Learning to Drive is a Free Gift: Large-Scale Label-Free Autonomy Pretraining from Unposed In-The-Wild Videos

Matthew Strong[1,2†] Wei-Jer Chang[1,3†] Quentin Herau[1] Jiezhi Yang[1]
Yihan Hu[1] Chensheng Peng[1,3†] Wei Zhan[1,3‡]

[1] Applied Intuition [2] Stanford University [3] UC Berkeley

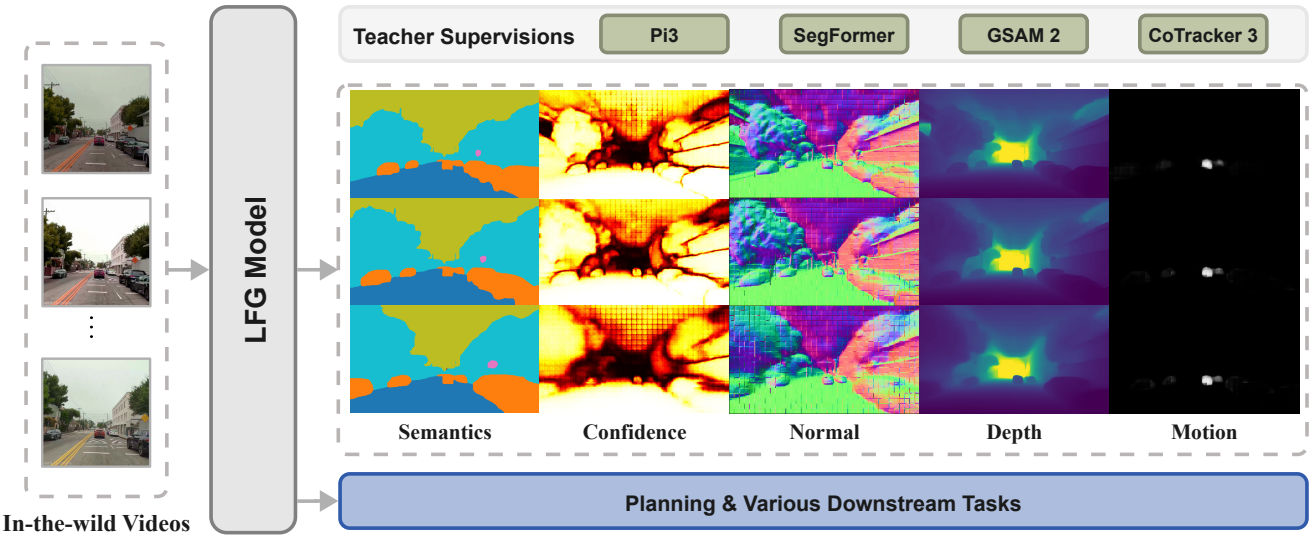

Figure 1. LFG learns a unified pseudo-4D representation of geometry, semantics, motion, and short-term future evolution directly from unposed, unlabeled single-view driving videos. A single feedforward encoder processes observed frames and produces temporally consistent predictions of 3D point maps, camera poses, semantic layouts, confidence, and motion masks for both current and future frames.

Abstract

Ego-centric driving videos available online provide an abundant source of visual data for autonomous driving, yet their lack of annotations makes it difficult to learn representations that capture both semantic structure and 3D geometry. Recent advances in large feedforward spatial models demonstrate that point maps and ego-motion can be inferred in a single forward pass, suggesting a promising direction for scalable driving perception. We therefore propose a label-free, teacher-guided framework for learning autonomous driving representations directly from unposed videos. Unlike prior self-supervised approaches that focus primarily on frame-to-frame consistency, we posit that safe and reactive driving depends critically on temporal context. To this end, we leverage a feedforward architecture equipped with a lightweight autoregressive module, trained using multi-modal supervisory signals that guide the model to jointly predict current and future point maps, camera poses, semantic segmentation, and motion masks. Multi-modal teachers provide sequence-level pseudo-supervision, enabling LFG to learn a unified pseudo-4D representation from raw YouTube videos without poses, labels, or LiDAR. The resulting encoder not only transfers effectively to downstream autonomous driving planning on the NAVSIM benchmark, surpassing multi-camera and LiDAR baselines with only a single monocular camera, but also yields strong performance when evaluated on a range of semantic, geometric, and qualitative motion prediction tasks. These geometry and motion-aware features position LFG as a compelling video-centric foundation model for autonomous driving.

† Work done as intern at Applied Intuition.
‡ Corresponding author: wei.zhan@applied.co

1. Introduction

In-the-wild, ego-centric driving videos available online provide an abundant source of visual data for driving, yet their lack of annotations makes it difficult to learn representations that capture both semantic, temporal structure and 3D geometry. Inspired by the recent success of GPT-style models [5, 20] and DINOv3 [23] trained on massive unlabeled internet corpora, a natural question arises: can we similarly leverage large amounts of raw video to learn geometry and motion aware features for autonomy?

Recent research in autonomy has shown that scaling up improves performance [4, 8, 16], yet most approaches still rely heavily on *labeled* data in the form of expert actions, LiDAR scans, odometry, and semantic annotations. Meanwhile, in-the-wild driving videos are abundant and capture a wide range of visual conditions and traffic situations. Although these videos provide only RGB information, they contain rich visual and motion cues that can be learned. If we aim to build scalable autonomy models capable of producing expressive and actionable representations, they should benefit from large-scale pretraining on unlabeled images and videos.

This motivates the goal of learning structure and motion directly from video. Feedforward 3D reconstruction models already demonstrate that it is possible to estimate camera poses and point maps from unposed image sequences using a single forward pass [26, 28]. Egocentric driving videos provide ideal data for such models, as consecutive frames naturally encode geometry and ego-motion, even with sparse viewpoints. Yet for autonomous driving, a model must ultimately do more: beyond reconstructing the present, *it must predict future motion and geometry*.

