LEARNING MAXIMALLY COMPUTABLE VISUAL REPRESENTATIONS FOR CONTROLLABLE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

3D reconstruction provides explicit geometry and accurate transformation of observed content, but requires dense inputs and often degrades under large viewpoint changes. In contrast, generative world models can synthesize realistic future observations, yet their data-driven dynamics make precise geometric control and long-horizon consistency difficult to guarantee. We introduce **Maximally Computable Visual Representation (MCVR)**, which combines explicit reconstruction with generative modeling. Rather than predicting the entire future visual state, MCVR learns a persistent representation whose physically supported content is deterministically propagated under known camera and object transformations, while learned generation is reserved for unrecoverable content. This enables accurate object manipulation and geometry-aware generation while reducing unnecessary prediction. Experiments on autonomous-driving scenes show that MCVR achieves a strong trade-off between visual sufficiency and physical computability, improving controllability and geometric consistency under large motion and long horizons.

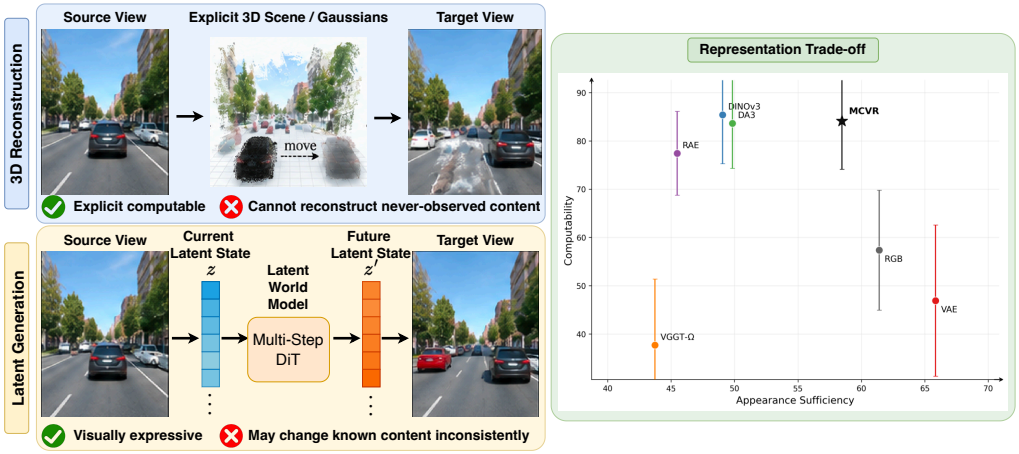

Figure 1: **State computability versus representation sufficiency.** Computability measures how accurately persistent scene state can be updated using known physical transformations without learned future prediction, while sufficiency measures how much scene information the representation retains for visual reconstruction and downstream tasks. MCVR targets the regime where both are high.

1 INTRODUCTION

Controllable visual world models must satisfy two competing requirements. They should preserve the physical state of a scene under commanded motion, while also synthesizing visual content that is not directly observable from the current view. Existing approaches largely address these requirements from opposite directions. Explicit 3D reconstruction represents observed content in physical space and can transform it accurately under camera or object motion, but cannot recover scene content that was never observed. Latent generative models can synthesize such missing content, but

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

typically predict the entire future visual state, including changes already determined by known geometry. As illustrated in Fig. 1, reconstruction can move a vehicle precisely yet expose unsupported regions behind it, whereas latent generation can fill the scene but may unnecessarily alter content that should have remained unchanged.

This suggests a simple question: *how much future visual evolution actually needs to be predicted?* We define *state computability* as the accuracy of updating persistent visual state under known physical transformations without learned prediction:

$$S_t \approx \rho(g_{s\to t}, S_s).$$

Geometry alone does not ensure cross-view feature consistency. We therefore propose **Maximally Computable Visual Representation (MCVR)**, jointly optimized for visual reconstruction and physical transport consistency, balancing computability with visual sufficiency. This makes computability a property of the learned representation. MCVR reformulates future generation by separating physically determined scene evolution from genuinely unknown visual content. Persistent scene information is anchored to the physical world, allowing static regions and moving objects to be updated directly from known camera and object motion. Instead of predicting the entire future representation, the model invokes learned generation only for newly revealed or otherwise unsupported content, after which a view-conditioned renderer produces the target image. In this way, future generation follows a simple principle: *compute known state, generate missing state, and render the observation*.

Our contributions are threefold. First, we formulate physical computability and visual sufficiency as joint representation objectives and introduce MCVR to study their trade-off. Second, we develop a persistent-state generation framework that explicitly propagates static and object-centric visual state while reserving learned generation for uninstantiated content. Third, we evaluate the complete causal chain from representation computability to prediction burden and future generation, showing when visual state can be computed, how much neural prediction can be eliminated, and how this affects geometric control and long-horizon consistency.

2 Related Work

2.1 3D Reconstruction and Novel View Synthesis

Reconstruction-based methods synthesize novel views using continuous radiance fields, as in NeRF (Mildenhall et al., 2020), or explicit Gaussian primitives, as in 3D Gaussian Splatting (Kerbl et al., 2023). Neural Scene Graphs (Ost et al., 2021) and UniSim (Yang et al., 2023) separate backgrounds from actors for independent manipulation, while Street Gaussians (Yan et al., 2024), DrivingGaussian (Zhou et al., 2024b), and OmniRe (Chen et al., 2025) extend explicit representations to dynamic driving scenes. These methods provide strong geometric control, but typically rely on multi-frame observations, scene-specific reconstruction, and sufficient view coverage.

EmerNeRF (Yang et al., 2024), LERF (Kerr et al., 2023), and Feature 3DGS (Zhou et al., 2024a) enrich 3D/4D representations with pretrained visual features. DUSt3R (Wang et al., 2024a) and MASt3R (Leroy et al., 2024) infer geometry and correspondence, while pixelSplat (Charatan et al., 2024) and MVSplat (Chen et al., 2024) predict Gaussian representations from sparse views without per-scene optimization.

2.2 Latent Generative Models and Controllable World Models

Latent diffusion (Rombach et al., 2022) and diffusion transformers (Peebles & Xie, 2023) provide priors for synthesizing unseen content and visual dynamics, with large-scale video models such as Sora (Brooks et al., 2024) extending these approaches to video generation. For autonomous driving, GAIA-1 (Hu et al., 2023) and GAIA-2 (Russell et al., 2025) condition generation on actions and scene context; DriveDreamer (Wang et al., 2024b) and DriveDreamer-2 (Zhao et al., 2025b) introduce maps, object states, trajectories, and language control; and MagicDrive (Gao et al., 2024a) and MagicDrive-V2 (Gao et al., 2025) strengthen geometric and multi-view conditioning. DrivingDiffusion (Li et al., 2024) and Panacea (Wen et al., 2024) generate layout-conditioned multi-view videos, while X-Drive (Xie et al., 2025) jointly synthesizes camera and LiDAR observations.

