

SVS-GAN for Semantic Synthesis of Traffic Videos for Autonomous Driving

Supplementary Material

Khaled M. Seyam¹, Julian Wiederer², Markus Braun², Bin Yang¹

¹Institute of Signal Processing and System Theory, University of Stuttgart, Germany

²Mercedes-Benz AG, Stuttgart, Germany

{khaled.seyam, bin.yang}@iss.uni-stuttgart.de

{julian.wiederer, markus.ma.braun}@mercedes-benz.com

A. Losses

Our architecture is optimized using an extensive set of loss functions designed to refine different aspects of the generated output. The individual losses and their formulations are as follows:

- **OASIS Adversarial Loss for D_I :** As [9], to ensure that the generator synthesizes images that align with the input semantic label maps, we use a discriminator that can perform both semantic segmentation and fake/real detection. This is achieved by casting the discriminator task as a multi-class semantic segmentation problem. The loss is defined as:

$$\mathcal{L}_{D_I} = -\mathbb{E}_{(I,S)} \left[\sum_{c=1}^K \alpha_c \sum_{i,j} S_{i,j,c} \log D_I(I)_{i,j,c} \right] - \mathbb{E}_S \left[\sum_{i,j} \log D_I(G(S))_{i,j,c=K+1} \right].$$

where K is the number of real classes, α_c is the weight for class c , $s_{i,j,c}$ is the label at position (i, j) for class c , $D_I(I)_{i,j,c}$ is the discriminator output at position (i, j) for class c , and $D_I(G(S))_{i,j,c=K+1}$ is the discriminator output at position (i, j) for the fake class $K + 1$.

To balance the contributions of each class, we weight each class by its inverse per-pixel frequency, giving more importance to rare classes and encouraging the generator to synthesize less-represented classes adequately.

- **OASIS Adversarial Loss for G [9]:** The objective of this loss is to make sure that the images generated by G are both realistic and semantically consistent with the input label maps. This is achieved by training the generator to maximize the likelihood that the discriminator

D_I will classify the generated images as belonging to the correct semantic classes.

$$\mathcal{L}_{\text{OASIS}} = -\mathbb{E}_S \left[\sum_{c=1}^K \alpha_c \sum_{i,j} S_{i,j,c} \log D_I(G(S))_{i,j,c} \right]. \quad (1)$$

- **Video GAN loss $\mathcal{L}_{\text{GAN}}^V$:** We adopt the least-squares formulation to stabilise training of the video discriminator D_V :

Discriminator update

$$\mathcal{L}_{D_V} = \mathbb{E}_v [(D_V(v) - 1)^2] + \mathbb{E}_{\hat{v}} [D_V(\hat{v})^2]. \quad (2)$$

Generator update

$$\mathcal{L}_{\text{GAN}}^V = \mathbb{E}_{\hat{v}} [(D_V(\hat{v}) - 1)^2], \quad (3)$$

where v and $\hat{v} = G(S)$ are real and generated 3-frame clips, respectively. The LS penalty encourages D_V to output 1 on real clips and 0 on fake, while G is pushed to make $D_V(\hat{v})$ approach 1, thus promoting temporally coherent videos without unstable gradients.

- **VGG Loss [4]:** This loss ensures that the generated images are perceptually similar to the real images by utilizing features extracted from multiple layers of a VGG network [8]. The VGG loss measures the difference between the feature representations of the generated image and the real image:

$$\mathcal{L}_{\text{VGG}} = \mathbb{E}_{(I,S)} \left[\sum_{l=1}^L \beta_l \|\phi_l^{\text{VGG}}(G(S)) - \phi_l^{\text{VGG}}(I)\|_1 \right]. \quad (4)$$

Here, ϕ_l^{VGG} denotes the feature maps from layer l of the VGG network, and L is the total number of layers used.