Motivated by findings that humans make low-level driving decisions from only a short motion history, we extend the feedforward reconstruction model π^3 [28] to predict future geometry, confidence, and motion. Our model is trained using signals from multiple large-scale models trained on unposed data, which provide complementary cues for geometry, motion, and semantics. By integrating these cues and incorporating segmentation and motion components, the student model learns from in-the-wild driving videos to produce a pseudo-4D output that captures scene structure together with the motion of dynamic agents.

We introduce LFG – **L**earning to drive is a **F**ree **G**ift – a label-free, teacher-guided approach for learning such representations from vision alone. We formulate future prediction as a next-token prediction problem over geometry, motion, and semantic features. A lightweight autoregressive transformer is added after the reconstruction aggregator, enabling a student model trained on a subset of views to benefit from stronger models with access to the full sequence. Supervision comes from several specialized teachers—SegFormer [31] for semantics, SAM2 [13] and CoTracker3 [10] for motion cues —each used in a way that best leverages its strengths on unlabeled driving video.

Unlike large world models that still require a degree of supervised labels [1, 6, 7, 12], LFG focuses on a short-horizon, feedforward formulation that sets a new standard among geometry-aware models for autonomy. On the NAVSIM planning benchmark [4], LFG achieves state-of-the-art performance using *only a single front-camera view*, outperforming multi-view and BEV-based methods such as UniAD [14] and HydraMDP [16], which rely on multiple cameras, LiDAR, or both. LFG pretraining also provides strong sample efficiency: with only 10% labeled data, it achieves competitive planning performance, underscoring the value of large-scale training on unlabeled driving video. Beyond planning, LFG produces geometry- and motion-aware features that transfer effectively to tasks spanning semantics, 3D structure, and decision making, underscoring its broader applicability as a backbone for next-generation autonomous driving systems.

Our main contributions are as follows:

- We propose **LFG**, a label-free, video-centric pretraining framework that learns geometry-, motion-, and semantics-aware representations directly from unposed, single-view driving videos.
- We design a unified architecture built on a pretrained encoder and a causal autoregressive module, enabling short-horizon prediction of point maps, camera poses, semantic layouts, confidence maps, and motion masks under multiple teacher-guided supervision.
- We demonstrate that LFG serves as a strong foundation for autonomy: it achieves state-of-the-art planning performance using only a single front camera, exhibits compelling data efficiency, and transfers effectively across semantic, geometric, and motion tasks. We emphasize that the novelty of LFG lies more within the pretraining paradigm than the model itself.

2. Related Work

Pretraining for Autonomous Driving. Pretraining for autonomous driving has only recently gained traction. Early self-supervised pretraining work such as SelfD [36] and ACO [37] demonstrated that large-scale in-the-wild driving videos can provide supervisory signals for learning semantic and geometric priors without human labels. PPGeo [29] further explored geometry-oriented pretraining using photometric and consistency-based objectives to learn depth and ego-motion. ViDAR [34] proposes to use future point-cloud prediction from historical camera inputs as a unified pretext task. UniPAD [32] introduces a self-supervised learning paradigm that uses 3D volumetric differentiable rendering to implicitly encode continuous 3D

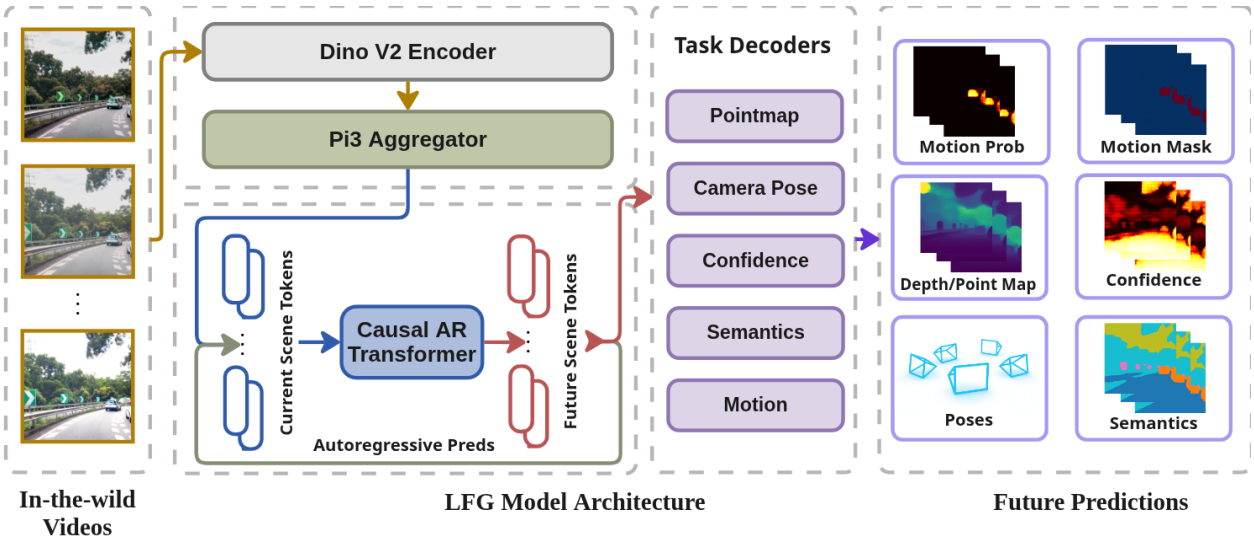

Figure 2. **LFG architecture.** Starting from unposed single-view driving clips, a pretrained π^3 backbone encodes N observed frames into latent scene tokens. A lightweight causal autoregressive transformer rolls out M future tokens, which a shared decoder maps to point maps, camera poses, semantic segmentation, confidence maps, and motion masks for all $N+M$ frames. Multi-modal teachers provide pseudo-supervision, enabling LFG to learn a unified pseudo-4D representation that transfers effectively to downstream planning.

structures. VisionPAD [35] focuses on vision-centric algorithms by leveraging efficient 3D Gaussian Splatting and a multi-frame photometric consistency objective to reconstruct multi-view representations using only images. However, these approaches largely rely on frame-to-frame consistency losses that implicitly assume static scenes, limiting their ability to capture dynamic objects that are central to real driving environments. In contrast, our method is pretrained directly on unlabeled driving video by explicitly modeling *dynamic* geometry, motion cues, and scene semantics, yielding a dense 4D representation that better captures the structure and dynamics of real-world driving.