Drive-WM (Wang et al., 2024c) and Vista (Gao et al., 2024b) model future driving scenes. UniFuture (Liang et al., 2026) jointly predicts future RGB and depth from a current image, RAYNOVA (Xie et al., 2026) uses scale-temporal autoregression with ray-based spatial encoding, and OmniDreams (NVIDIA et al., 2026) targets real-time, closed-loop driving simulation.

Generative priors also complement explicit geometry. DriveDreamer4D (Zhao et al., 2025a) uses generated novel-trajectory videos to improve 4D reconstruction, while WorldSplat (Zhu et al., 2026) predicts feed-forward 4D Gaussians and refines their renderings with video diffusion.

2.3 Visual Representations for Understanding and Generation

Vision Transformers (Dosovitskiy et al., 2021) provide a common patch-based representation for visual learning. Building on this paradigm, the DINO family (Caron et al., 2021; Oquab et al., 2024; Siméoni et al., 2026), CLIP (Radford et al., 2021), and MAE (He et al., 2022) learn transferable features through self-supervised, language-supervised, and reconstruction-based objectives. In parallel, geometry-oriented models such as VGGT (Wang et al., 2025), Depth Anything 3 (Lin et al., 2026), and UniDepthV2 (Piccinelli et al., 2025) recover dense scene geometry, camera parameters, or metric depth from visual observations, enabling lifting, reprojection, and other geometry-aware operations.

Generative representations, including VAEs (Kingma & Welling, 2014), VQ-VAE (van den Oord et al., 2017), VQGAN (Esser et al., 2021), and Representation Autoencoders (Zheng et al., 2026), instead emphasize preserving information necessary for faithful reconstruction and synthesis. However, existing visual representations are typically optimized either for semantic understanding, geometric prediction, or generative fidelity, rather than for directly updating rich visual state under known physical transformations. MCVR bridges these objectives by learning a representation that remains visually sufficient for generation while being explicitly transportable under scene motion.

3 Method

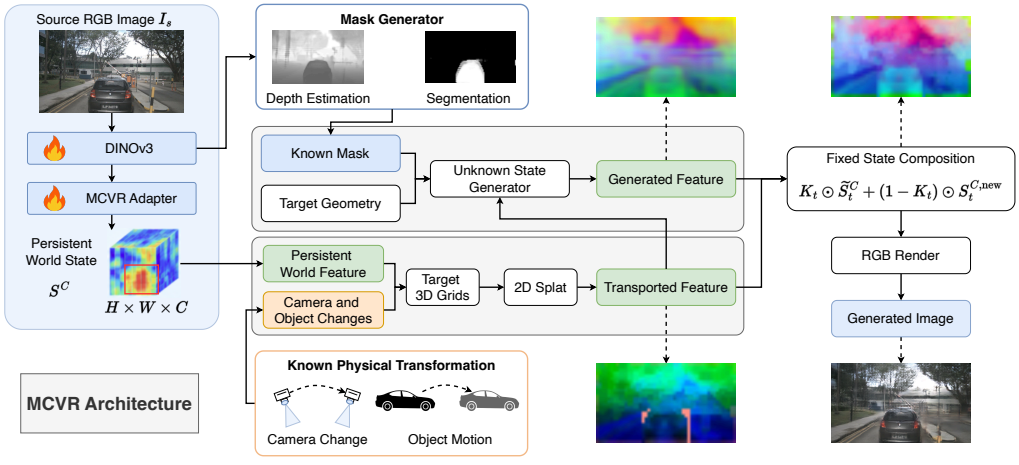

Figure 2: **Overview of the MCVR architecture.** A source RGB image is encoded by DINOv3 and the MCVR adapter into a persistent visual world state. Depth and segmentation heads estimate geometric support and the known-region mask. Given known camera and object transformations, persistent features are lifted into target 3D grids, transported, and splatted back to the image plane. Regions without reliable transported support are completed by an unknown-state generator. The transported and newly generated features are then composed using a fixed support mask and decoded by the RGB renderer to produce the target image.

3.1 Problem Formulation

Let I_t denote the RGB observation of a scene at time t, and let $S_t^C = E_\theta(I_t)$ denote its persistent visual representation. For two observations I_s and I_t of the same scene related by a known physical transformation $g_{s\to t}$, we call the representation *computable* if its instantiated content can be propagated by an explicit operator ρ such that $\rho(g_{s\to t}, S_s^C) \approx S_t^C$, without relying on a learned future-state transition.

Computability alone, however, admits trivial solutions such as purely geometric representations. We therefore additionally require S_t^C to retain sufficient visual information for reconstructing the corresponding observation: $\hat{I}_t = R_\phi(S_t^C, G_t)$, where G_t denotes the observation geometry at time t.

We therefore formulate MCVR as learning the most visually informative representation subject to a computability constraint:

$$\begin{aligned} S^{C*} = \arg\min_{S^C} \quad & \mathcal{E}_{\text{vis}}(S^C) \\ \text{s.t.} \quad & \mathcal{E}_{\text{comp}}(S^C) \le \epsilon_{\text{comp}}, \end{aligned} \tag{1}$$

where $\mathcal{E}_{\text{vis}}$ measures reconstruction error and $\mathcal{E}_{\text{comp}}$ measures cross-view inconsistency after explicit physical transformation. This formulation captures the central objective of MCVR: *maximize what can be computed while minimizing what must be predicted*.

3.2 Learning the Persistent Representation

We instantiate E_θ using a frozen visual foundation encoder followed by a trainable MCVR adapter, $S_t^C = A_\theta(E_D(I_t))$. S^C is the only persistent visual representation in our framework. Unlike approaches that separate a transportable state from a non-transportable appearance code, we require the complete representation to satisfy the same computability constraint. The representation is therefore encouraged to preserve persistent semantics, identity, geometry, color, texture, material, and other visual properties associated with physical scene content.