- **Feature Matching Loss [5]:** This loss ensures that the feature representations of generated images are closely aligned with those of real images, by comparing outputs from specific layers of the discriminators D_I and D_V . The feature matching loss is computed as the L1-norm difference between the features of the generated image and the real image across multiple layers:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{(I,S)} \left[\sum_{l=1}^L \gamma_l \|\phi_l^D(G(S)) - \phi_l^D(I)\|_1 \right]. \quad (5)$$

Here, ϕ_l^D represents the output from the l -th layer of the discriminators (D_I or D_V), where L denotes the total number of layers evaluated.

- **I3D perceptual loss \mathcal{L}_{I3D} .** To regularize motion over an entire clip we compare generated and real videos in the feature space of a Kinetics-400-pretrained I3D network [1], whose intermediate activations encode both appearance and temporal dynamics:

$$\mathcal{L}_{\text{I3D}} = \mathbb{E}_{(v,S)} \left[\sum_l \delta_l \|\phi_l^{\text{I3D}}(\hat{v}) - \phi_l^{\text{I3D}}(v)\|_1 \right], \quad (12)$$

where v is a real clip, \hat{v} its generated counterpart, $\phi_l^{\text{I3D}}(\cdot)$ denotes the activation of I3D layer l , and δ_l is a fixed per-layer weight.

The final loss function for our generator model is composed of several components, weighted appropriately based on insights derived from prior frameworks. The composition of the generator loss function is detailed as follows:

$$\begin{aligned} \mathcal{L}_G = & \lambda_{\text{OASIS}} \mathcal{L}_{\text{OASIS}} + \lambda_V \mathcal{L}_{\text{GAN}}^V \\ & + \lambda_{\text{VGG}} \mathcal{L}_{\text{VGG}} + \lambda_{\text{FM}} \mathcal{L}_{\text{FM}} + \lambda_{\text{I3D}} \mathcal{L}_{\text{I3D}}. \end{aligned} \quad (6)$$

These losses ensure that our architecture not only generates realistic and temporally coherent video sequences but also adheres closely to the given semantic maps.

B. Background on Key Components

For completeness, we briefly summarize the prior components integrated into SVS-GAN, highlighting the specific modules we adopt and how they are used in our framework.

SPADE Block SPADE [7] (Spatially-Adaptive Normalization) modulates feature activations with semantic information by replacing batch/instance normalization with spatially-varying affine transforms derived from segmentation masks. This preserves semantic detail that would

otherwise be washed out by standard normalization. In SVS-GAN, we adopt SPADE blocks inside the ResNet decoder, ensuring that semantic maps influence every scale of the decoding process for sharper alignment of generated frames with input masks.

OASIS Discriminator and Loss OASIS [9] introduced a segmentation-aware discriminator that judges realism at the *pixel* level, conditioned on semantic class labels, and uses a cross-entropy adversarial loss over semantic categories instead of a binary real/fake loss. This enforces stronger per-class fidelity and helps the generator respect semantic boundaries. We integrate the OASIS discriminator and loss into SVS-GAN to strengthen pixel-wise semantic adherence, especially on small or safety-critical classes.

EDSC Motion Compensation EDSC [2] (Enhanced Deformable Spatial Convolutions) extends deformable convolutions by learning dynamic spatial offsets and modulation masks, enabling more flexible geometric alignment than fixed-kernel convolutions. Unlike optical-flow-based warping, EDSC directly learns to align features under large displacements and occlusions. In SVS-GAN, we replace explicit flow with an EDSC-based motion branch that warps the previous frame toward the next semantic map, producing motion-compensated inputs that improve temporal coherence without relying on pre-computed flow.

C. High-Resolution Results (1024×512)

To verify that SVS-GAN scales beyond the 512×256 setting shown in the results of the main paper, we do spatial progression to 1024×512 as explained in the paper. We keep all hyper-parameters from unchanged, except for adding one extra SPADE stage to the decoder (Phase 3 of the progressive schedule).

