Towards using Diffusion Models for Compressive Video Recovery from Coded Exposures

Daniel Saragih

Abstract—Coded exposures capture multiple frames of video within a single measurement image enabling low-speed cameras to capture information from high-speed video. The problem of video recovery from coded exposures involves reconstructing the constituent frames (henceforth, subframes) of the coded image, which is often an ill-posed inverse problem: coded exposures modulate high-speed frames through masks which are then summed together. In this report, we consider using diffusion generative models as the reconstruction algorithm to infuse strong priors on the subframes, while experimenting with baselines from a related problem in snapshot compressive imaging (SCI). In particular, we focus on the method of diffusion posterior sampling (DPS) to move the output subframes towards an image subspace that may correspond to the coded image. DPS works by a measurement loss on top of a pre-trained image diffusion model, and hence it does not require extra training for inference. Extensive results on common video datasets are provided to compare against baselines such as STFormer and a specialized 3D-UNet, which show there is still room for improvement in terms of reconstruction quality and efficiency.

Index Terms—Computational Imaging, Diffusion Models, Video Reconstruction

1 Introduction

Computational imaging provides new ways to capture high-speed images in a memory-efficient manner. Coded exposures or video snapshot compressive imaging (SCI) is one such computational solution, shown in Fig. 1, which modulates high-speed images with different masks and integrates the result to produce the coded image as measurement. This allows for efficient compression during imaging and efficient storage of high-speed data. Over the years, many systems of this kind have been built [1], [2], [3], and they all include a software component that decodes the measurement and mask back into the high-speed video.

This latter decoding module, which tackles the problem of video recovery from coded exposures is an ill-posed inverse imaging problem. Therefore, past methods [1] have incorporated prior knowledge about natural images to constrain the feasible generation space. It is still an open problem to find the best method to incorporate this knowledge. In recent years, following the explosion of deep learning tools, learning-based models have gained popularity [4] due to their notable ability to generalize. However, learning-based methods are often restricted to their training regime, and lack the flexibility to accommodate different coding methods without re-training.

In this report, we explore the use of diffusion models for video compressive sensing. As we shall see later, there is an extensive literature on using diffusion models for inverse imaging problems. For one, such methods often require no extra training, building off pre-trained models on large datasets. Moreover, this approach slots right into the SCI framework, being the reconstruction algorithm in Fig. 1. We also experiment with prior methods from SCI and video compressive sensing such as STFormer [5] and a UNet-based framework [6] for our camera setup [2], [3]. Towards a more robust evaluation of their abilities, we train a Generative Adversarial Network (GAN) to simulate camera measurement noise, and perform extensive experiments on common benchmarks to obtain quantitative and qualitative comparisons.

• *D. Saragih is with the Department of Computer Science, University of Toronto, Toronto, ON.*
E-mail: daniel.saragih@mail.utoronto.ca

1.1 Contributions

In particular, we make the following contributions:

- We build a GAN for generating camera noise based on captures from a new coded two-bucket camera.
- We build a reconstruction algorithm based on diffusion models for the purpose of video recovery from coded exposures.
- We adapt methods in [5], [6] to our camera setup for video compressive sensing.
- We obtain experimental results on two video datasets that demonstrate the reconstruction abilities and inference speed of our methods.

2 Related Work

Video compressive sensing is closely related to SCI [4], and is more generally an ill-posed inverse problem in imaging. In this section, we briefly introduce related works on sensor simulation, followed by a review of learned approaches for similar inverse problems.

2.1 Simulation of Camera Measurements

Much work has been done on optimizing high dynamic range captures [7], which often wrestles with the problem of measurement noise. To investigate the noise, it is often convenient to simulate synthetic captures by emulating measurement noise [7]. As precise modeling of each image