Geometry-aware vision backbones for driving. Classical 3D reconstruction pipelines in autonomy rely on Structure-from-Motion (SfM) and Multi-View Stereo (MVS) [22], often combined with LiDAR, to triangulate scene points and build dense maps for localization. While effective, these methods are typically tailored per scene and are not naturally suited as general-purpose backbones for large-scale video pretraining. In contrast, recent feedforward approaches [15, 19, 25, 26, 28] amortize reconstruction by predicting point maps, confidence maps, and camera poses for unposed image sequences in a single pass, making them attractive as scalable, geometry-aware backbones for driving. LFG belongs to this family but focuses on *temporal* understanding, producing a pseudo-4D representation of dynamic driving scenes that is well suited for downstream planning and perception.

3. Method

We introduce LFG (shown in Fig. 2), a method for learning a powerful driving-vision model from unposed and unlabelled single view Youtube videos.

3.1. Problem Formulation

We consider the case of learning to drive, where a large parameterized model is given a consecutive sequence $(I_t)_{t=1}^{N}$ of N ego-centric RGB images $I_t \in \mathbb{R}^{3\times H\times W}$, in a variety of driving scenes. The goal is to efficiently predict scene information that is useful for autonomous driving. We posit that this includes both current and short-horizon future information. We posit that this includes current *and* future information in the recent future. Such a model should predict $\mathcal{O}$ relevant modalities of scene information, as well as the future M frames of the scene. Inspired by prior work in label-free pretraining and world models for driving, we choose to predict the following outputs.

LFG processes in-the-wild video through a pretrained encoder and a causal autoregressive transformer to jointly predict current and short-horizon future scene geometry, semantics, and motion. First, our model should predict **point maps** for the ego-view camera over time. $(P_t)_{t=1}^{N+M}$, $P_t : \mathcal{I}(I_t) \to \mathbb{R}^3$, where $\mathcal{I}(I_t)$ maps each pixel in I_t to $P_t(\mathbf{y}) \in \mathbb{R}^3$, which is the 3D world point corresponding to pixel $\mathbf{y}$ at time t.

Second, our model predicts **camera poses**: $(T_t)_{t=1}^{N+M}$, $T_t \in \mathbb{R}^{4\times4}$, where each p_t is a full 4×4 homogeneous transformation matrix encoding both rotation and translation. Such poses define the ego-motion

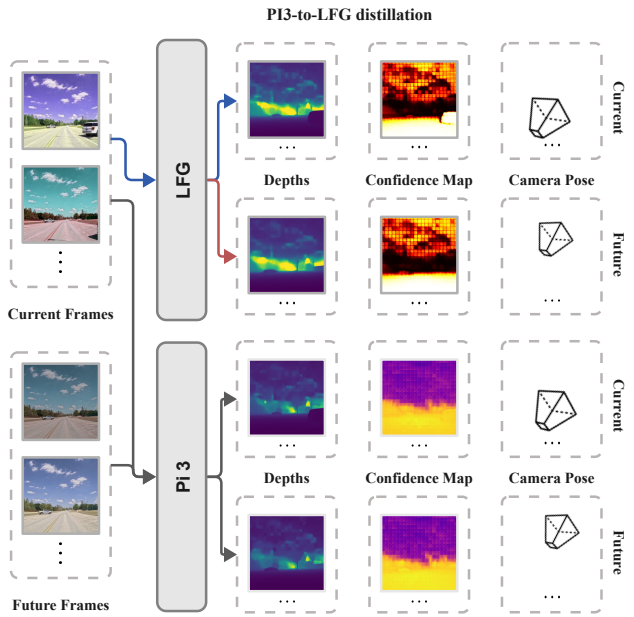

Figure 3. π^3**-to-LFG distillation.** We transfer geometric knowledge from the pretrained π^3 teacher to LFG by supervising point maps, confidence maps, and camera poses for all observed and future frames. While the teacher has access to the full sequence, the student sees only the first N frames and must predict both current and future geometry, enabling LFG to learn temporally consistent scene structure and future ego-motion from partial observations.

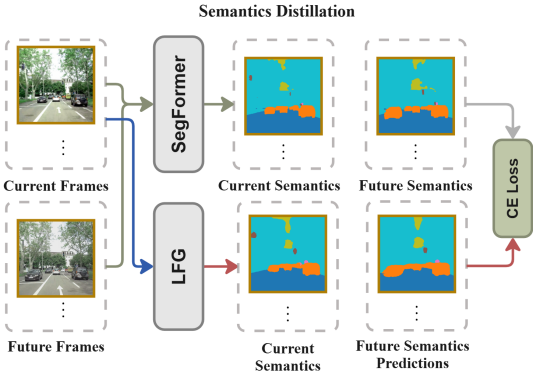

Figure 4. **Semantic distillation.** A pretrained SegFormer teacher, trained on Cityscapes, provides soft semantic pseudo-labels for each frame. LFG predicts semantic maps for both observed and future frames using only the first M inputs, learning temporally consistent scene semantics through teacher–student supervision aligned with the model's geometric features.

trajectory of the camera and enable mapping all predicted local 3D points into a shared world coordinate frame.