Method	FID ↓	FVD _{i3d} ↓	FVD _{cd} ↓	mIoU ↑
Vid2Vid	69.07	126.71	97.90	55.40
SVS-GAN	39.42	72.64	59.75	66.86

Table 1. Cityscapes-Seq performance at 1024×512.

Discussion. (i) *Image / video fidelity.* SVS-GAN reduces FID by around 43% and both FVD metrics by 40–45% relative to Vid2Vid, confirming that our framework remain effective at higher resolution.

(ii) *Semantic alignment.* mIoU improves, compared to V2V, from 55.4% to 66.9%, demonstrating that dense mask adherence is still preserved when up-scaling.

(iii) *Training cost.* Because we fine-tune from the 512×256 checkpoint, the additional compute is modest

(eight epochs), showing that SVS-GAN scales gracefully without retraining from scratch.

D. Qualitative Ablation

Figure 1 illustrates how every architectural change translates into visible gains.

Scene 1 – bicycle approaching. The *Vid2Vid* baseline completely omits the on-coming bicycle. Adding the *OASIS* discriminator recovers the bicycle but leaves the van as is. The *Triple Pyramid* removes those artifacts and restores lane colors; however, flow warping still drags the rear of the van on the right. Replacing optical flow with *EDSC* alignment erases the smear and yields a crisp bicycle and correctly rendered van.

Scene 2 – static van with passing traffic. Colour drift afflicts the parked van in both *Vid2Vid* and *+OASIS*. The *Triple Pyramid* corrects tone but residual flow tearing blurs the cars on the right and paints the van reddish. With *EDSC* those cars become sharp and the van regains a consistent hue, underscoring the advantage of deformable, flow-free motion modelling.

Scene 3 – oncoming van. The baseline mis-colours and distorts the approaching van; *+OASIS* still lacks clear edges. The *Triple Pyramid* recovers shape and colour yet shows flow ghosts along the van. The final *EDSC* variant removes the smearing entirely and delivers the sharpest, most faithful reconstruction.

Although the quantitative gap between *+Triple Pyramid* and *+EDSC* is modest, these visuals reveal the practical impact of deformable alignment: *EDSC* systematically eliminates flow-induced blurs and tearing, producing cleaner and more stable videos that better respect object boundaries.

E. Further Results

Figure 2 presents additional results from the KITTI-360 dataset. This comparison illustrates the ability of our model to generate more realistic scenes, with particular emphasis on the coherence between the cars and the lighting conditions. Notably, in the frame before the last, the car positioned on the right is under a shadow, while the car on the left is exposed to direct light. This differentiation is crucial in generative models, as it enhances the utility of the synthesized data for training downstream tasks.

We further illustrate diversity on Cityscapes in Figure 3. Our framework maintains coherent geometry and appearance across varied road types and scene layouts—including cobblestone streets, highways, bridges, constructions, and crowded intersections with bicycles and pedestrians—while

preserving small structures (e.g., poles, signage) and consistent illumination. This variability is important for downstream training and evaluation, as it exposes perception models to a wider range of textures, densities, and lighting conditions without sacrificing temporal stability.

F. Failure cases

Single-frame temporal context. A limitation of SVS-GAN is its *single-frame* temporal context. During long occlusions, the model may lose appearance identity (e.g., object color) and reintroduce the object with a shifted look when it reappears. In Figure 4, the white car visible early in the sequence returns with a different hue after pedestrians occlude it. By contrast, **CTRL-V** generates 30 frames in a single pass, which promotes appearance consistency *within* that window; however, consistency is not guaranteed across adjacent windows, since each 30-frame segment is re-anchored to its own reference frame. Incorporating lightweight appearance memory (e.g., per-object feature banks or tracking-by-detection), periodic keyframe re-anchoring, or a shallow world-view module (persistent scene state across frames) into SVS-GAN is a promising direction to mitigate this failure mode.