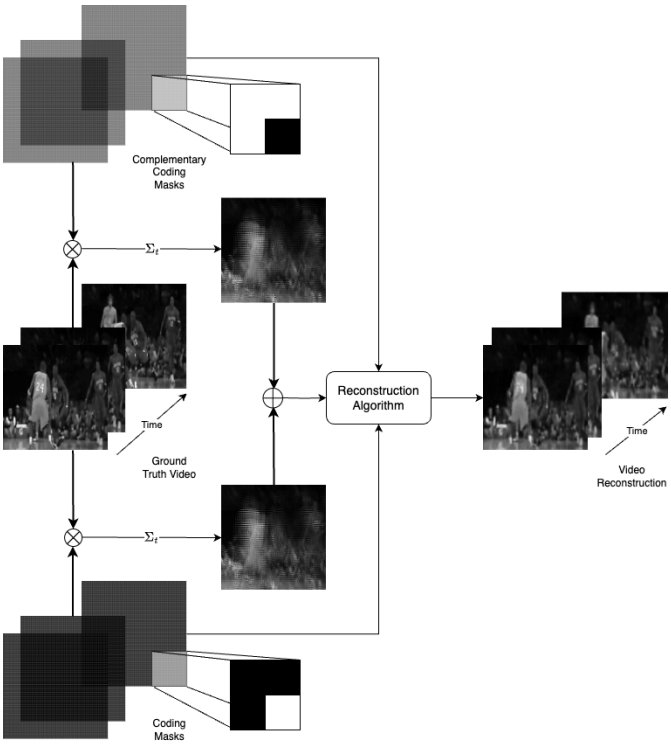

Fig. 1: Schematic diagram of the coding process of a two-bucket sensor [2], [3]. A series of ground truth frames $\mathbf{X} \in \mathbb{R}^{H\times W}$ are modulated through a mask video and its complement (here $k = 2$ and $i = 4$ is magnified) to produce two concatenated measurements (coded images) $\mathbf{Y} \in \mathbb{R}^{2\times H\times W}$. Then, the measurement and masks are passed into the reconstruction algorithm to produce the video frames.

processing step is challenging, machine learning techniques have been used for the task of inverting each step of the image processing pipeline, enabling realistic measurements of raw sensor data [8]. Recent advances in generative models, such as generative adversarial networks (GAN) [9], provide a powerful framework to generate data across many different media, and have recently been used for sensor simulation [10]. We build on this most recent work to simulate camera noise for realistic video recovery.

2.2 Learned Baselines for Video Reconstruction

The problem of video compressive sensing is quite similar to that of SCI. The latter has an extensive list of approaches such as OCD [1], STFormer [5], BIRNAT [11], AAUN [12], and GAP-Net [13]. Most related to ours is the unified framework [6] which works on images from the C2B setup, which is similar to our camera setup. As SCI and video reconstruction are ill-posed inverse problems, it is crucial that these methods incorporate image priors through their training. Moreover, an important prior for realistic generation is temporal consistency, hence all of the above account for both spatial and temporal information, which we found to yield a significant performance improvement.

More recently, the rise of foundational image and video diffusion models [14], [15], [16] motivates their use in video reconstruction, as they infuse strong natural image priors. Indeed, a few recent works have explored their usage in SCI [17], [18]. We take a step back from training a diffusion model for this task, and instead use conditioning methods such as diffusion posterior sampling (DPS) [19], [20].

Of course, DPS is not the only inverse problem solver that uses diffusion models. Follow up papers [20], [21], [22] have all shown improvements, but we shall stick to DPS as it is the progenitor approach and is simple to integrate. We should also note that the idea of using the measurement to preempt the reconstruction algorithm is a common idea [5], [23], even without diffusion models. However, these implementations lack the theoretical justifications developed in DPS [19] and trades off reconstruction accuracy with considerations such as model size and inference speed [23].

3 Proposed Method

Inspired by the success of DPS and kin methods in SCI, we apply these approaches to video reconstruction for coded exposures with a coded two-bucket setup [2]. In this section, we introduce a mathematical model of our video recovery problem based on [5], briefly describe how we simulate camera measurements, and our approach with DPS.

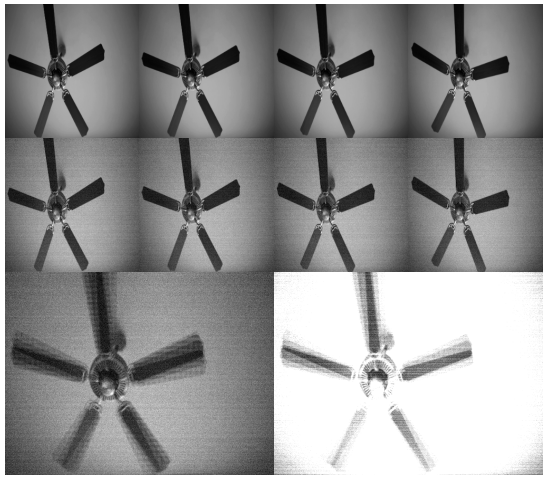

Fig. 2: GAN-based measurement noise simulations on video of a spinning fan. The last row is $\mathbf{Y} \in \mathbb{R}^{2\times H\times W}$ from a two-bucket exposure (left: bucket-0, right: bucket-1) following (3) with $F = 4$.

3.1 A Model for Video Compressive Sensing

The video compressive technique we consider encodes high-dimensional video data into a single 2D image. Specifically, we work with the coded two-bucket setup (C2B) [2] from [3] which produces two measurements. As shown in Fig. 1, the three dimensional video data is modulated by structured masks, and then summed together across the time domain to produce two images which are then concatenated along the first axis.