We jointly optimize two complementary objectives. First, same-view reconstruction enforces visual sufficiency, $\mathcal{L}_{\text{rec}} = \ell_{\text{img}}\left(R_\phi(S_t^C, G_t), I_t\right)$. Second, for a source–target pair, the source representation is explicitly transported and compared against an independently encoded target state,

$$\widetilde{S}_t^C = \rho(g_{s\to t}, S_s^C), \qquad S_t^{C,*} = \text{sg}[E_\theta(I_t)], \qquad \mathcal{L}_{\text{comp}} = \frac{\sum_p W_t(p)\, d_C(\widetilde{S}_t^C(p), S_t^{C,*}(p))}{\sum_p W_t(p) + \epsilon}.$$

Here, W_t selects geometrically valid correspondences and d_C is a feature distance. The representation objective is $\mathcal{L}_{\text{repr}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{comp}}\mathcal{L}_{\text{comp}}$. Thus, visual reconstruction prevents S^C from collapsing to geometry alone, while cross-view computability encourages the same visual content to remain attached to persistent physical entities rather than to a particular image observation.

3.3 Persistent-State Memory-Based Generation

After learning S^C, we ground it to explicit 3D support and construct a persistent scene memory. Static content is stored in world coordinates, while dynamic objects are stored in canonical object-local coordinates:$\mathcal{M}^C = \mathcal{M}^{C,S} \cup \bigcup_i \mathcal{M}_i^{C,D}$. At future time t, known camera and object states $\mathcal{G}_t$ explicitly transform the geometric support of the memory. The associated persistent feature itself is preserved: $\mathbf{x}_{ij,t}^w = T_{i,t}^{o\to w}\mathbf{x}_{ij}^o, \mathbf{s}_{ij,t}^C = \mathbf{s}_{ij}^C$. Projection and visibility reasoning then produce the portion of the target state that can be directly recovered from memory,$(\widetilde{S}_t^C, M_t^{\text{cov}}, Q_t, G_t) = \rho(\mathcal{M}^C, \mathcal{G}_t)$, where M_t^{cov} denotes memory coverage and Q_t its confidence. Geometry can only compute state that has already been instantiated. Newly visible surfaces therefore require learned completion. Using a soft known-state confidence K_t, we predict missing state as $S_t^{C,\text{new}} = G_\psi(\widetilde{S}_t^C, G_t, 1 - K_t)$, and compose the visible target representation as

$$S_t^{C,\text{vis}} = K_t \odot \widetilde{S}_t^C + (1 - K_t) \odot S_t^{C,\text{new}}. \tag{2}$$

Finally, the target image is rendered directly from the visible persistent state and target-view geometry,$\hat{I}_t = R_\phi(S_t^{C,\text{vis}}, G_t)$. The generated RGB image is therefore an observation of the persistent world state, rather than the state used to drive the next prediction. For long-horizon generation, each target state is recomputed from persistent memory and the commanded scene geometry instead of recursively propagating previously generated images. Newly instantiated state can be written back into memory and subsequently treated as persistent content.

4 EXPERIMENTS

Our experiments test the causal chain behind MCVR: whether a visual representation can remain informative while being explicitly transported, where such transport is reliable and how much future state can be computed, and whether this improves target-frame synthesis. We organize the evaluation around three questions: **(RQ1)** Can a representation preserve visual information while remaining physically computable? **(RQ2)** Where is explicit transport reliable, and how does the computable area change with viewpoint and horizon? **(RQ3)** Does computable state improve target-frame synthesis under joint ego–agent motion?

4.1 EXPERIMENTAL PROTOCOL

Dataset and pair construction. We evaluate on nuScenes using the official train/validation split. Each sample contains a source image I_s, a target image I_t, calibrated source/target cameras, and annotated 3D tracks for dynamic objects. We construct source–target pairs over multiple temporal gaps and use the same pairs, geometry, projection, visibility reasoning, and valid correspondence masks for all representation-level comparisons. Dynamic actors are transformed using their annotated object poses, while static content is transformed in the world frame.

Representation baselines. To isolate the representation itself, we compare RGB patches, a reconstruction-oriented VAE (**VAE**), frozen **DINOv3**, **RAE**, intermediate features from **VGGT** and **Depth Anything 3 (DA3)**, and **Recon-only**, which uses the same MCVR adapter and renderer but removes $\mathcal{L}_{\text{comp}}$. All representations are transported by the same geometry operator and evaluated with the same target correspondences. For visual sufficiency, each representation is paired with the same-capacity probe renderer trained with the same data and optimization budget; this limits confounding from native decoder capacity.

Generation baselines. For target-frame synthesis under joint ego–agent motion, we compare against **Street Gaussians** (Yan et al., 2024) and **MagicDrive-V2** (Gao et al., 2025). Street Gaussians uses additional observations and per-scene optimization; all target evaluation images are excluded from scene fitting. MagicDrive-V2 is a structure-conditioned generation reference. We report its native image history, camera coverage, and geometric inputs separately rather than assuming they match MCVR's single-image setting.

We report representation-level scores, target-frame quality, and control fidelity. Metric formulas, thresholds, aggregation, and evaluation details are summarized in Appendix A.3.

4.2 VISUAL SUFFICIENCY AND SINGLE-STEP COMPUTABILITY

We first isolate the representation from end-to-end image synthesis. A source representation is encoded, transported once with the common physical operator, and compared with an independently encoded target representation. In parallel, the same representation is decoded to measure how much visual information it preserves. Table 1 therefore evaluates three complementary properties: same-view visual sufficiency, intrinsic feature transportability, and the quality of RGB rendered only from physically supported transported state.

Table 1 reveals a clear mismatch between visual sufficiency and intrinsic transportability. MCVR (Recon-only) preserves substantially more visual information than the pretrained representations, achieving the best same-view reconstruction quality with 27.49 dB PSNR and 0.549 LPIPS, as well as the strongest transported rendering on physically supported regions (23.71 dB PSNR and 0.584 LPIPS). However, this reconstruction advantage does not translate into greater feature-level

Table 1: Single-step visual sufficiency and transportability on nuScenes.

Representation	Visual sufficiency		Intrinsic transportability		Transported rendering	
	$\text{PSNR}_{\text{self}}$ ↑	$\text{LPIPS}_{\text{self}}$ ↓	E_{comp} ↓	Comp@.9↑	$\text{PSNR}_{\text{known}}$ ↑	$\text{LPIPS}_{\text{known}}$ ↓
RAE	22.65	0.710	0.279	31.7	22.80	0.671
DINOv3	24.14	0.673	0.168	50.6	22.99	0.661
VGGT	21.97	0.738	0.468	40.0	17.56	0.727
DA3	23.83	0.657	0.135	57.8	22.71	0.653
MCVR(Recon-only)	**27.49**	**0.549**	0.210	36.0	**23.71**	**0.584**

E_{comp} and Comp@.9 use negative-calibrated compatibility rather than raw cross-representation cosine. Unknown cells are filled by a fixed zero token and are excluded from RGB metrics.