Third, our model predicts **semantic segmentation** with 7 classes: $(S_t)_{t=1}^{N+M}$, $S_t \in \mathbb{R}^{7\times H\times W}$, where each pixel's one-hot vector $S_t(\mathbf{y}) \in \mathbb{R}^7$ encodes the semantic category (e.g., road, vehicle, pedestrian, building, vegetation, sky, and background). These semantic predictions provide a semantic, structured understanding of the scene, which we consider to be useful for the downstream task.

We also predict **confidence maps** $(C_t)_{t=1}^{N+M}$, $C_t : \mathcal{I}(I_t) \to [0,1]$, which quantifies the reliability of each pixel's 3D prediction.

Finally, our model should predict **motion masks** $(M_t)_{t=1}^{N+M}$, $M_t : \mathcal{I}(I_t) \to [0,1]$, indicating which regions in the image correspond to independently moving objects (e.g., other vehicles, pedestrians) as opposed to the static environment. The motion masks help disentangle dynamic from static components of the scene, which can be used for downstream tasks, such as dynamic 4D Gaussian Splatting.

In total, the model predicts all outputs:

$$\mathcal{O} = \{(P_t, T_t, S_t, C_t)_{t=1}^{N+M},\ (M_t)_{t=1}^{N+M}\},$$

All modalities are learned jointly in an end-to-end fashion from video, with the assistance of robust teachers, promoting shared representations of geometry, semantics, and motion relevant to autonomous driving.

3.2. Architecture.

Our model (Fig. 2) is built on top of the π^3, [27] model, which is a purely feedforward model that predicts point maps, confidence maps, and camera poses from a series of unposed images. Contrary to prior work in VGGT [26], π^3 does not rely on a fixed referenced view, and is trained on more *dynamic* datasets, making it a suitable starting point for LFG. To receive the benefits of the pretrained π^3, we propose some simple additions on top of the model.

First, we propose to add a **causal attention autoregressive transformer** after π^3's alternating attention module or encoder. Let the output of the π^3 encoder be a sequence of latent scene tokens $\mathbf{Z}_{1:N}$, where N is the number of observed frames. The autoregressive transformer $\mathcal{T}_{\text{AR}}$ takes these tokens as input and causally predicts additional latent tokens for M future frames, producing $\mathbf{Z}_{1:N+M} = \mathcal{T}_{\text{AR}}(\mathbf{Z}_{1:N})$. Each newly generated token sequence $\mathbf{Z}_{N+1:N+M}$ represents latent scene features for unobserved frames, which are decoded into 3D point maps, confidence maps, camera poses, semantic maps, and motion masks. Our causal formulation ensures that each predicted future frame can attend to past and observed frames, but not to future frames, enforcing a forward-only information flow. The semantic and motion outputs are initialized from the point decoder and their respective heads, allowing the model to leverage shared geometric features while predicting scene semantics and dynamics.

π^3 **Teacher.** We employ a teacher π^3 model, seen in Fig. 3, that has access to $N + M$ frames from the unlabeled

OpenDV dataset [33]. The teacher outputs supervision signals in the form of point maps, confidence maps, and camera poses for **all** $N+M$ frames. Our student model only observes the first N frames and must predict both the observed (N) and future (M) outputs. Specifically, the student, LFG, predicts:

$$\{\mathbf{P}_t, \mathbf{C}_t, \mathbf{T}_t\}_{t=1}^{N+M},$$

where $\mathbf{P}_t$ denotes the point map, $\mathbf{C}_t$ the confidence map, and $\mathbf{T}_t$ the camera pose at frame t. While this method is not self-supervised as compared to other works, it forces LFG to predict future information, namely the future ego motion, confidence, and geometric updates of the scene.

3.3. Semantic Head

To enable semantic understanding of the scene, our model includes a **semantic head** (Fig. 4) that predicts dense per-pixel class probabilities for each camera and timestep. Given the input sequence of N images $(I_t)_{t=1}^{N}$, the semantic head outputs a corresponding sequence of current and future semantic maps $(S_t)_{t=1}^{N+M}$, $S_t \in [0,1]^{C_s \times H \times W}$. Since ground-truth semantic labels are unavailable for all frames, we turn to a simple *teacher–student* training strategy. A pretrained **SegFormer** model Φ_{seg}, trained on the Cityscapes dataset [2], serves as the teacher network. For each image I_t, we obtain pseudo-labels: $\hat{S}_t = \Phi_{\text{seg}}(I_t)$. These pseudo-labels act as soft supervision targets for the semantic head. The SegFormer teacher is given access to all frames, while LFG has to predict the current and future segmentation predictions.

3.4. Motion Head

Our motion head in Fig. 5 predicts per-pixel motion masks that identify dynamic regions within a scene. Since explicit motion annotations are unavailable, we generate pseudo ground-truth (pseudo-GT) labels in a fully feedforward, label-free manner.

We begin by segmenting human and vehicle instances from the first frame by using an off-the-shelf segmentation model, Grounded SAM2 [21], which produces a list of tracked mask instances per object. For each detected object, we track its 2D motion across frames using *CoTracker3* [11], which provides dense correspondences in image space: $\mathbf{u}_t^{(i)} = \text{CoTracker3}(\mathbf{I}_1, \ldots, \mathbf{I}_T, i)$, where $\mathbf{u}_t^{(i)}$ denotes the 2D tracked keypoints of object i at frame t.