As our experiments are restricted to grayscale images, we may simplify notation and let $\mathbf{X} \in \mathbb{R}^{F\times H\times W}$ denote a F-frame video, which conceptually is our ground truth. Next, let $\mathbf{M} \in \mathbb{R}^{F\times H\times W}$ denote the masks to be used. For our specific problem, we restrict the universe of masks to

that defined below. Indeed, in SCI, often a Bernoulli random mask is used [11], [24]. In our case, for $k \in \mathbb{N}$ and $i \in [F]$, define

$$\mathbf{M}_k[i,y,x] = \begin{cases} 1 & \text{if } (y = \lfloor i/k \rfloor \bmod k, \\ & \quad x = (i \bmod k) \bmod k) \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where we one-index image pixels. To simplify things, we only work with k even, $F = 0 \mod k^2$, and $H = W = 0 \mod k$. Intuitively, first consider a $k \times k$ tile, then the i-th frame has a 1 on the i-th pixel (mod k^2) by reading order and 0 elsewhere; repeat this pattern to tile the entire image. In our case, we always set $\mathbf{M} = \mathbf{M}_k$ for some fixed $k \in \mathbb{N}$, and we often drop this subscript.

First, let us consider bucket-0. For each frame within the video $f \in [F]$, it is modulated by a mask as follows:

$$\mathbf{X}'[f] = \mathbf{X}[f] \odot \mathbf{M}[f]. \tag{2}$$

We then compress across time:

$$\mathbf{Y}[f] = \sum_{f=1}^{F} \mathbf{X}'[f] + \mathbf{Z} \tag{3}$$

where $\mathbf{Z} \in \mathbb{R}^{H \times W}$ denotes measurement noise.

It is often convenient to work in terms of vectors, so we may vectorize $\mathbf{x} = [\mathbf{x}_1^\top, \dots, \mathbf{x}_F^\top]^\top \in \mathbb{R}^{HWF}$ where $\mathbf{x}_f = \text{vec}(\mathbf{X}[f]) \in \mathbb{R}^{HW}$. By linearity of the operations in (2) and (3), we may write a vectorized version:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z} \tag{4}$$

where $\mathbf{H} \in \mathbb{R}^{HW \times HWF}$. The objective is thus: given the coded image $\mathbf{y}$ and the measurement operator $\mathbf{H}$, solve for the subframes $\mathbf{x}$.

The measurements produced by bucket-1 are the same except that we use $1 - \mathbf{M}[f]$ instead to produce another $\tilde{\mathbf{Y}} \in \mathbb{R}^{H \times W}$ as in (3). We may then concatenate the two measurements to form an image in $\mathbb{R}^{2 \times H \times W}$.

3.1.1 GAN-based simulation of camera noise

In our case, $\mathbf{Z}$ is the camera noise generated using spectral normalization [25] to augment a GAN, following a previous implementation[1]. The training data for this process consists of images of a monochrome surface, and black and saturated images, captured with 50 different exposure times. Using a parameterized noise model [7], [10], we may explicitly calculate the noise parameters for all 50 exposure settings; the GAN is then trained to generate these parameters for new images, as shown in Fig. 2.

3.2 Video compressive sensing with DPS

To reconstruct subframes, we use the pre-trained diffusion model from the original paper [19] which aims to nudge pre-trained diffusion models towards generating the subframes $\mathbf{x}$. More concretely, we wish to push generations towards the subspace $\{\mathbf{x} \mid \mathbf{y} = \mathbf{H}\mathbf{x}\}$, which as shown in DPS [19], can be done in expectation by the conditional score approximation

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid y) \simeq \mathbf{s}_\theta(\mathbf{x}_t, t) - \rho \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_0\|_2^2, \tag{5}$$

1. https://github.com/zachsalehe/t6_simulation

where $\mathbf{x}_t$ is the intermediate diffusion step at time $t \in [0, T]$, $\hat{\mathbf{x}}_0$ is the predicted reconstruction, and $\mathbf{s}_\theta$ is the unconditional score. It is worth noting that the model does not contain a mechanism for temporal consistency and so the batch of images are entirely independent of each other. Moreover, we only use the bucket-0 measurements, hence $\mathbf{Y} \in \mathbb{R}^{H \times W}$.