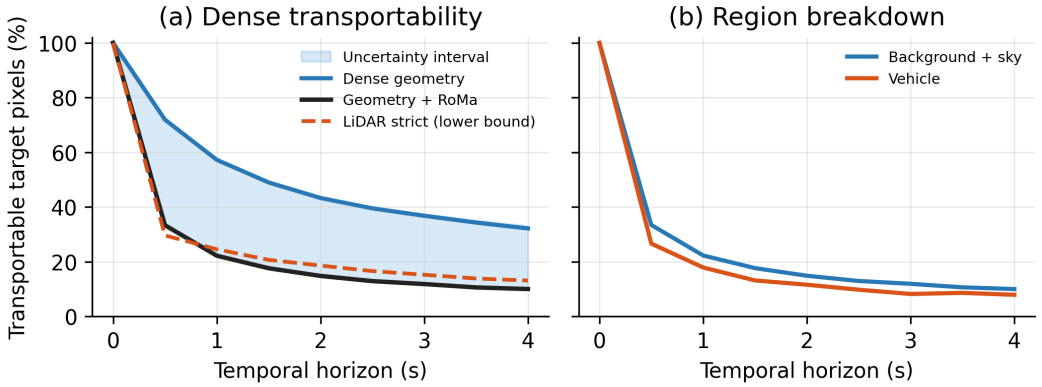

Figure 3: **Transportability of persistent visual state over time.** (a) Fraction of target pixels that remain transportable as the temporal horizon increases. Dense geometry provides an optimistic estimate, while geometry combined with RoMa correspondence gives a more conservative estimate close to the strict LiDAR-based lower bound; the shaded region indicates the resulting uncertainty interval. (b) Region-wise breakdown of transportability for background/sky and vehicles, showing a rapid decrease in reusable scene content as viewpoint and object motion accumulate over time.

computability: DA3 achieves the lowest E_{comp} (0.135) and the highest Comp@.9 (57.8%), followed by DINOv3, while MCVR (Recon-only) reaches only 36.0% Comp@.9. These results demonstrate that optimizing a representation for reconstruction alone does not make its latent state physically transportable, motivating the explicit computability objective introduced in MCVR.

4.3 Where and How Much State Can Be Transported?

Transportability has two different failure modes: *representation mismatch*, where the same physical content changes its feature after transport, and *missing support*, where target content was never instantiated in the source view.

As Figure 3 separates these effects. First, we decompose valid correspondences into static background, rigid dynamic objects, and occlusion-boundary regions, and report Comp@.9 for each type. Newly revealed pixels are excluded from this intrinsic-consistency score because no source correspondence exists. Second, we measure R_{comp}, the target area that can be directly recovered from persistent state. This area is reported as viewpoint change and temporal horizon increase, directly quantifying how much neural prediction remains necessary.

Figure 3 shows that most directly reusable visual state is concentrated within the first second of temporal evolution. Transportability drops sharply from 100% at the source frame to roughly 20–25% under the conservative correspondence estimates by 1 s, although dense geometry provides a higher upper bound of about 57%. Beyond 1 s, the remaining transportable fraction decreases more gradually, reaching only about 10–15% conservatively at 4 s. The same trend holds across scene regions: background and sky remain slightly more transportable than vehicles, but both decay rapidly as camera and object motion accumulate. These results suggest that explicit state transport is

Table 2: Target-frame synthesis under joint ego and agent motion.

Method	Observed input	Scene opt.	PSNR↑	SSIM↑	LPIPS↓	CenterErr↓	BoxIoU↑	DetRecall↑
MagicDriveV2 (Gao et al., 2025)	Native†	No	14.21 ± 0.66	0.394 ± 0.034	0.519 ± 0.021	1.75 ± 0.19	0.556 ± 0.029	0.986 ± 0.018
MCVR	1 RGB	No	21.07 ± 0.60	0.567 ± 0.036	0.286 ± 0.014	2.10 ± 0.60	0.517 ± 0.068	0.809 ± 0.082

† Native conditioning is not assumed to match a single source image; report the exact history, input cameras, map, and geometry for each released configuration. All rows use the same 150 pairs from 30 scenes; values are scene-balanced means $\pm$ 95% bootstrap CI half-widths. Missed targets contribute zero to BoxIoU; CenterErr is conditional on successful matching and must be interpreted together with DetRecall. "–" denotes a pending measurement; N/A denotes a quantity that is not applicable.

Table 3: Geometry-head accuracy on nuScenes validation.

(a) Metric depth against projected LiDAR

Method	Support	Points	MAE↓	AbsRel (%)↓	δ_1 (%)↑
MCVR depth head	valid	125.43M	2.15	12.68	85.67
UniDepthV2-L + GT K	valid	125.43M	2.14	13.52	87.36

(b) Movable segmentation against nuScenes lidarseg

Method	Support	Samples	Mov. IoU↑	F1↑	AP↑
MCVR segmentation head	sparse GT	125.80M	80.02	88.90	96.43

(c) Movable segmentation against SAM2–box pseudo-labels

Method	Support	Samples	Mov. IoU↑	Boundary F1↑	F1↑
MCVR segmentation head	dense pseudo	52.76M	88.64	88.83	93.98

"–" marks DA3-dependent results pending full-validation evaluation. Depth is unaligned; δ, IoU, F1, and AP are percentages. DA3 and SAM2–box provide pseudo-labels, not sensor ground truth.

most effective over short horizons, while progressively more content must be adapted or generated as the prediction horizon increases. (Metric definitions are in Appendix A.3.)

4.4 Target-Frame Synthesis

We next test whether representation-level computability improves synthesis under joint camera and object motion. Table 2 evaluates individual target frames.

Target-frame synthesis under joint ego–agent motion. Given the source observation and recorded target ego and agent poses, each method synthesizes the corresponding target image. Both camera motion and non-ego vehicle motion are included; static background content remains fixed in world coordinates. Using recorded motions provides real target RGB for paired evaluation, unlike arbitrary counterfactual edits with no observed image ground truth. The same source–target pairs and target states are used for all methods. Extra observations, geometry, and scene optimization used by reconstruction references are disclosed separately. For structure-conditioned generators, controls are derived from the same target states; target RGB is never supplied as an appearance condition.

