

Supplementary Materials: Training-free Video Semantic Segmentation based on Diffusion Models

A. Clarification on training-free definition

We classify our method as training-free because it does not retrain any large neural networks, e.g. the UNet in Diffusion. However, we acknowledge that the classic KNN classifier is trained.

B. Broader impact

Video semantic segmentation has important applications in autonomous driving and surveillance security. However, the source videos used for segmentation may contain private information such as human faces and driving plates. Guidelines for responsible usage need to be made to prevent privacy invasion. Also, diffusion models can inherit and propagate biases from their training data, leading to unfair treatment of certain groups.

C. Taxonomy of video segmentation tasks

Here we provide a short taxonomy of different video segmentation tasks:

- Video Semantic Segmentation (VSS) aims to predict a semantic class for every pixel according to the pre-defined categories in a video.
- Video Object Segmentation (VOS) aims to segment and track the dominant object(s) in a video.
- Video Instance Segmentation (VIS) aims to segment and track individual instances of object(s) in a video.
- Video Panoptic Segmentation (VPS) aims at segmenting every pixel either into foreground object instances or background semantic classes in a video.
- Promptable Video Segmentation (PVS) it is a new video segmentation paradigm introduced by [5] that aims to segment an object through a video as specified by a user prompt (point, bounding box, or mask).

D. Settings

Dataset. VSPW (Video Scene Parsing in the Wild) is a large-scale video semantic segmentation dataset that consists of a wide range of real-world scenarios and categories. It has 124 categories in total. The resolution of this dataset is 480×853 . Since each video consists of less than 10 classes, we set the number of clusters for KMeans as 20. Cityscapes is a large-scale urban streets video sequence dataset. The objects are grouped into 30 classes in total. Each video clip has 30 frames, and only the 20th frame has dense annotations. We use its validation set, which contains 15000 frames from three cities. As the original resolution of the frames is 1024×2048 , which is too big to fit into the GPU memory with SVD, we downsample the frames to 256×512 for all the experiments and evaluate at the original resolution by upsampling the segmentation maps. As each video clip may contain classes of more than 10, we set the number of clusters as 30 in order to capture the small objects. CamVid (Cambridge-driving Labeled Video Database) is a road scene dataset with dense segmentation annotations with 11 classes in total. The resolution of this dataset is 360×480 . We use its validation set, which has one video clip and contains 100 frames. We set the number of clusters for KMeans as 20.

Implementation details. We fix all hyperparameters for all videos and datasets, and we do not apply any post-processing methods like a conditional random field (CRF) on the output segmentation maps. All experiments were conducted on a single NVIDIA A100 40G GPU. We build our method on top of SD 2.1 and SVD code repository <https://github.com/Stability-AI/generative-models>.

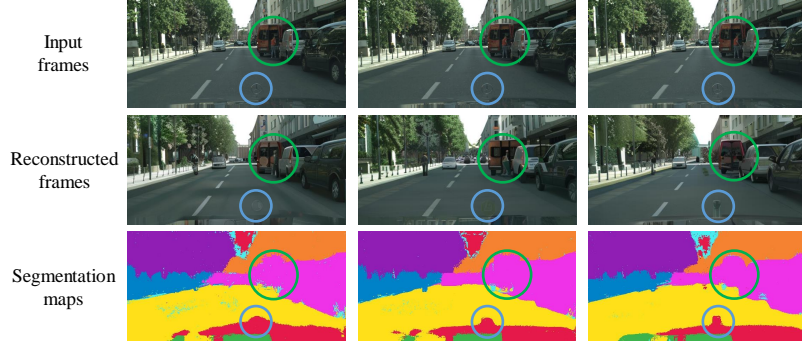


Figure 1. Failure case: inversion. In adjacent frames, the car and the sign of the car experience shape changes, which finally result in obvious shape changes on the segmentation maps. We highlight the main areas of discrepancy in green and blue circles.