In addition, we implement a variation that involves pixel reshuffling which is specific to the family of masks defined above. To explain the process, we look at an example of $k = 4$ and $H = W = 256$. In this case, we partition the image into 4×4 tiles; for $\mathbf{X}'[1]$, we have the original pixel for all top-left-corner pixels in each tile while the rest of the image is black. Hence, we may pick out these original pixels from $\mathbf{X}'[1]$. As $256/4 = 64$, the resultant image is 64×64. We can do this for all $f \in [16]$ to produce $\mathbf{W} \in \mathbb{R}^{16 \times 64 \times 64}$. Conceptually, $\mathbf{W}$ is like the ground truth video $\mathbf{X}$ but downsampled. Hence, this affords us another guidance vector. In particular, if we let $\mathbf{D}_k \in \mathbb{R}^{HWF/k^2 \times HWF}$ be the k-factor downsampling operator, we modify (5) to be

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid y) \simeq \mathbf{s}_\theta(\mathbf{x}_t, t) - \rho_1 \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_0\|_2^2 - \rho_2 \nabla_{\mathbf{x}_t} \|\mathbf{w} - \mathbf{D}_k\hat{\mathbf{x}}_0\|_2^2, \tag{6}$$

where $\mathbf{w}$ is the vectorized version of $\mathbf{W}$.

4 Experimental Results

In this section, we compare the performance of the aforementioned approaches. The peak signal-to-noise ratio (PSNR), the structured similarity index metrics (SSIM) [26], and learned perceptual image patch similarity (LPIPS) [27] are used to evaluate the performance of the reconstruction methods on two video datasets.

4.1 Datasets

We use *DAVIS2017* [28] as our training dataset, which contains 90 different scenes at 480×894 or 1080×1920 resolution. For our tests, we either use the DAVIS2017 test set, or we follow [5] and use benchmark videos `Kobe`, `Crash`, `Traffic`, and our addition of `Fan` (see Fig. 2) with a size of $k \times 256 \times 256$. Note that we excluded a few of the videos used in [5] as they have less than 16 frames which prohibit testing with 4×4 masks. Moreover, for both benchmarks, we limit ourselves to reconstructing the first k subframes.

4.2 Baselines

For comparison we use two other learned approaches: STFormer [5] and a unified video recovery framework with a UNet backbone [6]; the former is a SOTA model for SCI and the latter is closely related to our C2B use case.

STFormer uses a spatio-temporal vision transformer with a token generation block and video reconstruction block connected by STFormer blocks. The main contribution within the STFormer block is the self-attention mechanism present in both the spatial and temporal dimension. This lets attention parameters specialize to their specific medium. On the other hand, the framework in [6] uses the simpler temporal UNet architecture to generate reconstructions $\hat{\mathbf{X}}$.

TABLE 1: Quantitative Results of Methods on the STFormer benchmark.

k	Method	Kobe			Crash			Fan			Traffic			Time/Image
		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	
2	Lin. Interp.	31.78	0.979	0.095	23.48	0.796	0.224	28.35	0.872	0.215	22.13	0.823	0.194	< 1 s.
	DPS	31.72	0.898	0.266	25.68	0.698	0.334	34.28	0.875	0.174	22.03	0.664	0.321	158 s.
	Multi-DPS	32.38	0.944	0.198	25.27	0.734	0.271	32.47	0.870	0.187	22.00	0.753	0.232	168 s.
	STFormer [5]	38.04	0.985	0.049	31.10	0.895	0.048	38.47	0.888	0.093	30.92	0.934	0.063	< 1 s.
	UNet [6]	**38.54**	**0.986**	**0.039**	**31.66**	**0.900**	**0.038**	**40.76**	**0.891**	**0.086**	**32.07**	**0.949**	**0.048**	< 1 s.
4	Lin. Interp.	24.56	0.857	0.316	19.21	0.663	0.433	23.47	0.816	0.334	16.72	0.561	0.454	< 1 s.
	DPS	22.20	0.601	0.515	21.63	0.561	0.473	29.15	0.846	0.205	17.89	0.432	0.474	190 s.
	Multi-DPS	24.09	0.719	0.412	21.80	0.600	0.416	28.36	0.841	0.214	17.50	0.479	0.395	172 s.
	STFormer [5]	**29.54**	**0.931**	**0.153**	**26.60**	**0.813**	**0.164**	**29.78**	**0.867**	**0.182**	**22.67**	**0.838**	**0.167**	< 1 s.
	UNet [6]	25.53	0.807	0.322	25.55	0.785	0.188	24.41	0.765	0.366	21.58	0.711	0.310	< 1 s.

The UNet is augmented with a shift-variant convolution to extract features that are then passed into the UNet and regularizes outputs by a TV-Loss: $\mathcal{L}_{TV} = \left\| \nabla_{x,y} \hat{\mathbf{X}}[x, y] \right\|_1$. Moreover, making use of the two measurements provided by C2B, [6] ablates improvements from an extra input $\tilde{\mathbf{Y}}$ which is the bucket-1 measurement versus that of the "blurred" measurement which is the sum of both bucket-0 and bucket-1 measurements.