Turning maneuvers and rapid viewpoint change. Sharp ego turns expose large previously unseen regions. In these cases SVS-GAN can under-synthesize the newly revealed content while preserving already visible areas, reflecting a bias toward small per-step updates and object motion. As shown in Figure 5, **CTRL-V** often fills new regions more aggressively—likely due to its generative pre-training—but with an off-domain, slightly “cartoonish” appearance relative to Cityscapes. It also tends to drop previously visible actors (e.g., pedestrians), whereas SVS-GAN maintains those entities with better semantic adherence.

G. Bounding-Box Adherence via YOLO mAP on KITTI-360

Following the evaluation protocol introduced in **CTRL-V**, we assess bounding-box adherence on KITTI-360 using mean Average Precision (mAP). Specifically, we apply YOLOv8 [3] to detect objects in both real and generated frames, then match bounding boxes across each pair of frames based on spatial overlap (IoU). As in **CTRL-V**, average Precision is then computed following the MS COCO protocol [6], and reported as mAP. This complements mIoU by directly measuring whether dynamic actors in the generated frames remain spatially consistent with the ground truth.

We use mAP primarily to test how well **Ctrl-V (BBOX)** adheres to its coarse conditioning compared to the

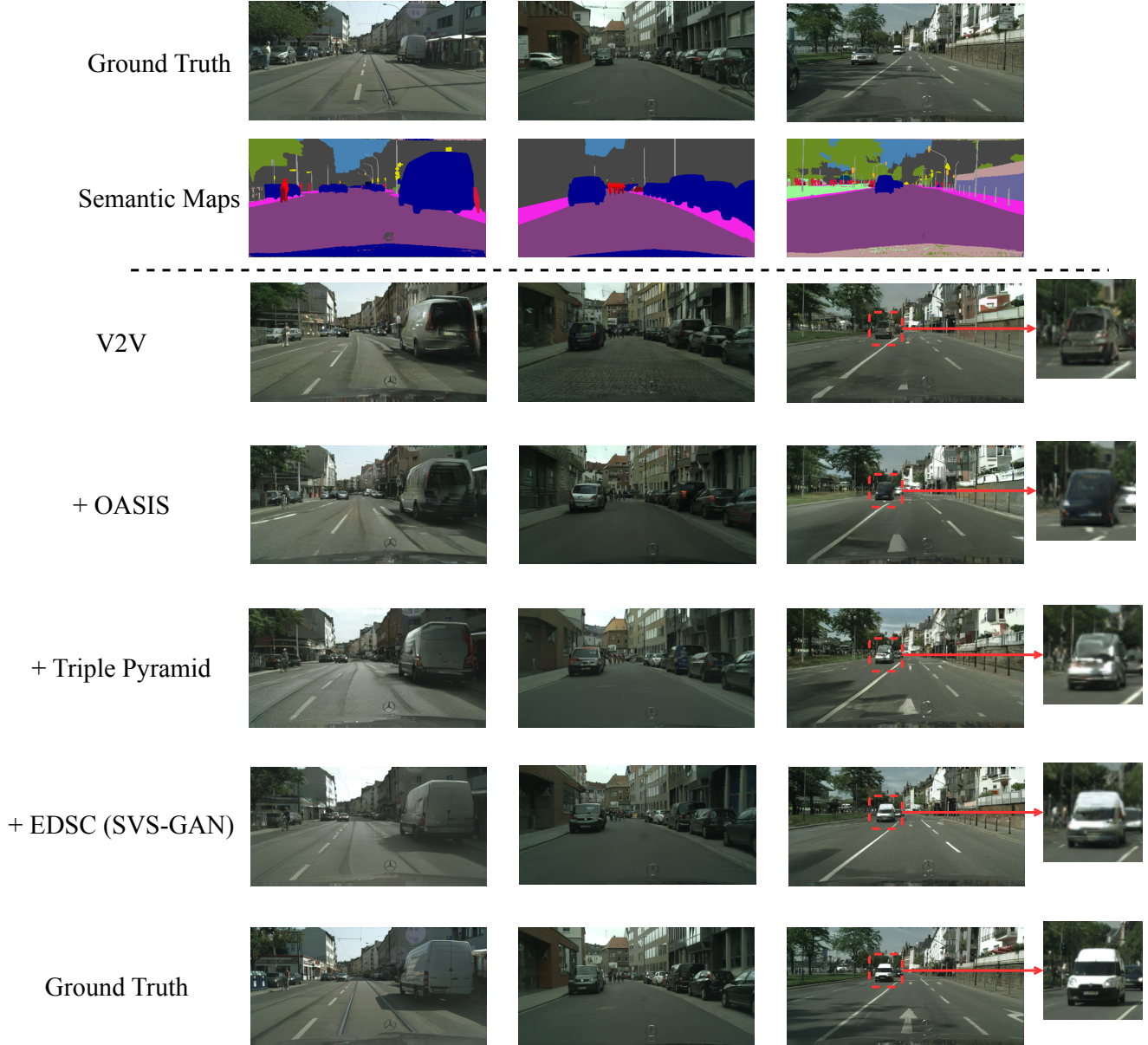