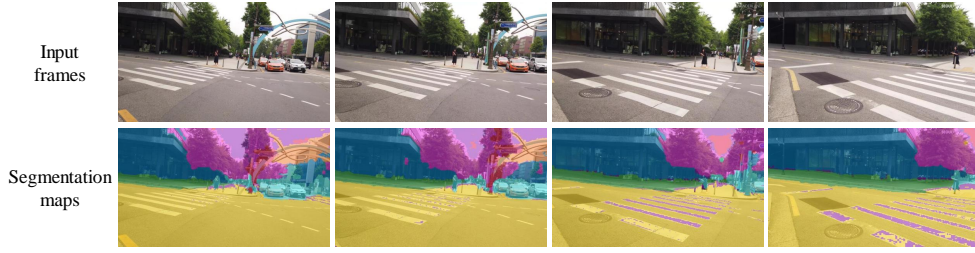


Figure 2. Failure case: temporal inconsistency. The sidewalk is clustered into the yellow region in the first frame, and later the same sidewalk is clustered into another region denoted as purple region.

Computational resources. We use one NVIDIA A100 GPU to conduct all our experiments. As the running time depends on the number of clusters and spatial resolution of the video frames, generally, it will take around 2 minutes to process a batch of video frames using SD 2.1 and 5 minutes using SVD. The main cost of the time comes from the modulating process, which involves several modified forward passes of the backbone model.

E. Failure cases

We identify several typical failure cases using our method. The first one is the inaccurate diffusion inversion process (Figure 1). In adjacent frames, the texture and shape details of the small objects can be different after inversion. Segmentation maps will also inherit this discrepancy between input frames and reconstructed frames, which will result in flickering on the frames. The second one is flickering between similar regions (Figure 2). Some objects are originally assigned with one cluster, while in the later frames, they are grouped into different clusters. The third one is dealing with unseen objects (Fig. 3). As we only use an anchor frame’s ground truth to guide the class-agnostic clusterings, a feature-bank based strategy could be adopted to adapt our approach to handle dynamic videos.

F. Hyperparameters

We provide all hyperparameters for SD 2.1 and SVD in Table 1.

G. Efficiency

We report the FPS and GPU memory consumption of our methods on the validation subset of the DAVIS 2017 dataset in Tab. 2. Our approach with SD2.1 as backbone processes 0.41 frames per second, where the primary bottleneck is the multiple forward passes of the diffusion model required for the feature aggregation and modulation steps. It is worth mentioning that SD2.1 and SVD models are not lightweight and not optimized for rapid forward pass compared to vision backbones such as ViTs and ResNets. Since our aim was to establish a new paradigm for zero-shot video segmentation based on diffusion models, our main focus was to achieve competitive segmentation accuracy. For real-time applications where achieving the



Figure 3. Failure case: unseen objects. The door and albums, which do not appear in the previous frames, are assigned with the wrong clusters.

Table 1. Hyperparameters settings for SD and SVD.

	Ours (SD 2.1)	Ours (SVD)
t_f	25	25
t_m	20	17
Sampling timesteps	25	25
Sampler	EDM	EDM
Block b_k	Block 6, 7 and 8	Block 6, 7 and 8
Block b_m	Block 7	Block 8
Block c	Block 7	Block 8
Spatial threshold \mathcal{T}	1	1
Filtering strength s	0.7	0.7
Modulating factor λ	50.0	50.0
Modulating attention type	cross attention	self attention
Injected features	spatial attention	spatial & temporal attention

highest efficiency is required, further efforts are needed as a follow-up to our work. These efforts can include training an adapter to perform the feature aggregation efficiently, similar to Luo et al. [3], replacing the modulation process with a feature upsampling module as in Fu et al. [2], or even fine-tuning Stable Diffusion to function as end-to-end segmentation models.

Table 2. Efficiency comparison.