In addition, we used a linear interpolation scheme. Recall from the exposition on pixel reshuffling that $\mathbf{X}'[f]$ contains mostly black pixels except the one pixel in each tile. Hence, video compressive sensing can be framed as an inpainting problem: conditioned on the sparse original pixels, fill in the gaps indicated by the black pixels.

4.3 Implementation

For the STFormer and UNet baselines, we used the code implementation provided. The only change we made was to the masks where we used the mask family in (1) for $k = 2, 4, 6$ and 8. Another detail is that we passed the "blurred" measurement, as defined above, as input to the UNet approach [6]. Regarding the DPS implementation, we also used the author's implementation, only adding our conditioning methods as defined in (5) and (6) with $\rho = \rho_1 = \rho_2 = 1$. In all cases, we forego the measurement noise, i.e. $\mathbf{Z} = 0$.

Training for STFormer and UNet was done on either a NVIDIA RTX A6000 or RTX TITAN, depending on availability at the time. DPS required no training as we only modify the inference process of a pre-trained diffusion model; however, due to memory constraints, we had to perform microbatching to fit the subframes into memory for $k \geq 4$.

4.4 Results on Test Videos

In this section, we present results on the test datasets. To distinguish the two benchmarks, we refer to the set of `Kobe`, `Crash`, `Traffic`, and `Fan` as the STFormer benchmark.

4.4.1 STFormer Benchmark

Here, we compare the methods on the STFormer benchmark with quantitative results in Table 1 and example reconstructions in Fig. 3. We limited our experiments to $k = 2$ and 4 as most videos in this benchmark have at most 32 frames, hence proper testing is not possible for $k = 6$ or 8. It's clear from the metrics in Table 1 that STFormer and UNet performed better than that of DPS or the naive interpolation baseline. The scores suggest that the case of $k = 2$ is much too easy for the learned methods as we attain near perfect scores on SSIM and LPIPS. In this case, the simpler UNet [6] comes out ahead of STFormer which suggests that the dedicated temporal attention takes away from the spatial knowledge needed for perfect reconstruction.

However, the advantage of STFormer becomes apparent for larger k, as now the scene undergoes more significant changes across time. Indeed, at $k = 4$, STFormer is now the consistent best performer, which is confirmed by the crisper video in Fig. 3. Linear interpolation also scores surprisingly well, even beating out the DPS methods. An interesting point is that DPS seems to perform quite well on `Fan`, even matching the scores of STFormer and UNet at $k = 4$. This might not be surprising as the video lacks a lot of the details present in the others (cf. Fig. 2 and Fig. 3). Further on the DPS methods, Multi-DPS does not seem to have that large of an effect on the metrics or the inference time. However, looking at the images in Fig. 3 and Fig. 4, we see that the details of the car/trucks are much better captured in space and more consistent over time. This is because the downsampled images are temporally consistent and so there is a weak signal to that effect in the reconstructions.

The inference times for the methods are also noted in Table 1. Each DPS image seems to hover around the 3 minute mark which is hundreds of time slower than other methods. It should be noted, that DPS does not require training, so there is a reasonable trade-off being made. However, it's often the case that we do not mind a long training run if we can perform efficient inference thereafter, so in most scenarios, DPS is less preferred.

4.4.2 DAVIS2017 Benchmark

As STFormer and UNet are the best performing, we proceed to compare them on the *DAVIS2017* test set. As the videos therein are much longer, this allows us to use larger k which present more difficulty. As seen in the STFormer benchmark, the UNet outperforms in the $k = 2$ case, but otherwise STFormer takes over. As expected, the performance degrades as we increase k, but as seen in Fig. 5 and 6, STFormer is able to maintain some detail even at $k = 8$,

TABLE 2: Quantitative Results of Methods on the *DAVIS2017* benchmark.

Method	$k=2$			$k=4$			$k=6$			$k=8$		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Lin. Interp.	22.46	0.748	0.287	18.69	0.570	0.516	16.85	0.473	0.605	15.51	0.408	0.653
STFormer [5]	27.37	0.832	0.142	**23.89**	**0.714**	**0.305**	**22.36**	**0.632**	**0.411**	**20.87**	**0.528**	**0.493**
UNet [6]	**29.67**	**0.871**	**0.109**	22.75	0.624	0.408	19.76	0.499	0.504	18.02	0.399	0.627