Figure 1. **Qualitative ablation on Cityscapes.** Each row shows, from left to right, the reference RGB, the last target map, the four generator variants—*Vid2Vid* baseline, +*OASIS*, +*Triple Pyramid*, +*EDSC*—and the ground-truth RGB. Red and green arrows highlight the bicycle in **Scene 1** and the parked van in **Scene 2**, respectively.

segmentation-trained Ctrl-V variant. As seen in Table 2, the box-only model is consistently lower across windows, indicating that coarse (box) control does not guarantee tight box adherence under long rollouts. In contrast, dense semantic conditioning (Ctrl-V w/ seg. maps) and GAN translators (*Vid2Vid*, SVS-GAN) generally maintain stronger object-level alignment. We note that *Vid2Vid*’s mAP remains high even late in the sequence, consistent with its design that renders dynamic objects via a dedicated branch; while this can boost per-frame localization, it does not by

itself ensure temporal coherence.

A key advantage of SVS-GAN is its ability to preserve small, safety-critical actors such as pedestrians in a temporally consistent way. As shown in Figure 6, YOLO detections reveal that SVS-GAN produces visible bounding boxes for pedestrians earlier in the rollout compared to other baselines. In contrast, both Ctrl-V variants and *Vid2Vid* delay or even miss early pedestrian appearances, despite producing cars with comparable confidence. This distinction is crucial for safety-oriented data augmentation: training