Method	Backbone	# of parameters	FPS	GPU Memory (GB)
EmerDiff	SD2.1	865M	0.44	10
Ours	SD2.1	865M	0.41	21
Ours	SVD	1.5B	0.12	39

H. Evaluation on Video Object Segmentation dataset

We provide results on the DAVIS 2016 and DAVIS 2017 Video Object Segmentation (VOS) datasets to demonstrate that our method can be applied to other video segmentation tasks. Our approach consistently outperforms EmerDiff by a huge margin on VOS, demonstrating its applicability to other VS tasks.

I. Long video segmentation

We are not aware of any existing long video semantic segmentation dataset. Therefore, we test our approach on the CLVOS [4] dataset for Video Object Segmentation (VOS), which has long videos of an average length of 1506 frames. This is significantly longer than the VSPW dataset (71 frames). In theory, there is no restriction on the maximum length of the video, as we process the video frames with a fixed batch size.

Table 3. Evaluation on DAVIS 2016 and DAVIS 2017 datasets.

	DAVIS 2016			DAVIS 2017		
	$\mathcal{J} \& \mathcal{F}$	\mathcal{J}	\mathcal{F}	$\mathcal{J} \& \mathcal{F}$	\mathcal{J}	\mathcal{F}
EmerDiff	22.1	20.3	23.9	18.6	15.9	21.3
Ours (SVD)	66.1	67.7	64.6	40.7	40.9	40.4
Ours (SD2.1)	79.0	70.5	69.3	60.1	57.6	62.6

We show the results in Tab. 4. Our approach still performs well on these significantly longer videos, while EmerDiff drastically fails. These results highlight the large improvement we made in the zero-shot segmentation setting. Future work can be further improvements on long video segmentation.

Table 4. Evaluation on CLVOS dataset.

	$\mathcal{J} \& \mathcal{F}$	\mathcal{J}	\mathcal{F}
EmerDiff	0.8	0.7	0.9
Ours (SVD)	23.2	22.1	24.3
Ours (SD2.1)	46.7	46.8	46.5

J. Comparison to CRF techniques

Conditional Random Fields (CRF) is a technique widely used in machine learning applications. It predicts a label for a single sample while also considering neighbouring samples. In semantic segmentation, CRF is commonly used as a post-processing technique to refine the raw predictions from the model. Here, we ablate on the improvement of our approach compared to the CRF technique. We use the implementation from a Deep CRF proposed in Deeplab[1] and adopt it to the predictions of EmerDiff. We compare this post-processed EmerDiff with our approach VidSeg *without CRF post-processing* on the first 30 videos of the validation set of the VSPW dataset. We provide the results in Tab. 5. We show that our approach still significantly surpasses the post-processed EmerDiff in terms of both mIoU and mVC.

Table 5. Ablation on CRF technique

	mIoU	mVC ₈	mVC ₁₆
EmerDiff	31.0	68.8	64.0
EmerDiff + CRF	31.8	75.5	71.0
Ours	47.2	89.4	87.9

K. Ablation study

We provide additional ablation experiments here. We ablate the modulating Block b_m in Figure 4 for both SD and SVD. Modulating Block 7 and Block 8 for SD and SVD, respectively, can give the most spatial details as well as maintain the semantics. We additionally provide quantitative ablation for the modulating block b_m for SD in Table 7. These blocks are, at the same time, the most semantically-riched blocks in SD and SVD. In Figure 5, we show more examples of how latent blending and difference map filtering help with removing spatial noises. In Figure 6, we show comparisons between the difference maps and segmentation maps produced by SD and SVD. We show that the difference maps of SD contain finer details, which bring sharper boundaries to the final segmentation maps. We further show PCA visualization of a 64×64 resolution block. We show that in high-resolution blocks, SD features have more semantic information and spatial details, as well as more temporally stable than SVD spatial features and temporal features. We also show that segmentation maps under different numbers of K-Means clusters and after GT labels reassignment in Figure 7. Increasing the number of clusters can help segment more small objects (from 5 to 20). However, further increasing the number of clusters may not necessarily

