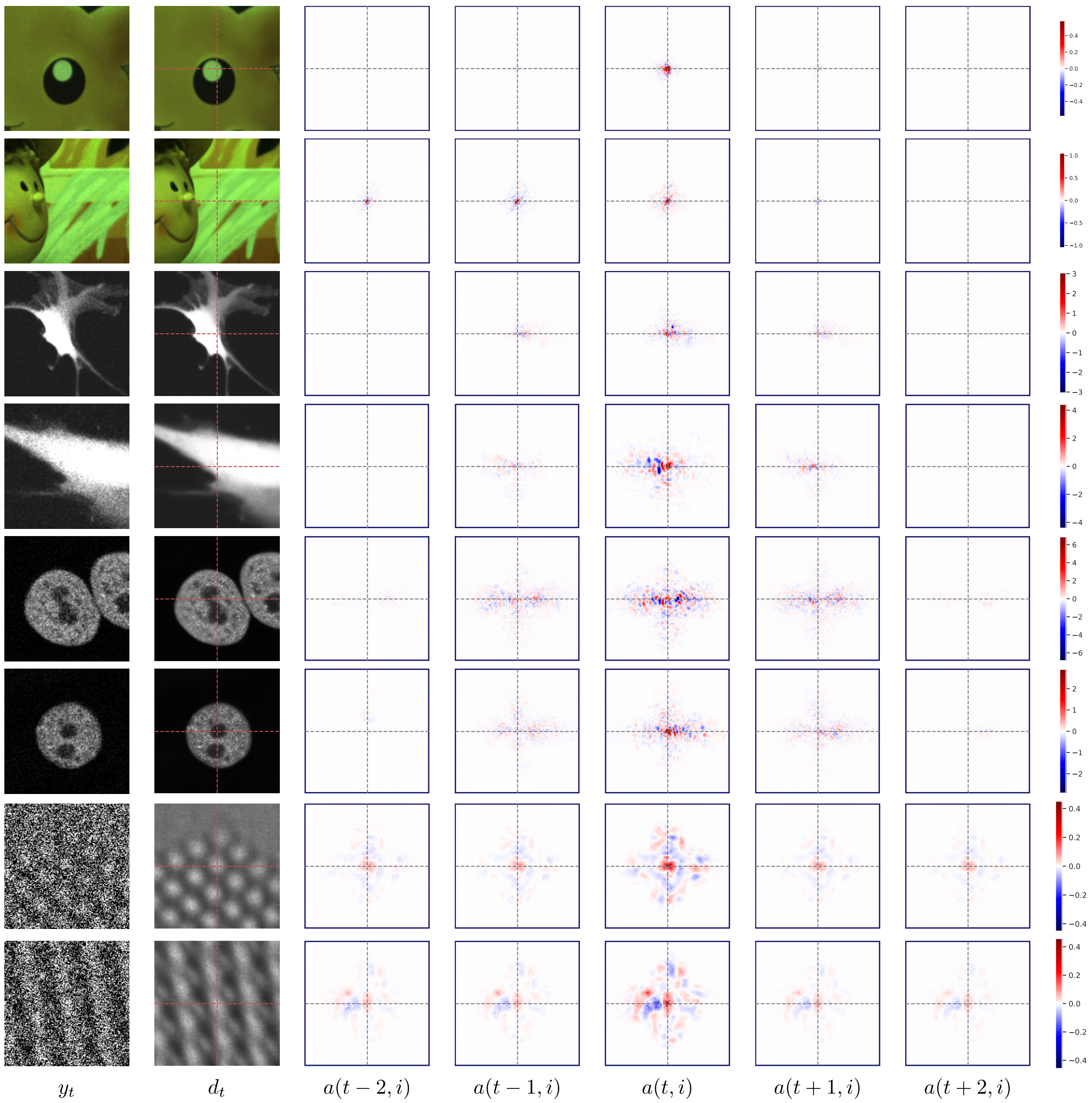


Figure 10: **Equivalent filters of UDVD when applied to real-world data**. Visualization of the linear weighting functions ($a(k,i)$, Section 6 of paper) of UDVD trained to denoise raw video, fluorescence and electron microscopy data. The left two columns show the noisy frame y_t and the corresponding 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 4).