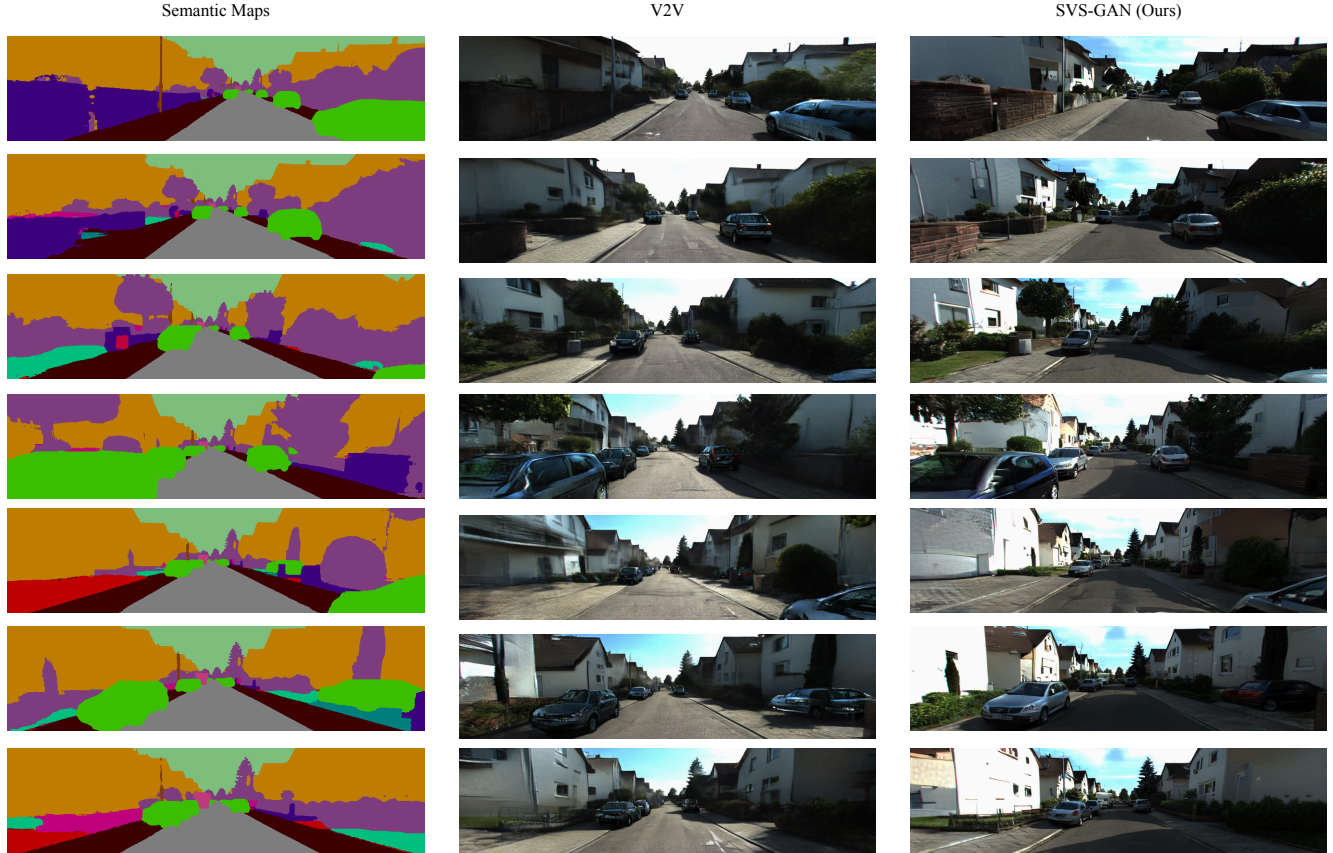


Figure 2. Further results from KITTI-360: On the left, the semantic maps are displayed; in the middle, the results from the V2V model; and on the right, our results. We extracted one frame from every 10 frames in the sequence to demonstrate the performance.

Method	0–3s	3–6s	6–9s	9–12s
Vid2Vid	55.41	49.39	64.11	63.33
Ctrl-V	56.71	80.66	62.38	31.22
Ctrl-V (BBOX)	55.68	39.13	34.78	23.98
SVS-GAN (Ours)	54.11	70.85	64.69	35.61

Table 2. Bounding-box adherence (YOLO mAP) on KITTI-360 across 3-second intervals.

downstream detectors on SVS-GAN videos exposes them to pedestrians sooner and more reliably, strengthening the robustness of perception systems under high-risk conditions such as crossings or sudden appearances.

References

- [1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. [2](#)
- [2] Xianhang Cheng and Zhenzhong Chen. Multiple video frame interpolation via enhanced deformable separable convolution, 2021. [2](#)
- [3] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. [3](#)
- [4] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016. [1](#)
- [5] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016. [2](#)
- [6] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. [3](#)
- [7] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019. [2](#)
- [8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [1](#)



Figure 3. Qualitative diversity results on Cityscapes. Each row shows a different road context: **Crossings**, **Highway**, **Bridge**, **Construction**, and **Cobblestone**. For each sequence we display the *reference* frame alongside the generated frames at steps I_{10} , I_{20} , and I_{30} . SVS-GAN preserves geometry and appearance across varied layouts and maintains temporal coherence over extended rollouts.