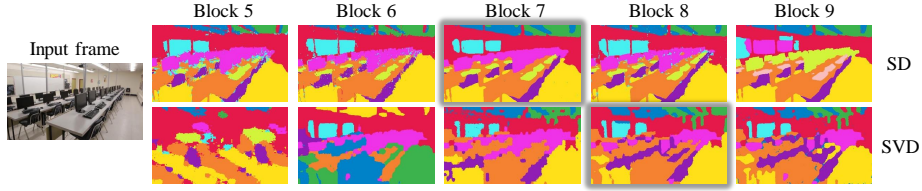


Figure 4. Ablation: modulating different block will result in different segmentation maps. Modulating Block 7 and Block 8 give the best results for SD and SVD, respectively (highlighted in shadow).

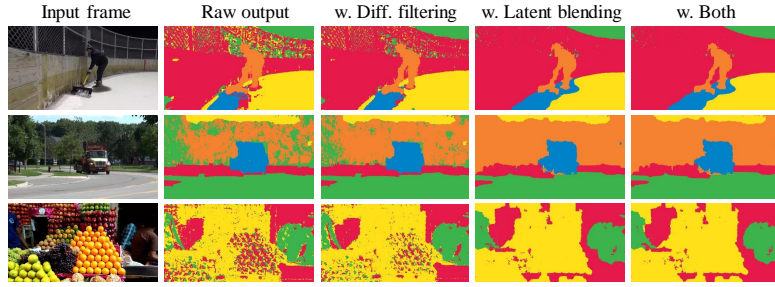


Figure 5. Ablation: latent blending and difference map filtering.

segment out the clusters that are aligned with defined clusters in ground truth (from 20 to 30). Although K-Means originally generated 30 clusters, most of them are merged to the same classes after GT label reassignment.

We also ablate on the classifier we use in Stage 2 on the first 30 videos of the VSPW validation set in Tab. 6. We opt for low-complexity classifiers for a reduced computational overhead and to avoid overfitting. The results show that both KNN and MLP achieve a good tradeoff between speed and performance. RandomForest performs slightly better but at an increased computational overhead.

Additionally, we ablate on the number of frames / batch size B we use. We do this ablation on a subset of Cityscapes dataset. The mIoU achieved with 5, 10, 14, and 20 frames are 32.9, 34.9, 34.4, and 31.8, respectively. We show that too many frames may cause inter-batch variance, while too few frames may not be able to maintain consistency. Therefore, through out our main experiments, we use $B = 14$.

Table 6. Ablation on different classifiers

Classifier	mIoU	mVC ₈	mVC ₁₆	Speed
Adaboost	34.1	82.4	79.2	1x
Random Forest	47.9	90.5	89.0	148x
MLP	47.1	89.1	87.6	240x
KNN	46.5	89.8	88.4	240x

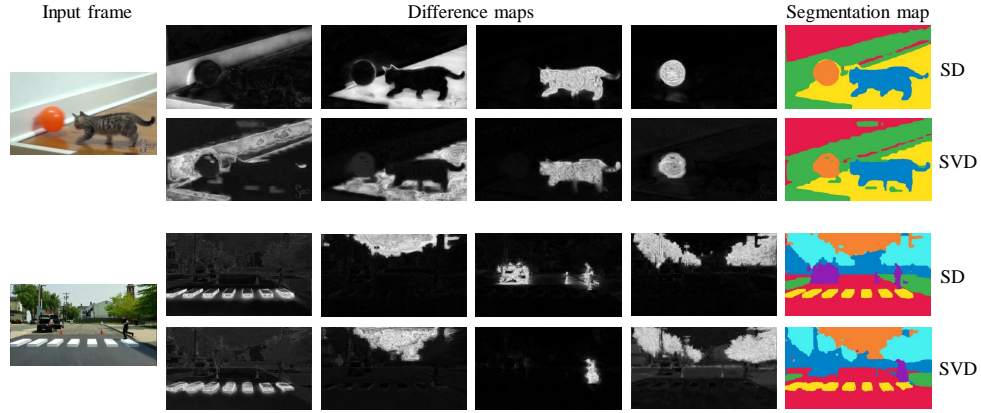


Figure 6. Ablation: Difference maps comparison between SD and SVD.