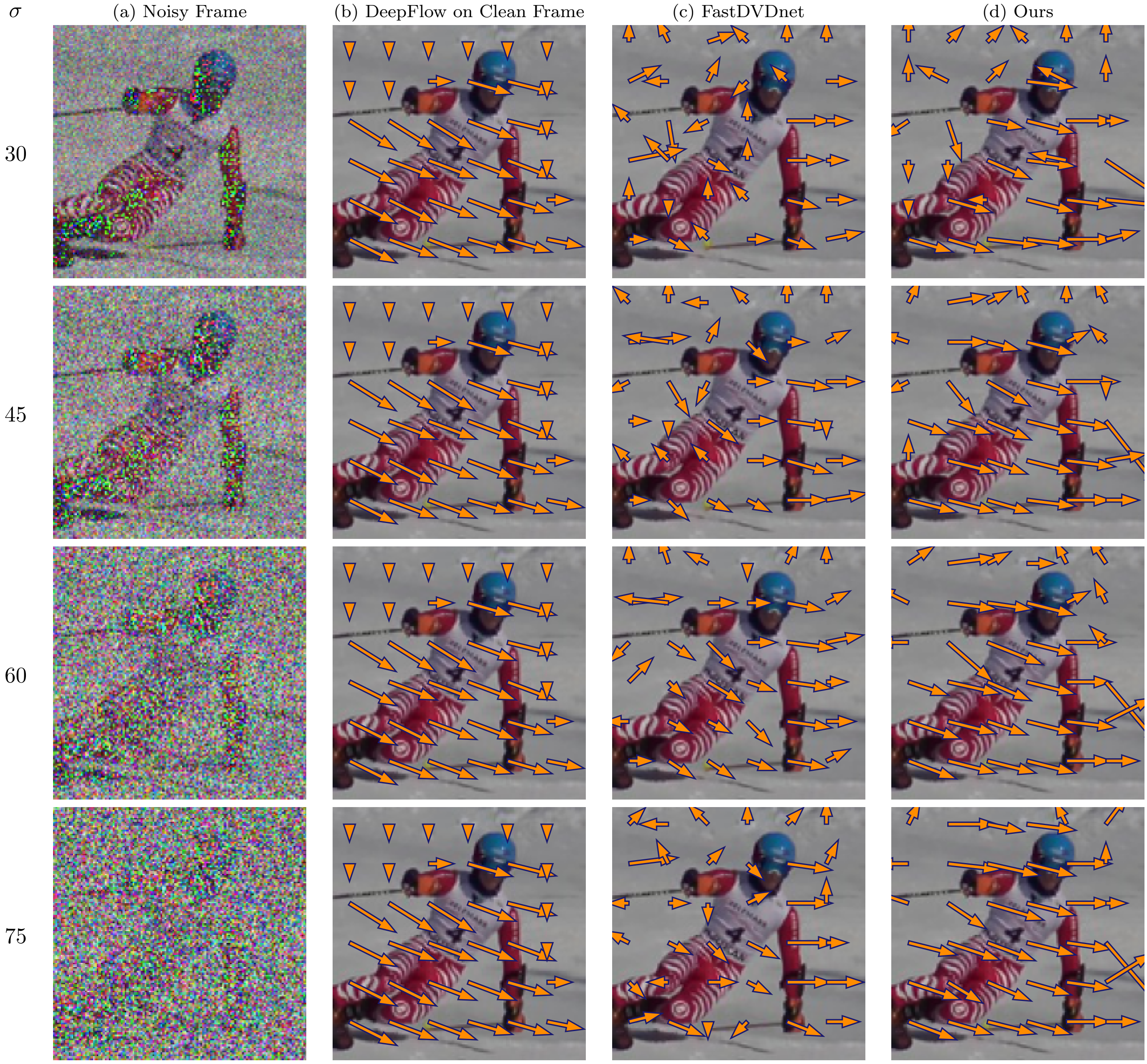


Figure 11: **CNNs trained for denoising automatically learn to perform motion estimation**. (a) Noisy frame from `giant-slalom` video in the DAVIS dataset. (b) Optical flow direction at multiple locations of the image obtained using a state-of-the-art algorithm applied *to the clean video*. Optical flow 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.