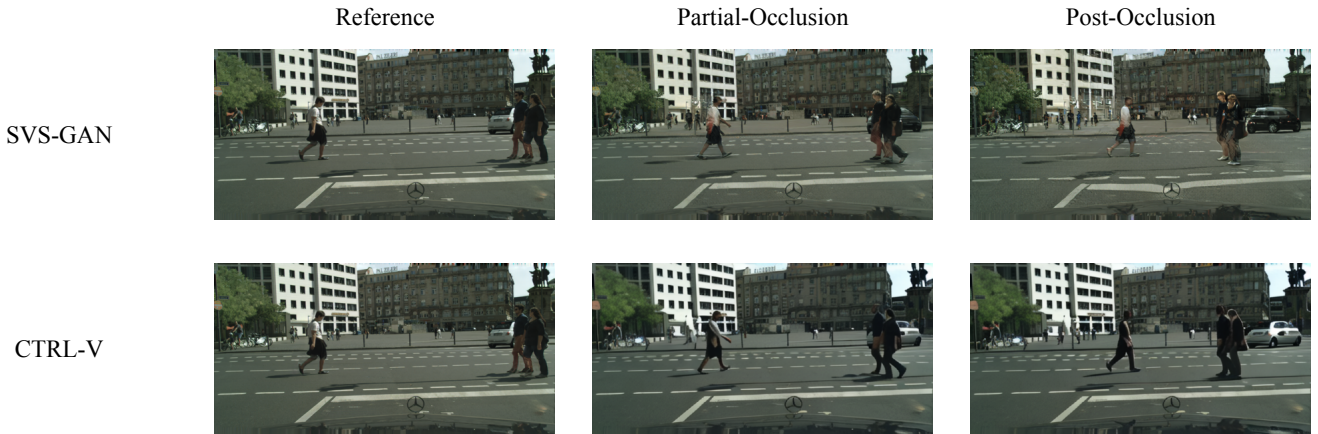


Figure 4. Failure case under long occlusion on Cityscapes. The white car is correctly rendered in early frames but changes color after being occluded by pedestrians in SVS-GAN, due to its single-frame temporal context. CTRL-V maintains appearance consistency within its 30-frame generation window, though consistency may break across windows since each segment is re-anchored to its own reference frame.