Next, we employ the teacher π^3 model to obtain corresponding 3D point maps for each frame. For each object instance i, we backproject the tracked 2D points into 3D using $\mathbf{P}_t$, and measure the temporal displacement of the mean 3D position:

$$d_t^{(i)} = \left\| \bar{\mathbf{p}}_{t+1}^{(i)} - \bar{\mathbf{p}}_t^{(i)} \right\|_2,$$

where $\bar{\mathbf{p}}_t^{(i)}$ is the mean 3D position of the object at time t. An object is considered *dynamic* if its displacement exceeds a motion threshold τ_{motion} for at least $K_{\min}$ frames. Finally, we convert instance-level motion indicators $m^{(i)}$ into dense motion masks $\mathbf{M}_t \in [0,1]^{H \times W}$ that serve as supervision for the motion head.

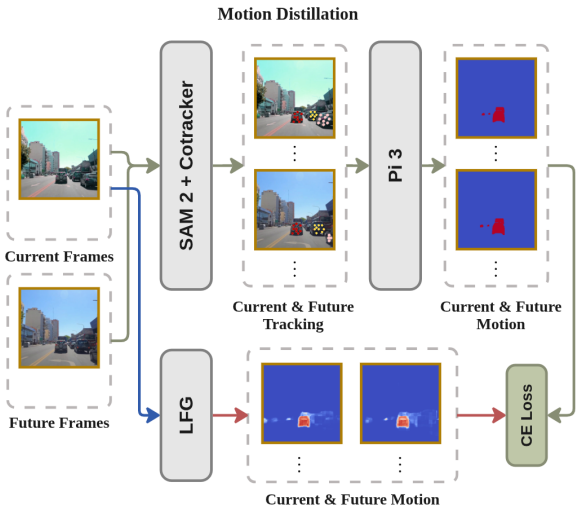

Figure 5. **Motion mask generation pipeline.** We first detect human and vehicle instances in the first frame using Grounded SAM2, then track their 2D trajectories across time with CoTracker3. Using teacher π^3 point maps, tracked pixels are back-projected into 3D and per-instance 3D displacements are measured over the sequence. Instances whose motion exceeds a threshold for at least $K_{\min}$ frames are labeled as dynamic, and their masks are rasterized into dense per-pixel motion masks $\mathbf{M}_t$, which supervise the motion head.

3.5. Losses

Our training objective combines multiple task-specific loss terms that jointly supervise segmentation, geometry, motion, and camera pose estimation. The total training loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{current}} + \lambda_{\text{future}} \mathcal{L}_{\text{future}}. \tag{1}$$

$$\begin{aligned}\mathcal{L}_{\text{current/future}} = \lambda_{\text{seg}} \mathcal{L}_{\text{seg}} + \lambda_{\text{pose}} \mathcal{L}_{\text{pose}} \\ + \lambda_{\text{point}} \mathcal{L}_{\text{point}} + \lambda_{\text{motion}} \mathcal{L}_{\text{motion}}.\end{aligned} \tag{2}$$

3.5.1. Segmentation Loss.

We use a weighted BCE loss for semantic segmentation, where we use class-specific weight to handle class imbalance. Please see the supplementary material for additional details.

3.5.2. Pose Loss.

Following the π^3 formulation, we supervise the predicted camera poses using relative pose consistency across frame pairs. For any two frames (i, j), we construct relative

transformations $(\Delta\mathbf{R}_{i\leftarrow j}, \Delta\mathbf{t}_{i\leftarrow j})$ from the student predictions and compare them against teacher-provided targets $(\widehat{\Delta\mathbf{R}}_{i\leftarrow j}, \widehat{\Delta\mathbf{t}}_{i\leftarrow j})$. The overall loss combines rotation and translation terms:

$$\mathcal{L}_{\text{pose}} = \mathcal{L}_{\text{rot}} \; + \; \lambda_{\text{trans}}\, \mathcal{L}_{\text{trans}}.$$

The rotation term penalizes geodesic distance on $\mathrm{SO}(3)$ between predicted and target relative rotations, while the translation term uses a robust regression loss (Huber) on relative translations to handle scale variation and outliers. This formulation enforces multi-frame pose consistency and stabilizes predictions over time.

3.5.3. Confidence Loss.

The confidence map estimates the reliability of each predicted 3D point. We supervise it using a binary target derived from the point-map reconstruction error: pixels whose point error falls below a threshold are treated as high-confidence, and others as low-confidence. We apply a binary cross-entropy loss to this target.

3.5.4. Point Map Loss.

We supervise the predicted 3D point maps using a scaled L_1 loss to account for varying scene scales:

$$\mathcal{L}_{\text{point}} = \alpha \, \|\mathbf{P} - \widehat{\mathbf{P}}\|_1,$$

where $\mathbf{P}$ and $\widehat{\mathbf{P}}$ denote the predicted and target point maps, respectively, and α is a learned or fixed scaling factor that normalizes for scene scale. This formulation encourages accurate 3D reconstruction while remaining robust to the absolute magnitude of the scene, analogous to the Huber-based translation loss used for relative camera motion, where we also apply a scale.

3.5.5. Motion Loss.

The motion head is trained with a binary cross-entropy loss between the model prediction (LFG) and the pseudo ground-truth (GT):

$$\mathcal{L}_{\text{motion}} = -\sum \left[M^{\text{GT}} \log M^{\text{LFG}} + (1 - M^{\text{GT}}) \log(1 - M^{\text{LFG}}) \right].$$

3.5.6. Future Frame Weighting.

To emphasize the model's ability to predict beyond observed frames, we apply a temporal weighting factor ω_t to all losses on future frames, keeping ω fixed:

$$\mathcal{L}_{\text{future}} = \sum_{t=M+1}^{N+M} \omega \, \mathcal{L}_t, \quad \text{with } \omega > 1.$$

This encourages accurate extrapolation of geometry and motion into the future time steps.