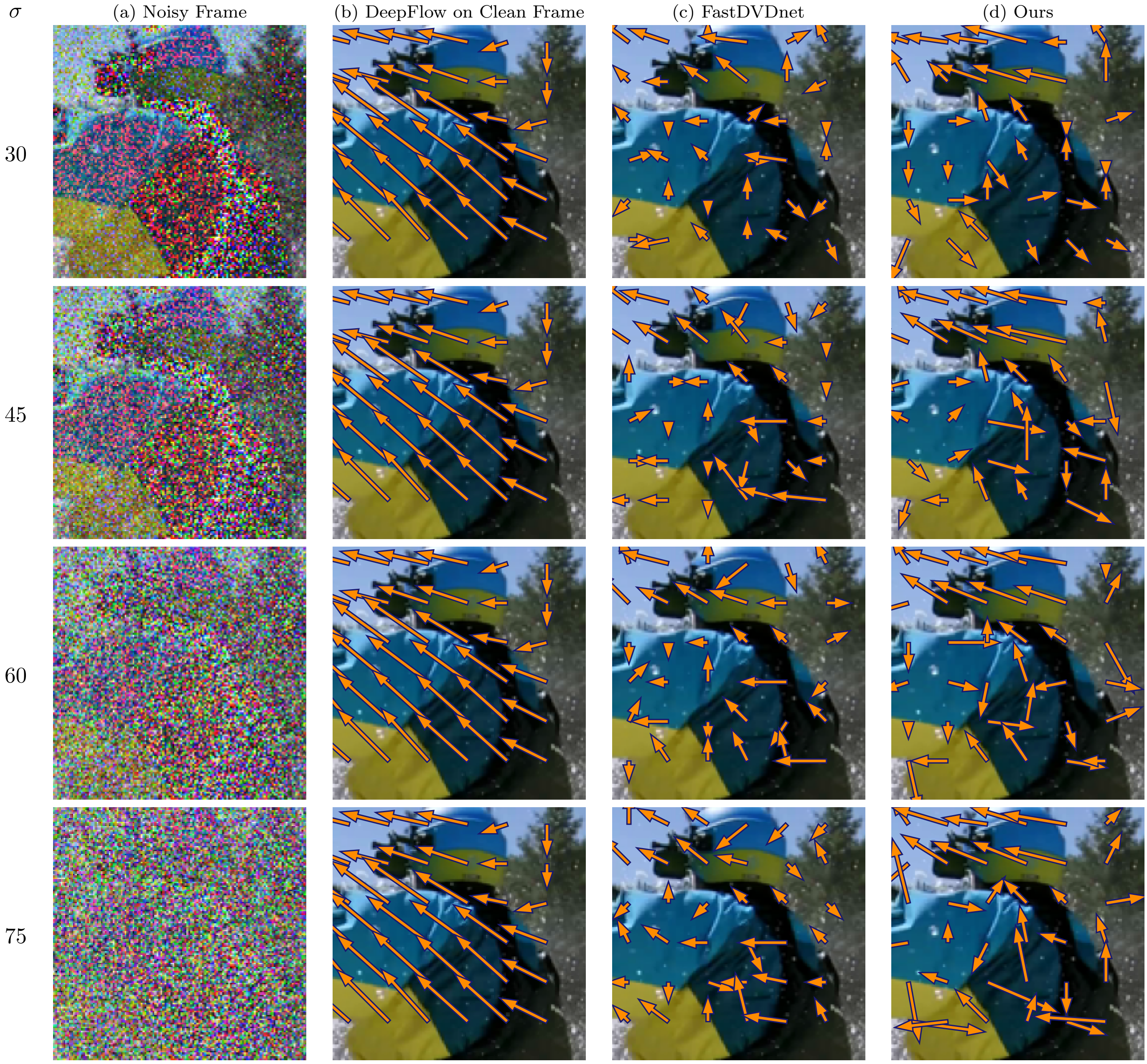


Figure 12: **CNNs trained for denoising automatically learn to perform motion estimation;** `rafting` **video from Set8**. Motion estimated from the gradients of UDVD and FastDVDnet. See description of Figure 11.