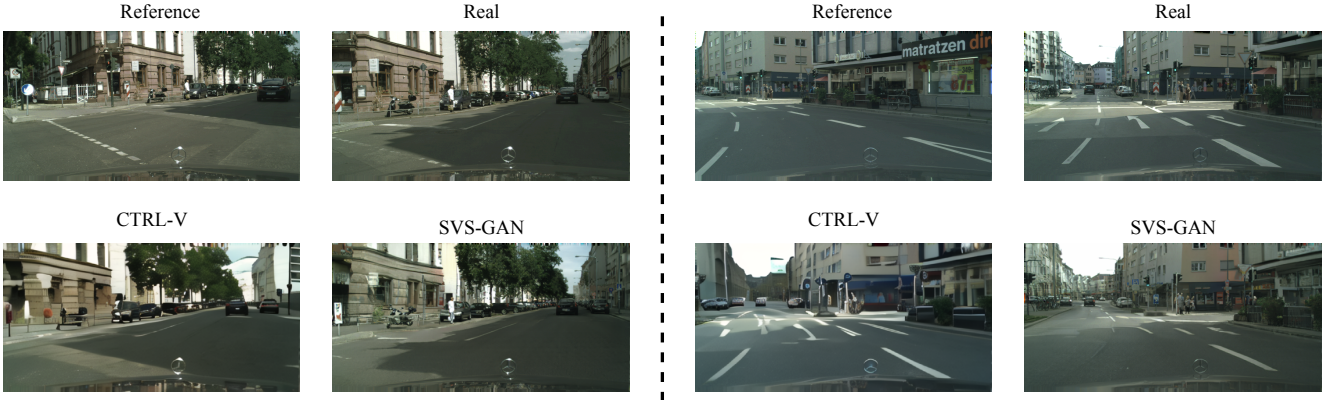


Figure 5. Turning maneuvers with rapid viewpoint change on Cityscapes. Two scenes are shown. For each scene, the top row shows the *Reference* frame and the corresponding *Real* target after a sharp turn; the bottom row shows **CTRL-V** and **SVS-GAN** at the same time step. Large, previously unseen regions appear after the turn. CTRL-V tends to fill these regions more aggressively but with an off-domain, stylized look, while SVS-GAN preserves previously visible actors and Cityscapes style, though it can under-synthesize newly revealed content.

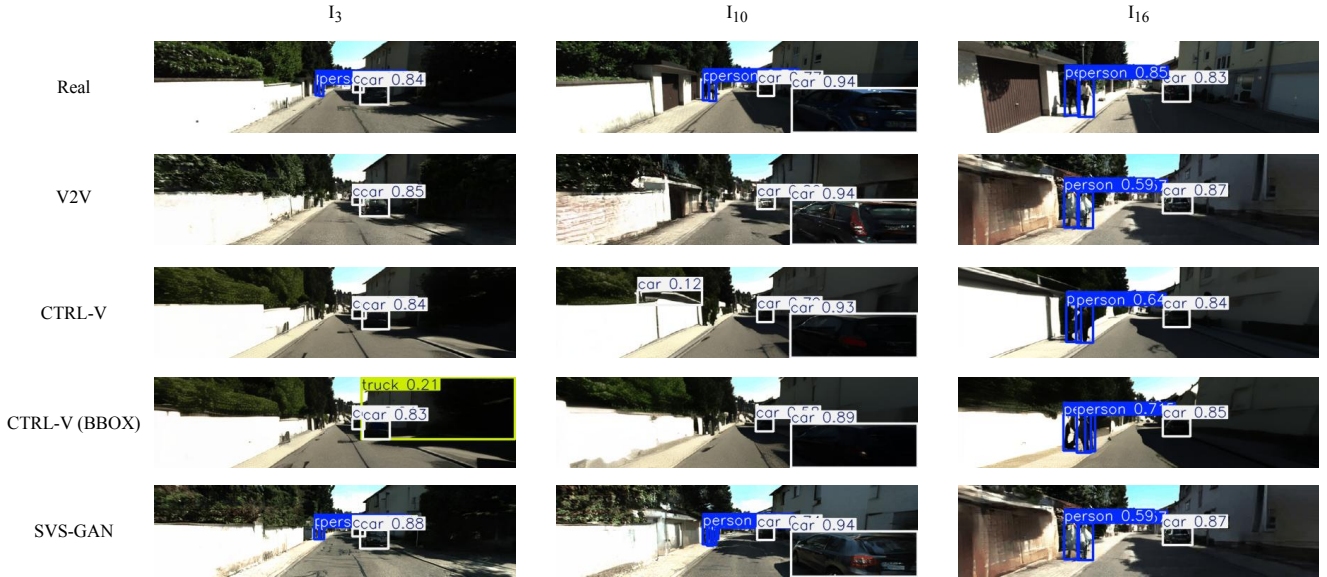


Figure 6. YOLOv8 detections on KITTI-360 sequences at frames I_3 , I_{10} , and I_{16} . While all methods capture vehicles with high confidence, **only SVS-GAN** consistently generates pedestrians early in the rollout, allowing YOLO to detect them well before other methods. This early and stable preservation of small, safety-critical actors highlights the utility of SVS-GAN for data augmentation and closed-loop training in autonomous driving.