Qualitative comparison. Figure 4 qualitatively compares controllable future-view generation under a 3-meter forward motion and 1.5-meter lateral motions to the left and right. MCVR preserves the global scene layout and the relative position and orientation of the foreground vehicle more consistently with the reference views across different motion commands. In contrast, the baseline methods exhibit larger deviations in object placement and scene structure, particularly under lateral viewpoint changes. These results highlight the benefit of explicitly transporting persistent scene state according to the commanded geometry rather than relying entirely on learned synthesis. MCVR still shows noticeable loss of high-frequency appearance details and some blurring, indicating that improving generation fidelity remains an important direction for future work.

Downstream capbility Table 3 separately measures the depth and movable-object segmentation used to construct transportable primitives. As shown here in the table 3, the learned representation improved over depth and segmetanion psudo label due to the joint training.

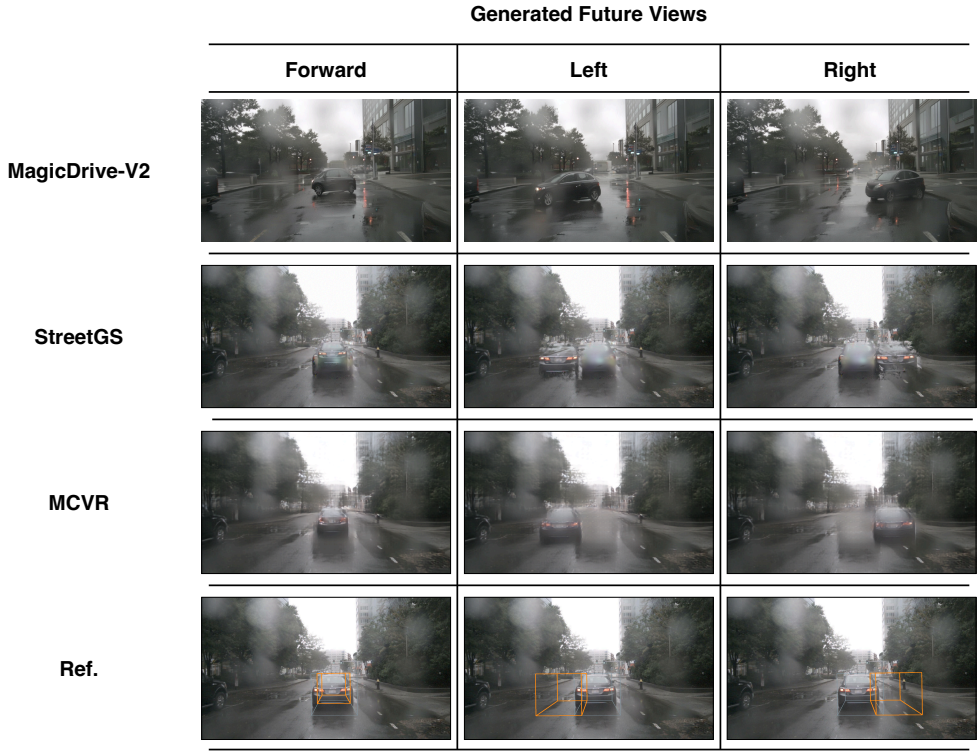

Figure 4: Qualitative comparison of transported-only representations and target images.

Table 4: Fixed-weight inference-time transport diagnostics on nuScenes.

Operator change	$\Delta\text{PSNR}_{\text{full}}$	$\Delta\text{LPIPS}_{\text{full}}$	$\Delta\text{PSNR}_{\text{actor}}$	$\Delta\text{LPIPS}_{\text{actor}}$
Baseline → fine transport	-0.463	+0.002	-0.577	+0.001
Fine → confidence/reveal	-0.619	+0.239	-0.257	+0.067
Combined → strict rescue	-0.001	+0.000	-0.001	+0.000
Combined → candidate replay	-0.004	-0.001	-0.008	+0.000
Strict rescue → surface pooling	-0.240	+0.021	-0.171	+0.006
Replay → surface pooling	-0.245	+0.022	-0.184	+0.006

Differences are scene-balanced means of paired changes (after − before); positive ΔPSNR and negative ΔLPIPS are favorable. "Actor" denotes a fixed, baseline-predicted actor-vicinity mask, not ground-truth instance pixels. All rows use the same checkpoint and are not training ablations.

4.5 Ablation Study

Table 4 tests whether post-hoc changes to transport improve target-frame rendering. We keep one trained Stage-5 checkpoint fixed and evaluate the same 150 Table 2 pairs (five per scene in 30 scenes). Each row compares the named inference settings on identical pairs and fixed evaluation masks; the reported difference is the latter setting minus the former, averaged within each scene and then across scenes. The native output includes completion and the shared renderer. Thus these are *inference-time diagnostics*, not retrained component ablations.

No tested change raises full-frame PSNR with the checkpoint held fixed. Fine transport reduces it by 0.463 dB with almost unchanged LPIPS; adding evidence-based confidence and reveal protection reduces it by a further 0.619 dB and increases LPIPS by 0.239. Strict background rescue and capped candidate replay have negligible net effects, whereas surface pooling reduces PSNR on both rescue branches. The actor-vicinity results follow the same PSNR trend. These results show that a finer physical operator cannot simply be substituted into a generator and renderer trained with a different transport distribution; they do not measure the effect of retraining MCVR with these operators. We

therefore retain the trained inference policy for the reported model. Protocol and operator definitions are in Appendix A.3.

Summary. The experiments are designed to establish a complete causal chain rather than only final-frame quality: MCVR first changes the representation itself (Table 1 and Fig. 4); this changes where and how much state can be physically reused (Fig. 3); target-frame synthesis tests the effect of compute–generate separation (Table 2). Table 4 tests the sensitivity of a fixed model to inference-time transport changes.

5 Conclusion and Future Work

Our results suggest that foundation vision models encode substantially richer scene information than is typically exploited by downstream generation pipelines. With an appropriate representation and prompting mechanism, this knowledge can be distilled into a physically meaningful state space that supports controllable and long-horizon world modeling.

Future work will focus on improving fine-grained visual fidelity and further reducing generation cost and latency. More broadly, this framework represents an initial step toward transferring the knowledge of foundation vision models into generative world models, while simultaneously preserving representations that remain useful for downstream visual perception tasks.

AI use statement

In this work, we used generative AI tools for [implement methods, assist with translation]. We have not used generative AI tools for [Generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, implement methods, support qualitative and thematic data analysis, interpret results.], and [assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experimentsclean and reformat dataset,] are not applicable to this work.