Together, these terms ensure LFG spatially and semantically understands the scene, as well as how the scene will evolve in a recent future time window. By nature, LFG exhibits generative qualities in its autoregressor; however, we assert that this is needed for next frames prediction.

3.6. Training

We train LFG in three stages. The first stage ensures that LFG can predict *future* geometry and pose autoregressively. This provides the autoregressive transformer a strong initialization to train the segmentation head, while not having to relearn future geometry and motion. Finally, we train on the motion masks, initialized from the point decoder. In each stage, LFG is trained end-to-end. We exclusively use the OpenDV Driving Youtube Daatset, and opt for a subset of it, consisting of approximately 2 million samples across varied driving conditions, scenes, traffic and external driver/pedestrian situations. We train our model on 2, 5, and 10 Hz frames (without any conditioning LFG on frequency of input frames) to improve robustness.

3.7. Fine-tuning for Planning

With a strong pretrained encoder that captures temporal and spatial scene structure from sequential images, we now demonstrate how this representation benefits downstream planning. We fine-tune on the NAVSIM planning benchmark [4] using only front-view camera inputs over three consecutive frames to predict future trajectories in complex driving scenarios.

The pretrained image encoder backbone is kept frozen and, for each frame, outputs high-dimensional **autonomy tokens** that encode the ego vehicle's motion state and surrounding context. We run LFG to produce the *future tokens* for learning. These per-frame features are aggregated and passed to a lightweight multi-modal **anchor-based trajectory decoder** that directly predicts multiple candidate trajectories in a single forward pass, similar to [17] but without any diffusion or iterative refinement. The decoder attends from autonomy features to trajectory anchors and across trajectory modes, then outputs confidence scores and coordinate offsets, selecting the highest-confidence mode as the final plan.

This simple yet effective fine-tuning strategy allows the planner to directly leverage the pretrained temporal representation for the planning task, leading to strong gains in data efficiency. In our experiments (Sec. 4.2.4), we show that this strong pretrained encoder substantially improves planning performance and data efficiency compared to state-of-the-art models that utilize multi-view or LiDAR inputs, as well as other pretrained encoders [30].

4. Experiments

4.1. Implementation and Training

4.1.1. Model and Pretraining.

We implement our model by closely following the architecture of the original π^3 backbone, which contains approximately 1 billion parameters. In total, LFG contains 1.45B parameters, and runs at 5Hz on an NVIDIA RTX 5090

GPU. The image encoder is initialized from a DINOv2-pretrained backbone, and we directly follow the π^3 alternating attention module. The point, confidence, and camera heads are frozen. The semantic and motion mask head are initialized from the point head. Our causal autoregressive transformer consists of 4 layers with 8 attention heads and a dropout rate of 0.1. It takes the latent scene tokens from the π^3 encoder and autoregressively predicts future frame tokens, which are then decoded to point maps, semantic maps, confidence maps, camera poses, and motion masks.

4.1.2. Training Setup.

We train the model using the AdamW optimizer with a base learning rate of 10^{-4}. A linear warmup schedule is used for the first 500 steps, starting from $0.1\times$ the base learning rate and increasing to the full learning rate. After warmup, we apply cosine annealing over the remaining training steps. Gradients are clipped to a maximum norm of 1.0, and mixed-precision training (BF16) is enabled. We perform gradient accumulation to increase batch size. We also randomly apply color jittering, Gaussian blur, and grayscale augmentation to the frames of the student LFG, while letting the teacher receive unaugmented images. We train on 32 A100 GPUs for 40,000 iterations. The model is trained using a combination of losses, including scaled L_1 for 3D points ($\alpha = 1.0$), Huber loss for camera translation (0.1), confidence loss (0.05), segmentation loss (1.0), and motion loss (1.0). To emphasize accurate prediction of future frames, we apply a weight of $\omega = 10.0$ to the corresponding losses. Finally, we normalize *all geometric outputs* to ensure stable learning during training.

4.2. Results

We evaluate LFG on a suite of downstream tasks that jointly probe semantics, geometry, motion, and decision making. Concretely, we consider (i) semantic segmentation, (ii) depth, point map, and camera pose prediction, and (iii) encoder-only downstream benchmarks, planning. We additionally provide qualitative motion visualizations. These tasks allow us to assess both the quality of the learned scene representation and its usefulness as a backbone for autonomous driving.

4.2.1. Semantic segmentation

Table 1. **Semantic segmentation metrics (overall vs. predicted).**

Method	Overall				Pred.			
	PA	mIoU	mDice	FW	PA	mIoU	mDice	FW
Static baseline	–	–	–	–	0.888	0.420	0.502	0.810
SegFormer	0.926	0.677	0.744	0.723	0.926	0.680	0.747	0.725
MaskFormer	0.922	0.760	0.829	0.760	–	–	–	–
LFG	0.947	0.768	0.827	0.770	0.942	0.751	0.814	0.759

We evaluate on semantic segmentation using KITTI-360 [18] with samples of 6 consecutive frames for 200 varied sequences. We compare: the segmentation teacher model SegFormer with all 6 RGB images as input, a MaskFormer baseline evaluated on overall frames (no future prediction), and our model with only the first 3 frames as input while predicting for all 6 frames. To measure the model's ability to anticipate future scene layout, we also provide the score between the ground truth semantic segmentation of the **third frame** compared to the following frames. We report standard segmentation metrics (pixel accuracy, mIoU, mDice, frequency weighted IoU) on all frames and only future frames. Table 1 shows that SegFormer is a stronger baseline than MaskFormer in this setting, and that our model not only beats its SegFormer teacher on overall semantic segmentation, but also on future frames where the teacher model was fed the RGB images and LFG was not.