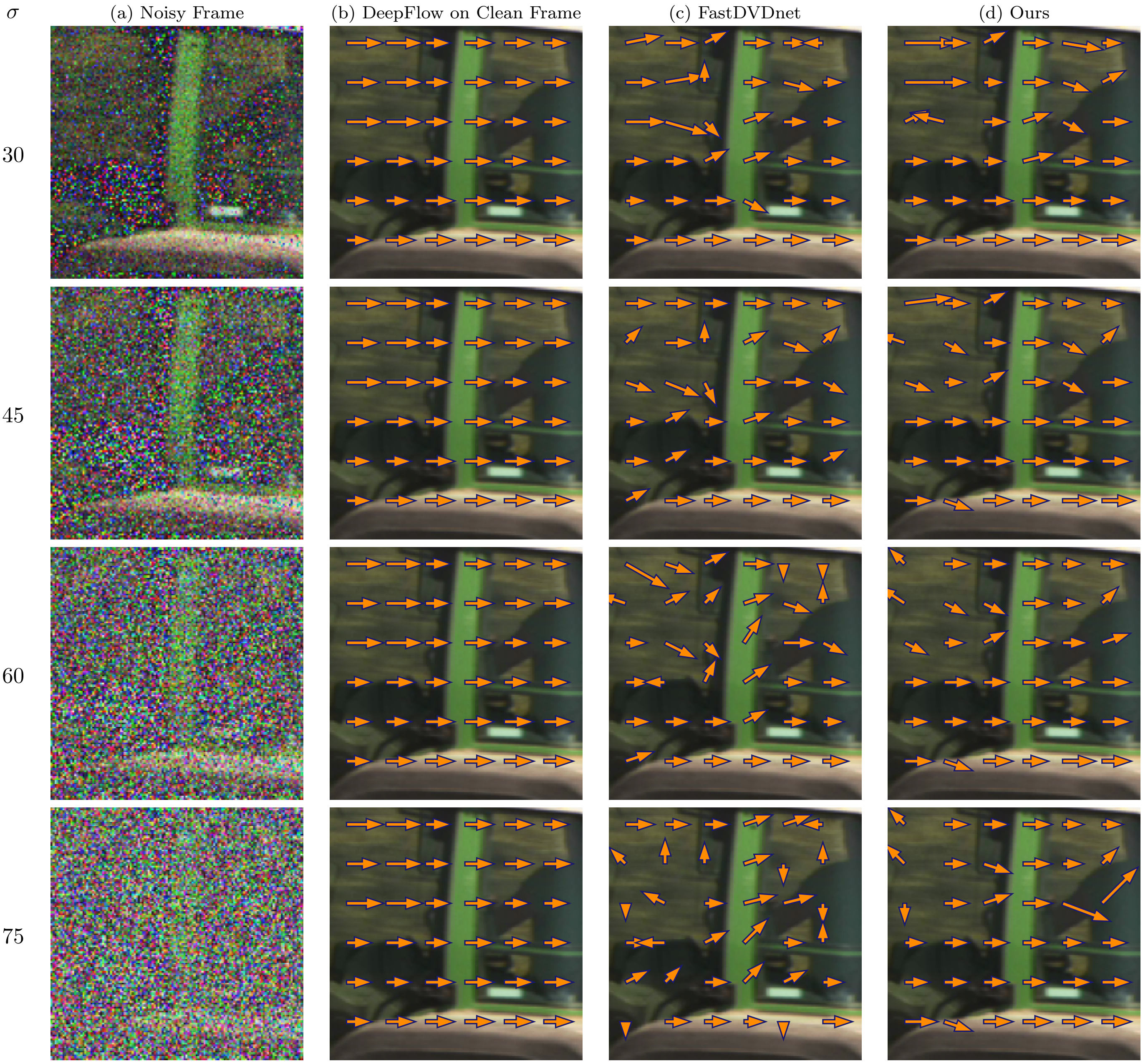


Figure 13: **CNNs trained for denoising automatically learn to perform motion estimation;** `tractor` **video from Set8**. Motion estimated from the gradients of UDVD and FastDVDnet. See description of Figure 11.