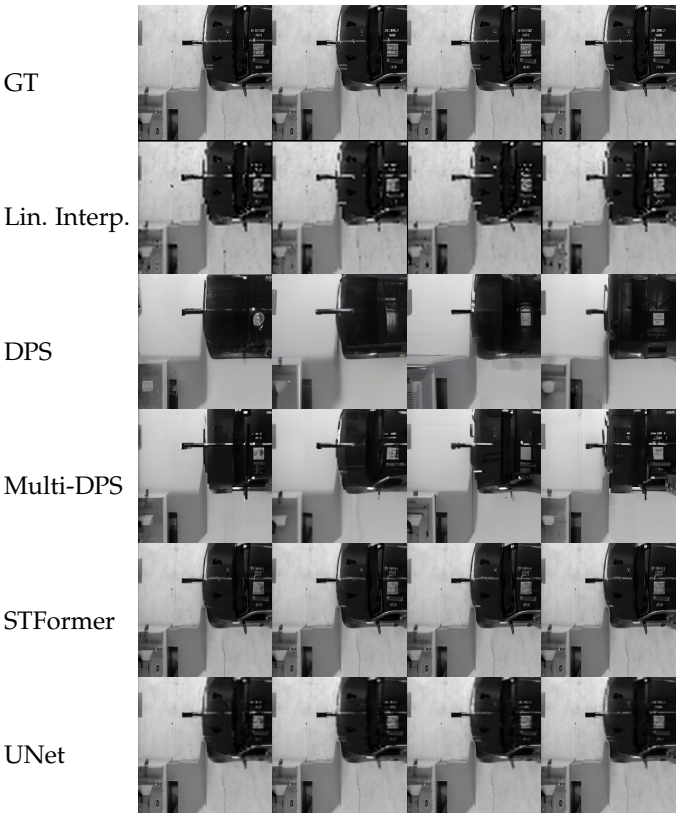

Fig. 3: Reconstructions of `crash32` when $k=4$. We display the first 4 reconstructed subframes from left to right.

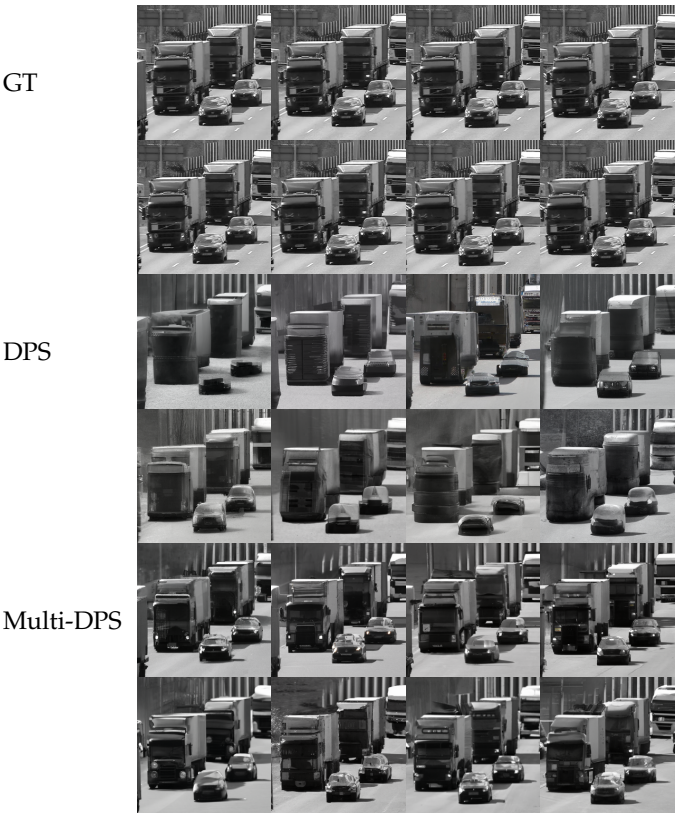

Fig. 4: DPS reconstructions of `traffic` when $k=4$. We display the first 8 reconstructed subframes in reading order.

whereas UNet fails to draw much distinction between the people and background in both figures. Overall, the tests on *DAVIS2017* confirm the claim in the previous section that STFormer will outperform UNet for larger k as there is more variation across time, allowing the dedicated temporal attention to play a larger role.

5 Conclusion

In this article, we present the use of diffusion models and prior video reconstruction approaches for video recovery with our particular cameras [2], [3]. This setting uses a specialized mask family, parameterized by $k \in \mathbb{N}$, and is generally harder than the random masking in SCI benchmarks [5]. To tackle this problem, we employed diffusion posterior sampling (DPS) with the coding operator, and a Multi-DPS variant which uses reshuffled frames as a low-resolution guide. We supplement our investigation with past approaches: STFormer and the UNet-based framework in [6] using a common benchmark in SCI [5], [11]. Moreover, we tested the STFormer and UNet approach on the more challenging *DAVIS2017* test set to verify it's ability to perform video reconstruction.