Figure 6. **Segmentation quality on current and future frames.** We show results of segmentation on the 1st frame, as well as **future** frames. LFG decouples dynamic motion from its own movement.

4.2.2. Monocular depth estimation

Table 2. **Depth estimation results for overall and predicted frames.**

Dataset	Method	Overall		Predicted	
		AbsRel	RMSE	AbsRel	RMSE
KITTI-360	π^3	0.26 ± 0.08	4.37 ± 0.65	0.26 ± 0.07	4.37 ± 0.66
	LFG	0.27 ± 0.07	4.38 ± 0.64	0.31 ± 0.11	4.38 ± 0.68
	VGGT	–	4.46 ± 0.82	–	4.46 ± 0.82
	DA3	–	4.43 ± 0.81	–	4.44 ± 0.81
Waymo	π^3	0.19 ± 0.12	6.68 ± 3.10	0.19 ± 0.12	6.70 ± 3.13
	LFG	0.21 ± 0.11	6.87 ± 2.72	0.22 ± 0.11	7.12 ± 2.81

For the monocular depth prediction, we evaluate on the KITTI-360 and Waymo open dataset [24] with 200 sequences of 6 frames each. We compute root mean square error in meters after a scale and shift alignment with ground truth depth, and absolute relative depth error. Similar to semantic segmentation, we use the sample of 6 frames and give all of them to the teacher model π^3 and the first 3 to our model. We also include strong monocular baselines (VGGT and DA3) to contextualize teacher quality; these results indicate that π^3 remains the strongest teacher in our setting.

The results provided in Table 2 show that the depth prediction accuracy is on par with the teacher model (within 1 meter across the board) and only slightly worse on predicted future frames. More visualizations can be found in the supplementary. **Point cloud reconstruction.** Fig. 7 provides a qualitative comparison of full point cloud reconstructions from LFG and π^3, illustrating that LFG preserves geometric structure and camera motion even when predicting future frames.

4.2.3. Trajectory prediction

Table 3. **Trajectory estimation results.** RelPos is split into rotation (deg) and translation (m).

Dataset	Method	ATE	Rot	Trans
KITTI-360	π^3	0.43	1.32	0.31
	LFG	1.00	2.30	0.31
Waymo	π^3	0.02	0.98	0.44
	LFG	0.08	1.00	0.44

As our model predicts camera poses of input 3 frames and future 3 frames, we evaluate the trajectory prediction on KITTI-360 and Waymo open dataset (200 sequences of 6 frames each), and compare it to π^3 with all 6 frames as input. We report Absolute Trajectory Error (ATE), rotation error (Rot), and translation error (Trans). ATE measures the discrepancy between predicted and ground-truth trajectories after alignment. Rot and Trans denote the mean angular rotation error (deg) and mean translation error (m), respectively. In Table 3, we can observe that while the metrics are slightly worse that the teacher model, the result is still competitive, considering that our model does not have access to the last 3 frames.

We include a qualitative motion visualization in Fig. 8, highlighting a pseudo-ground-truth failure case where LFG correctly separates static and dynamic objects.

4.2.4. NAVSIM planning fine-tuning

PDMS summaries. We report PDMS scores for NAVSIM in the data-efficiency table (Table 4), the DiffusionDrive comparison (Table 6), and the component/scaling ablations (Table 7). Across these PDMS tables, LFG is consistently strongest at 1% and 10% labels, and remains competitive at 100%, outperforming DiffusionDrive-DINOv2 variants while benefitting from increased pretraining data and longer prediction horizons. Across these PDMS tables, LFG is consistently strongest at 1% and 10% labels, and remains competitive at 100%, outperforming DiffusionDrive-DINOv2 variants while benefitting from increased pretraining data and longer prediction horizons. **Data efficiency.** Tab. 4 evaluates how well different pretrained encoders transfer to NAVSIM planning as we vary the amount of training data. Among pretrained encoders, LFG consistently achieves the best PDMS across all label fractions: at 10% labels, LFG attains 81.4 PDMS, matching the full-data performance of DINOv3, which highlights the effectiveness of our in-the-wild video pretraining. We attribute these gains to the encoder's stronger temporal understanding of the scene, allowing it to better leverage short past frame sequences for planning. It surpasses both one of its teachers π^3 and PPGeo [30], demonstrating how both powerful feedforward architectures need semantic and temporal understanding of the future. More ablations are provided in the supplementary. **Ablations.** Table 7 shows that scaling pretraining data and extending the prediction horizon both improve PDMS at low-label regimes, while removing segmentation/motion supervision or the autoregressive head degrades performance, confirming the importance of these

Table 4. **Data-efficiency comparison (PDMS↑) on NAVSIM.** LFG's pretrained encoder yields superior data efficiency, demonstrating strong performance in the low-data regime and outperforming other pretrained encoders across all label fractions.

Method	Input	1%	10%	100% Data
DiffusionDrive	3Cam+L	64.9	72.6	88.1
DINOv3	1Cam	60.0	75.8	81.4
PPGeo	1Cam	61.5	65.6	74.6
π^3	1Cam	56.2	77.5	82.8
LFG (Ours)	1Cam	**66.3**	**81.4**	**85.2**

All pretrained encoders use a single front camera (3 frames) and the same anchor-based decoder. DiffusionDrive is trained end-to-end with a BEV-based ResNet backbone. **L** denotes LiDAR.

Table 5. **NAVSIM planning benchmark: single-camera LFG vs BEV-based baselines.** Higher is better for all metrics.