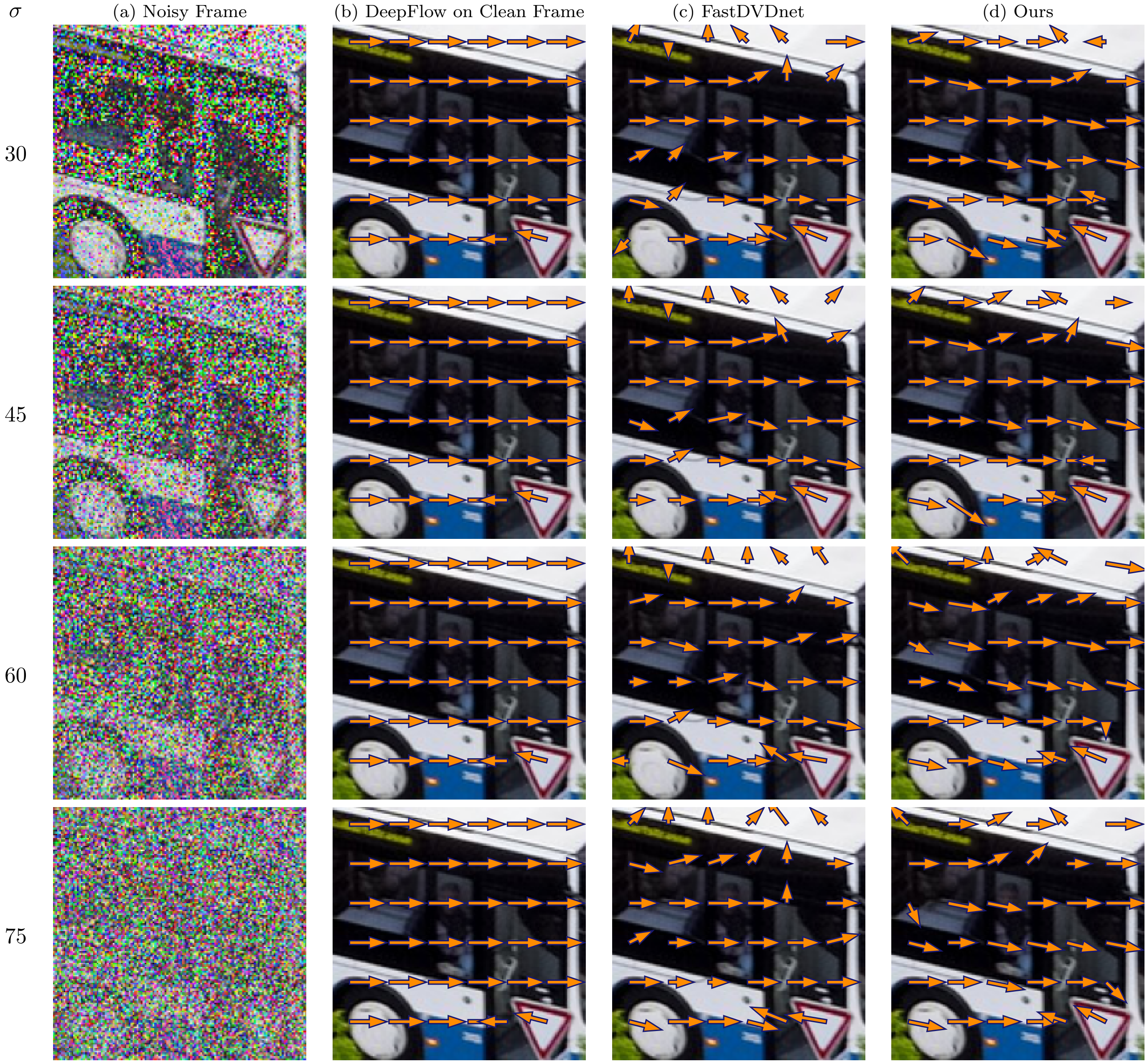


Figure 14: **CNNs trained for denoising automatically learn to perform motion estimation;** `bus` **video from the DAVIS dataset**. Motion estimated from the gradients of UDVD and FastDVDnet. See description of Figure 11.