We found that the STFormer and UNet approaches still outperform the DPS variants while also being more efficient during inference. One limitation of the diffusion methods is its lack of temporal consistency. As shown in recent video diffusion models [14], [29], this can be implemented in the denoising network to improve generations. Moreover, a few methods have been proposed to speed up DPS [30], [31], alongside offshoots of DPS that denoise more efficiently [20], [22]. Hence, incorporating these improvements would be interesting next steps to explore when using diffusion models for video reconstruction.

Acknowledgments

The author would like to thank Kelly Zhu for assistance on the image evaluation metrics and providing example testing scripts, and Bora Bayazit for help in gathering data needed to train the measurement GAN. The author also would

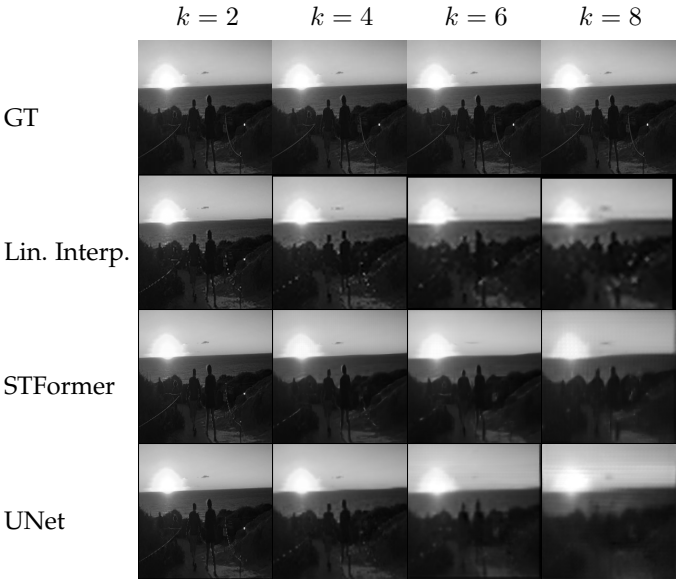

Fig. 5: Reconstructions of `people-sunset` from the *DAVIS2017* test set for different k.

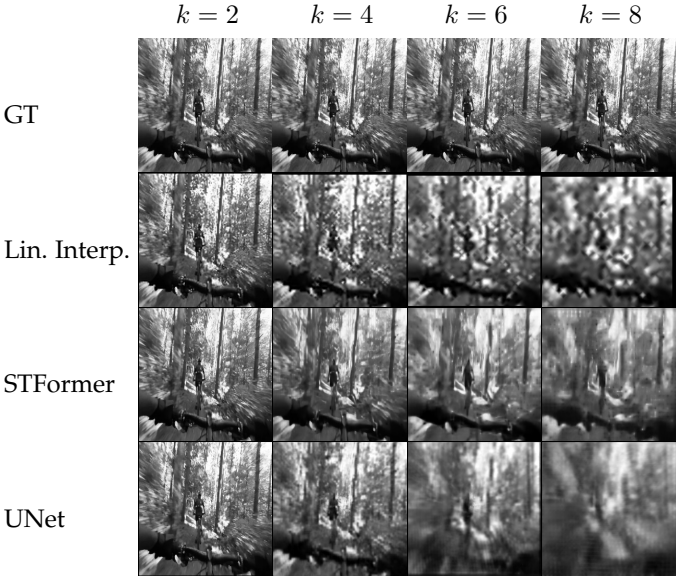

Fig. 6: Reconstructions of `mtb-race` from the *DAVIS2017* test set for different k.

like to thank David Lindell and Kyros Kutulakos for their supervision and helpful discussions.

REFERENCES

[1] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar, "Video from a single coded exposure photograph using a learned over-complete dictionary," in *2011 International Conference on Computer Vision*. Barcelona, Spain: IEEE, Nov. 2011, pp. 287–294.

[2] M. Wei, N. Sarhangnejad, Z. Xia, N. Gusev, N. Katic, R. Genov, and K. N. Kutulakos, "Coded Two-Bucket Cameras for Computer Vision," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, vol. 11207, pp. 55–73.

[3] R. Gulve, N. Sarhangnejad, G. Dutta, M. Sakr, D. Nguyen, R. Rangel, W. Chen, Z. Xia, M. Wei, N. Gusev, E. Y. H. Lin, X. Sun, L. Hanxu, N. Katic, A. M. S. Abdelhadi, A. Moshovos, K. N. Kutulakos, and R. Genov, "39 000-subexposures/s dual-adc cmos image sensor with dual-tap coded-exposure pixels for single-shot hdr and 3-d computational imaging," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 11, pp. 3150–3163, 2023.