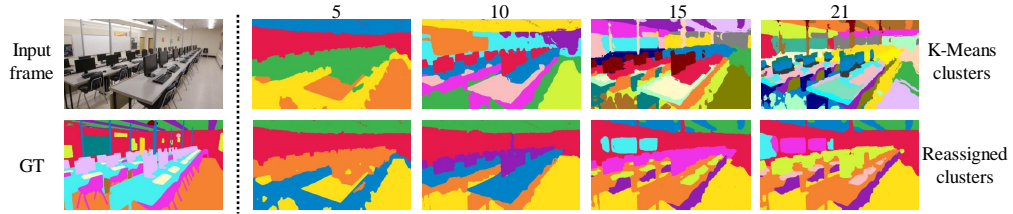


Figure 7. Ablation: number of K-Means clusters. In the first row, we show the segmentation maps generated by different numbers of clusters in K-Means. In the second row, we show the segmentation maps generated by the same number of clusters followed by GT labels reassignment.

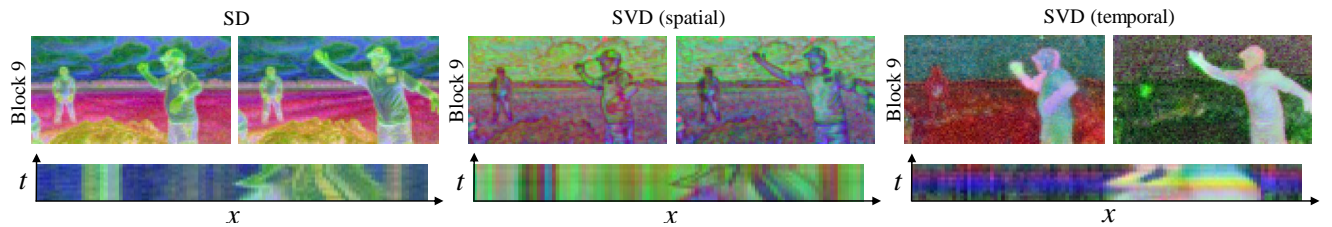


Figure 8. A visualization of the first three PCA components for the features extracted from the most semantically-rich blocks of Block 9 in both SD and SVD of the first and last video frames in a batch. In the second row, we show the x - t slice of a set of pixels (highlighted in the red line in the leftmost PCA visualization) horizontally across the PCA visualization (x -axis) and stack it chronologically across the full batch of video frames (t -axis).

Table 7. Ablation of the modulation block b_m for SD 2.1. Block 7 gives the best performance considering both mIoU and mVC.

Block	mIoU	mVC ₈	mVC ₁₆
3	45.6	89.4	88.2
4	47.0	88.4	86.9
5	47.0	87.1	85.5
6	45.1	82.3	79.9
7	47.2	90.1	88.7
8	44.5	83.7	81.3
9	46.2	90.2	88.9
10	43.7	91.1	89.8
11	47.7	87.3	85.6

References

- [1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. [4](#)
- [2] Stephanie Fu, Mark Hamilton, Laura E. Brandt, Axel Feldmann, Zhoutong Zhang, and William T. Freeman. Featup: A model-agnostic framework for features at any resolution. In *The Twelfth International Conference on Learning Representations*, 2024. [3](#)
- [3] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. In *Advances in Neural Information Processing Systems*, 2023. [3](#)
- [4] Amir Nazemi, Zeyad Moustafa, and Paul Fieguth. Clvos23: A long video object segmentation dataset for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2496–2505, 2023. [3](#)
- [5] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. [1](#)