We have reviewed all AI-assisted work to ensure its correctness and appropriateness. LLM-generated code was independently verified and tested for correctness by two authors. Intermediate outputs and results produced during implementation were carefully inspected to confirm that the code behaved as intended and that the resulting outputs matched the specified requirements. AI-assisted translation was reviewed and revised by the authors to ensure technical accuracy, consistency with the original meaning, and appropriate academic wording before inclusion in the manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian Splats from Image Pairs for Scalable Generalizable 3D Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19457–19467, 2024.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3D Gaussian Splatting from Sparse Multi-view Images. In *Computer Vision – ECCV 2024*, pp. 370–386. Springer, 2024. doi: 10.1007/978-3-031-72664-4_21.

Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, Li Song, and Yue Wang. OmniRe: Omni Urban Scene Reconstruction. In *International Conference on Learning Representations (ICLR)*, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming Transformers for High-Resolution Image Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12873–12883, 2021.

Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. MagicDrive: Street View Generation with Diverse 3D Geometry Control. In *International Conference on Learning Representations (ICLR)*, 2024a.

Ruiyuan Gao, Kai Chen, Bo Xiao, Lanqing Hong, Zhenguo Li, and Qiang Xu. MagicDrive-V2: High-Resolution Long Video Generation for Autonomous Driving with Adaptive Control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28135–28144, 2025.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A Generalizable Driving World Model with High Fidelity and Versatile Controllability. In *Advances in Neural Information Processing Systems*, volume 37, pp. 91560–91596. Curran Associates, Inc., 2024b. doi: 10.52202/079017-2906.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, 2022.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A Generative World Model for Autonomous Driving. *arXiv preprint arXiv:2309.17080*, 2023. doi: 10.48550/arXiv.2309.17080.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. doi: 10.1145/3592433.

Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. LERF: Language Embedded Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19729–19739, 2023.

Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding Image Matching in 3D with MASt3R. In *Computer Vision – ECCV 2024*, pp. 71–91. Springer, 2024. doi: 10.1007/978-3-031-73220-1_5.

Xiaofan Li, Yifu Zhang, and Xiaoqing Ye. Drivingdiffusion: Layout-guided multi-view driving scenarios video generation with latent diffusion model. In *European Conference on Computer Vision*, 2024.

Dingkang Liang, Dingyuan Zhang, Xin Zhou, Sifan Tu, Tianrui Feng, Xiaofan Li, Yumeng Zhang, Mingyang Du, Xiao Tan, and Xiang Bai. UniFuture: A 4D Driving World Model for Future Generation and Perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the Visual Space from Any Views. In *International Conference on Learning Representations (ICLR)*, 2026.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Computer Vision – ECCV 2020*, pp. 405–421. Springer, 2020. doi: 10.1007/978-3-030-58452-8_24.

NVIDIA, Aarti Basant, Amlan Kar, Despoina Paschalidou, Fangyin Wei, Francesco Ferroni, Guillermo Garcia Cobo, Haithem Turki, Huan Ling, Jaewoo Seo, James Lucas, Jay Zhangjie Wu, Jialiang Wang, Jonathan Lorraine, Jun Gao, Kai He, Katarina Tothova, Kevin Xie, Michał Tyszkiewicz, Qi Wu, Riccardo de Lutio, Ruilong Li, Sanja Fidler, Seung Wook Kim, Tianchang Shen, Tianshi Cao, Tobias Pfaff, William Lew, Xindi Wu, Xuanchi Ren, Yifan Lu, Yuxuan Zhang, Zan Gojcic, and Zian Wang. NVIDIA OmniDreams: Real-Time Generative World Model for Closed-Loop Autonomous Vehicle Simulation. *arXiv preprint arXiv:2606.03159*, 2026. doi: 10.48550/arXiv.2606.03159.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural Scene Graphs for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2856–2865, 2021.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023.

Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniDepthV2: Universal Monocular Metric Depth Estimation Made Simpler. *arXiv preprint arXiv:2502.20110*, 2025. doi: 10.48550/arXiv.2502.20110.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. GAIA-2: A Controllable Multi-View Generative World Model for Autonomous Driving. *arXiv preprint arXiv:2503.20523*, 2025. doi: 10.48550/arXiv.2503.20523.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michael Ramamonjisoa, Francisco Massa, Daniel HAZIZA, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Herve Jegou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D Vision Made Easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709, 2024a.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. DriveDreamer: Towards Real-World-Drive World Models for Autonomous Driving. In *Computer Vision – ECCV 2024*, pp. 55–72. Springer, 2024b. doi: 10.1007/978-3-031-73195-2_4.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the Future: Multiview Visual Forecasting and Planning with World Model for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14749–14759, 2024c.

Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6902–6912, 2024.

Yichen Xie, Chenfeng Xu, Chensheng Peng, Shuqi Zhao, Nhat Ho, Alexander T. Pham, Mingyu Ding, Masayoshi Tomizuka, and Wei Zhan. X-drive: Cross-modality consistent multi-sensor data synthesis for driving scenarios. In *International Conference on Learning Representations*, 2025.

Yichen Xie, Chensheng Peng, Mazen Abdelfattah, Yihan Hu, Jiezhi Yang, Eric Higgins, Ryan Brigden, Masayoshi Tomizuka, and Wei Zhan. RAYNOVA: Scale-Temporal Autoregressive World Modeling in Ray Space. *arXiv preprint arXiv:2602.20685*, 2026.

Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street Gaussians: Modeling Dynamic Urban Scenes with Gaussian Splatting. In *Computer Vision – ECCV 2024*, pp. 156–173. Springer, 2024. doi: 10.1007/978-3-031-73464-9_10.

Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, and Yue Wang. EmerNeRF: Emergent Spatial-Temporal Scene Decomposition via Self-Supervision. In *International Conference on Learning Representations (ICLR)*, 2024.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. UniSim: A Neural Closed-Loop Sensor Simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1389–1399, 2023.

Guosheng Zhao, Chaojun Ni, Xiaofeng Wang, Zheng Zhu, Xueyang Zhang, Yida Wang, Guan Huang, Xinze Chen, Boyuan Wang, Youyi Zhang, Wenjun Mei, and Xingang Wang. DriveDreamer4D: World Models Are Effective Data Machines for 4D Driving Scene Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12015–12026, 2025a.

Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. DriveDreamer-2: LLM-Enhanced World Models for Diverse Driving Video Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10412–10420, 2025b. doi: 10.1609/aaai.v39i10.33130.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion Transformers with Representation Autoencoders. In *International Conference on Learning Representations (ICLR)*, 2026.

Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21676–21685, 2024a.

Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. DrivingGaussian: Composite Gaussian Splatting for Surrounding Dynamic Autonomous Driving Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21634–21643, 2024b.

Ziyue Zhu, Zhanqian Wu, Zhenxin Zhu, Lijun Zhou, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Jin Xie, and Jian Yang. WorldSplat: Gaussian-Centric Feed-Forward 4D Scene Generation for Autonomous Driving. In *International Conference on Learning Representations (ICLR)*, 2026.

A Implementation Details

A.1 Key Components

Representation and geometry heads. We resize images to 448×784 and use a frozen DINOv3 ViT-L/16. The normalized outputs of zero-indexed blocks $\{5, 11, 17, 23\}$ are concatenated into $F^D \in \mathbb{R}^{4096\times28\times49}$. The MCVR adapter applies a 1×1 projection to 384 channels, three residual convolutional blocks, and a normalized 256-D output head, producing the only persistent visual state $S^C \in \mathbb{R}^{256\times28\times49}$. An auxiliary linear convolution reconstructs F^D from S^C during representation learning.

Depth and segmentation use the same frozen F^D but have independent parameters. Each head uses a 1×1 projection to 256 channels, three residual 3×3 blocks, and a $2\times$ PixelShuffle upsampler. The depth head predicts positive inverse depth $\hat{\eta}$ with Softplus and converts it to metric depth $\hat{d} = 1/\hat{\eta}$, clipped to $[0.5, 80]$ m. The segmentation head predicts movable-object and boundary logits. Both outputs have resolution 56×98.

Primitive construction and ownership. The 56×98 geometry grid places four support samples inside each DINO patch. For image coordinate u_j, we form ray $r_j = \mathbf{K}_s^{-1}\bar{u}_j$ and homogeneous inverse-depth support $x_j^h = [r_j^\top, \hat{\eta}_j]^\top$; the 256-D state is bilinearly sampled from S_s^C. Each primitive is

$$q_j = (s_j^C, x_j^h, o_j, c_j), \tag{3}$$

where o_j is its layer ownership and c_j its assignment confidence.

Ownership combines predicted object probability, source 3D boxes, and ray–box depth intervals. A primitive is assigned to object i when its object probability is at least 0.65, its ray intersects box i, and its depth is within 2 m of that interval. It is static background when its object probability is at most 0.25 and its ray intersects no object box; all other primitives are uncertain and excluded from transport. Static support is stored in world coordinates, while object support is stored in the corresponding source object-local frame. Thus the source memory contains no RGB or view-dependent latent code.

Physical propagation and target geometry. For each target state $\mathcal{G}_t$, static support remains in the world frame, whereas object support is transformed by its commanded SE(3) pose. All valid support is transformed into the target camera, projected, and bilinearly splatted onto the 28×49 state grid. The parameter-free rasterizer uses a target-view z-buffer (0.75 m tolerance) and confidence-weighted depth decay (0.25 m temperature). It returns transported state $\widetilde{S}_t^C$, coverage M_t^{cov}, confidence Q_t, depth, object likelihood, flow, and source/target viewing directions. A counterfactual projection at the source object pose identifies vacated object support M_t^{reveal}. The only main-path mixing coefficient is

$$K_t = M_t^{\mathrm{cov}} Q_t (1 - M_t^{\mathrm{reveal}}), \tag{4}$$

not a learned Copy/Adapt/Generate route.

The target-aligned geometry tensor is

$$G_t = [\Delta u, \Delta v, \bar{d}, m^{\mathrm{obj}}, M^{\mathrm{cov}}, Q, v_s^\top, v_t^\top, \bar{u}_t^\top] \in \mathbb{R}^{14\times28\times49}, \tag{5}$$

containing normalized image flow (2), log-normalized depth (1), object likelihood (1), coverage and confidence (2), source and target viewing directions (6), and normalized target coordinates (2).

Given these physically computed quantities, a lightweight feature U-Net receives $[K_t\widetilde{S}_t^C, G_t, 1 - K_t] \in \mathbb{R}^{271\times28\times49}$ and predicts $S_t^{C,\mathrm{new}} \in \mathbb{R}^{256\times28\times49}$. The target state is

$$S_t^{C,\mathrm{vis}} = K_t\widetilde{S}_t^C + (1 - K_t)S_t^{C,\mathrm{new}}. \tag{6}$$

After composition, $H_{\mathrm{obs}}(S_t^{C,\mathrm{vis}}, G_t)$ produces a transient 96-D observation code; a residual refiner is enabled from Stage 2. The shared renderer receives only the concatenation of the 256-D persistent state and 96-D observation code. It uses an eight-block ViT (width 512, eight heads), exposes zero-indexed blocks $\{1, 3, 5, 7\}$ to a 128-D DPT coarse-to-fine head, and reconstructs RGB at 448×784. Geometry is therefore consumed by H_{obs} rather than concatenated directly to the decoder input.

During training, the target image is independently encoded to obtain a stop-gradient target state for feature losses, and target RGB provides image supervision. Neither target features nor a target observation code are inputs to the source-to-target prediction path.

A.2 TRAINING

The depth head is initialized from the DA3–LiDAR student. Movable-object supervision combines box-prompted SAM2 masks, projected nuScenes lidarseg, and depth–box consistency. DINOv3 is frozen throughout; the depth and segmentation heads are pretrained before MCVR training and then frozen. Their losses combine sparse metric-depth supervision with DA3 pseudo-depth for depth, and pixel, boundary, and sparse lidarseg supervision for segmentation.

MCVR is trained progressively. Stage 0 learns dense and masked same-view reconstruction; Stage 1 adds physically transported feature and known-region RGB supervision; Stage 2 introduces observation refinement; Stage 3 trains unknown-state completion; Stages 4–5 jointly tune the representation and renderer, with geometry perturbation in Stage 5. The objectives combine feature reconstruction, transport consistency, completion, composed-feature, source reconstruction, and RGB reconstruction losses. The RGB objective uses pixel, structural, edge, and perceptual terms. Exact coefficients and optimizer schedules are implementation settings; the key distinction is the separate supervision of transported and newly generated content.

For sequential generation, the source initializes persistent memory. At each step the model projects and renders the current state, then predicts candidate primitives to write back under confidence and memory-budget constraints. The rollout uses only the source observation, predicted states, and commanded poses; future target RGB is used only as training/evaluation supervision.