[4] X. Yuan, D. J. Brady, and A. K. Katsaggelos, "Snapshot compressive imaging: Theory, algorithms, and applications," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 65–88, 2021.

[5] L. Wang, M. Cao, Y. Zhong, and X. Yuan, "Spatial-Temporal Transformer for Video Snapshot Compressive Imaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 7, pp. 9072–9089, Jul. 2023.

[6] P. Shedligeri, A. S, and K. Mitra, "A Unified Framework for Compressive Video Recovery from Coded Exposure Techniques," Nov. 2020.

[7] S. W. Hasinoff, F. Durand, and W. T. Freeman, "Noise-optimal capture for high dynamic range photography," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 553–560.

[8] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron, "Unprocessing images for learned raw denoising," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 036–11 045.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[10] K. Monakhova, S. R. Richter, L. Waller, and V. Koltun, "Dancing under the stars: Video denoising in starlight," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 16 241–16 251.

[11] Z. Cheng, R. Lu, Z. Wang, H. Zhang, B. Chen, Z. Meng, and X. Yuan, "BIRNAT: Bidirectional recurrent neural networks with adversarial training for video snapshot compressive imaging," in *European Conference on Computer Vision (ECCV)*, August 2020.

[12] Y. Li, M. Qi, R. Gulve, M. Wei, R. Genov, K. N. Kutulakos, and W. Heidrich, "End-to-End Video Compressive Sensing Using Anderson-Accelerated Unrolled Networks," in *2020 IEEE International Conference on Computational Photography (ICCP)*, Apr. 2020, pp. 1–12.

[13] Z. Meng, S. Jalali, and X. Yuan, "Gap-net for snapshot compressive imaging," 2020. [Online]. Available: https://arxiv.org/abs/2012.08364

[14] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis, "Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models," Dec. 2023.

[15] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, V. Jampani, and R. Rombach, "Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets," Nov. 2023.

[16] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," Apr. 2022.

[17] Z. Wu, R. Lu, Y. Fu, and X. Yuan, "Latent diffusion prior enhanced deep unfolding for spectral image reconstruction," *arXiv preprint arXiv:2311.14280*, 2023.

[18] Z. Pan, H. Zeng, J. Cao, K. Zhang, and Y. Chen, "Diffsci: Zero-shot snapshot compressive imaging via iterative spectral diffusion model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 25 297–25 306.

[19] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, "Diffusion Posterior Sampling for General Noisy Inverse Problems," https://arxiv.org/abs/2209.14687v4, Sep. 2022.

[20] L. Rout, N. Raoof, G. Daras, C. Caramanis, A. Dimakis, and S. Shakkottai, "Solving Linear Inverse Problems Provably via Posterior Sampling with Latent Diffusion Models," in *Thirty-Seventh Conference on Neural Information Processing Systems*, Nov. 2023.

[21] A. Tewari, T. Yin, G. Cazenavette, S. Rezchikov, J. B. Tenenbaum, F. Durand, W. T. Freeman, and V. Sitzmann, "Diffusion with Forward Models: Solving Stochastic Inverse Problems Without Direct Supervision," https://arxiv.org/abs/2306.11719v2, Jun. 2023.

[22] M. Mardani, J. Song, J. Kautz, and A. Vahdat, "A Variational Perspective on Solving Inverse Problems with Diffusion Models," https://arxiv.org/abs/2305.04391v2, May 2023.

[23] J. Martel, L. Müller, S. Carey, P. Dudek, and G. Wetzstein, "Neural Sensors: Learning Pixel Exposures for HDR Imaging and Video Compressive Sensing with Programmable Sensors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, p. 1642–1653, 2020.

[24] D. Wang, S. Hart, D. Surovik, T. Kelestemur, H. Huang, H. Zhao, M. Yeatman, J. Wang, R. Walters, and R. Platt, "Equivariant Diffusion Policy," https://arxiv.org/abs/2407.01812v1, Jul. 2024.
[25] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018. [Online]. Available: https://arxiv.org/abs/1802.05957
[26] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
[27] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," 2018. [Online]. Available: https://arxiv.org/abs/1801.03924
[28] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv:1704.00675*, 2017.
[29] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, D. Parikh, S. Gupta, and Y. Taigman, "Make-A-Video: Text-to-Video Generation without Text-Video Data," Sep. 2022.
[30] H. Chung, B. Sim, and J. C. Ye, "Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 403–12 412.
[31] H. Chung, S. Lee, and J. C. Ye, "Decomposed diffusion sampler for accelerating large-scale inverse problems," 2024. [Online]. Available: https://arxiv.org/abs/2303.05754