A.3 EVALUATION PROTOCOL AND METRIC DEFINITIONS

Table 2 pair construction. We construct a fixed single-frame benchmark from the official nuScenes validation split using ground-truth metadata only. Each source–target pair uses `CAM_FRONT` images separated by 1.0 ± 0.12 s and must contain at least 2 m of ego translation and one vehicle that (i) is present at both timestamps, (ii) translates by at least 1 m, (iii) lies within 40 m of the ego vehicle at both timestamps, (iv) has nuScenes visibility level at least two, and (v) occupies at least 0.2% of the image at both views. Scenes with at least five valid pairs are ranked using

$$d_{\text{scene}} = \sqrt{\frac{\text{median}(\Delta x_{\text{ego}})}{2\,\text{m}} \frac{\text{median}(\Delta x_{\text{veh}})}{1\,\text{m}}}. \tag{7}$$

The 80 eligible scenes are divided into three difficulty tertiles; with a fixed seed, we sample 10 scenes per tertile and select five temporally spread pairs per scene. This produces 30 scenes and 150 pairs. All methods use the same pairs, target camera poses, and target vehicle poses, while target RGB is held out and used only for evaluation. Exact image, control, and aggregation metrics are defined below.

Representation and target-frame metrics. Predictions and targets are resized by Lanczos interpolation to 448×784. For images in $[0, 1]$, PSNR is $10 \log_{10}(1/\text{MSE})$; SSIM uses the standard windowed luminance, contrast, and structure comparison, and LPIPS uses the AlexNet feature distance. Same-view visual sufficiency uses PSNR and LPIPS. For valid source–target correspondences, let $W_t(p)$ be the correspondence-validity weight and $S_t^{C,*}$ the independently encoded target feature. Intrinsic transport error and thresholded compatibility are

$$E_{\text{comp}} = \frac{\sum_p W_t(p)[1 - \cos(\widetilde{S}_t^C(p), S_t^{C,*}(p))]}{\sum_p W_t(p) + \epsilon}, \tag{8}$$

$$\text{Comp@}\tau = \frac{\sum_p W_t(p)\mathbf{1}[\cos(\widetilde{S}_t^C(p), S_t^{C,*}(p)) \geq \tau]}{\sum_p W_t(p) + \epsilon}, \qquad \tau = 0.9. \tag{9}$$

Transport coverage is $R_{\text{comp}} = (HW)^{-1} \sum_p K_t(p)$, with $R_{\text{pred}} = 1 - R_{\text{comp}}$. Target-frame appearance uses PSNR, SSIM, and LPIPS.

For control, a frozen COCO Faster R-CNN ResNet-50-FPN-v2 detector (confidence threshold 0.3) is matched one-to-one to eligible projected target-vehicle boxes using class-compatible Hungarian assignment with cost

$$c_{ij} = 1 - \mathrm{IoU}(b_i, \hat{b}_j) + 0.01 e_{ij}, \qquad e_{ij} = 100 \frac{\|c_i - \hat{c}_j\|_2}{\sqrt{H^2 + W^2}}, \quad e_{ij} \le 20. \tag{10}$$

CenterErr averages e_{ij} over matched boxes; BoxIoU assigns zero to missed eligible targets; DetRecall is the matched fraction. CenterErr is undefined when no target is matched. Vehicle-LPIPS enlarges the target box by 10%, resizes the crop to 224×224, and computes AlexNet LPIPS.

Fixed-weight transport diagnostics (Table 4). We use one Stage-5 checkpoint on the same 150 Table 2 pairs (30 scenes, five pairs each), with the completion generator and renderer enabled for every reported prediction. No variant is retrained. Relative to the checkpoint's baseline transport, *fine* uses a $2\times$ finer rasterization grid; *combined* additionally changes confidence estimation and protects newly revealed background. *Strict rescue* admits background only under the source ownership checks, while *candidate replay* admits a fixed set of uncertain source-only candidates with reliability capped at 0.35; these candidates are not ground-truth ownership. *Surface pooling* selects the largest depth-compatible, same-owner support cluster. Each table row compares the named pair of settings, not every setting against the original baseline. Target RGB is loaded only after source-only predictions and is used solely for metrics. Full-frame and baseline-predicted actor-vicinity masks are held fixed across variants; the latter dilates predicted actor footprints by three feature cells and can include non-vehicle actors. For each metric we average paired differences within a scene and then average the 30 scene means equally.

Depth and segmentation metrics (Table 3). For valid prediction/reference depths $(\hat{d}_i, d_i)$ in meters,

$$\mathrm{MAE} = N^{-1} \sum_i |\hat{d}_i - d_i|, \quad \mathrm{RMSE} = \sqrt{N^{-1} \sum_i (\hat{d}_i - d_i)^2}, \tag{11}$$

$$\mathrm{AbsRel} = N^{-1} \sum_i \frac{|\hat{d}_i - d_i|}{d_i}, \quad \delta_k = N^{-1} \sum_i \mathbf{1}\left[\max\left(\frac{\hat{d}_i}{d_i}, \frac{d_i}{\hat{d}_i} \right) < 1.25^k \right]. \tag{12}$$

SILog is reported as $100\sqrt{\max(\mathbb{E}[z^2] - 0.85\mathbb{E}[z]^2, 0)}$, where $z = \log \hat{d} - \log d$. No scale or shift alignment is applied. Sparse metrics use valid projected LiDAR pixels; the DA3 row has fewer points because it is evaluated on its own valid support. Dense agreement uses the DA3-valid mask, intersected with UniDepthV2 validity for its row. DA3 is pseudo-GT, not sensor ground truth. Segmentation probabilities are thresholded at 0.65; sparse metrics use nuScenes lidarseg and dense metrics use SAM2–box pseudo-labels. IoU, precision, recall, F1, AP, boundary F1, and ECE use their standard binary definitions.

Video metrics and aggregation. Future-generation FID and FVD use a fixed feature extractor, preprocessing, temporal sampling, clip length, and sample count; literature scores from different protocols are not directly comparable. LPIPS and control metrics are also measured at 1 s and 4 s from the last real conditioning frame. All methods share target scenes, timestamps, output camera, and metric resolution; extra conditioning views and histories are disclosed. External models without commanded agent states are evaluated for agreement with recorded futures, not control compliance. Frame metrics are averaged within each scene and then across scenes; control metrics are aggregated over eligible objects per scene. When confidence intervals are reported, they use 10,000 scene-level bootstrap resamples.