Method	Input	NC	DAC	TTC	C.	EP	PDMS
BEV Baselines							
UniAD	6Cam	97.8	91.9	92.9	100.0	78.8	83.4
TransFuser	3Cam+L	97.7	92.8	92.0	100.0	79.2	84.0
Hydra-MDP	3Cam+L	96.9	94.0	94.0	100.0	78.7	84.7
DiffusionDrive	3Cam+L	96.8	**95.4**	**94.7**	100.0	**82.0**	**88.1**
LFG (Ours)	1Cam*	**98.2**	93.7	94.4	100.0	79.1	85.2

L = LiDAR. **1Cam*** uses only the front-view camera with past temporal frames (3-frame input).

Table 6. **DiffusionDrive comparison on NAVSIM (PDMS↑).**

Method	Input	1%	10%	100%
DiffusionDrive-DINOv2	3Cam+L	57.3	74.4	81.5
DiffusionDrive-DINOv2	1Cam	57.5	73.0	79.7
LFG (Ours)	**1Cam**	**66.3**	**81.4**	**85.2**

Figure 7. Qualitative comparison of full point cloud reconstructions of LFG vs. π^3. The current camera poses are in blue, and future poses in red. LFG point maps retain overall geometric quality, even on future frames, and the predicted camera motion remains precise. **Dashed red** outlines denote predicted frames with *no ground-truth image input*, produced solely from the model's future tokens.

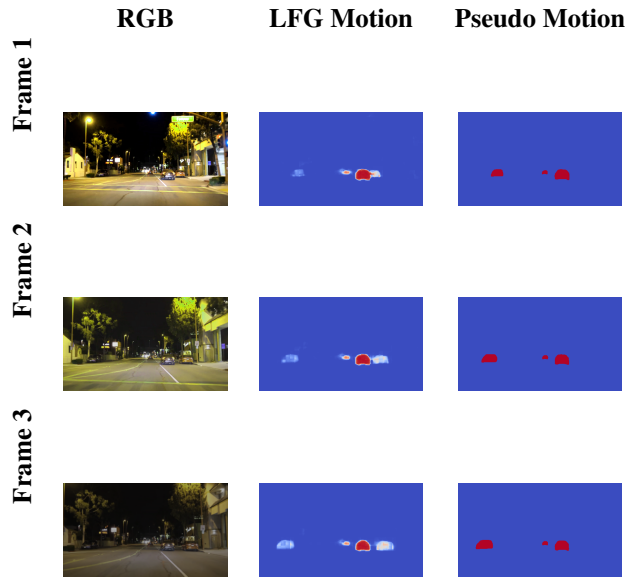

Figure 8. **Failure Case of Pseudo-GT on motion**. Qualitative comparison of motion predictions (LFG vs Pseudo) with corresponding RGB frames. In this scene, the pseudo ground truth incorrectly predicts a moving car on the far left when it is parked. LFG correctly predicts the static parked car (left) and the dynamic vehicle in front of it.

Table 7. **Component and scaling ablations on NAVSIM (PDMS↑).**

Setting	1%	10%	100%
Original setting	66.3	81.4	85.2
+ 2× pretraining data	76.6	82.3	84.8
+ Longer prediction horizon	80.5	84.4	84.8
- Seg, Motion	64.8	77.1	84.6
- Autoregressive head	66.3	77.7	84.2

components.

Benchmark results. Compared to prior methods on NAVSIM (Tab. 5), LFG, using only single front-view camera inputs, outperforms heavily engineered BEV-based baselines such as UniAD [9] and Hydra-MDP [16], which rely on multi-view cameras and/or LiDAR. LFG achieves the best Not at-fault collision (NC) score (98.2) and competitive TTC and EP scores (94.4 and 79.1), resulting in an overall PDMS of 85.2. This demonstrates that a single-camera encoder pretrained with large-scale video can rival specialized BEV-based systems that leverage significantly richer sensor suites.

5. Conclusion

In all, LFG learns directly from in-the-wild, unposed driving videos, and thanks to its strong pretrained encoder, it achieves competitive planning performance despite using only a single front-view camera. For fairness, we compare against DiffusionDrive-DINOv2 variants that use both multi-camera+LiDAR and single-camera inputs, and LFG remains stronger in this setting. For future direction, LFG predicts only short-term futures (3–6 frames), and extending the autoregressive module to longer or multi-scale temporal horizons may improve long-range reasoning. Second, we use only a single front-view camera, reflecting the fact that most in-the-wild driving videos provide only one viewpoint; while this setting already highlights the strength of video-based geometric priors, incorporating multi-view cues could further improve robustness in complex scenes. As larger multi-camera datasets such as the recently released PhysicalAI-Autonomous-Vehicles dataset [3] become available, exploring multi-view training represents a promising direction for future work.

References

[1] Andreas Blattmann et al. Stable video diffusion. *arXiv:2311.16779 [cs.CV]*, 2023. 2

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[3] NVIDIA Corporation. Physicalai-autonomous-vehicles dataset. https://huggingface.co/datasets/nvidia/PhysicalAI-Autonomous-Vehicles, 2025. Accessed: YYYY-MM-DD. 10

[4] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2, 6

[5] DeepMind. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 2

[6] Danijar Hafner et al. Mastering diverse domains through world models. *arXiv:2301.04104 [cs.AI]*, 2023. 2

[7] Jonathan Ho and Tim Salimans. Video diffusion models. *arXiv:2204.03458 [cs.CV]*, 2022. 2

[8] Yihan Hu, Jiazhi Li, Li Chen, Chonghao Sima, Xizhou Zhu, Siqi Wang, Guan_Heng Lin, Sen Zhang, X. H. Geng, Yihang Liu, Chen Jiang, Lewei Lin, Hongyang Li, Yu Qiao, and Jifeng Dai. Planning-oriented autonomous driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[9] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page —, 2023. 10

[10] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.10659*, 2024. 2

[11] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6013–6022, 2025. 5

[12] Jonathan Kulhánek et al. World models for visual navigation. In *ICLR Workshop*, 2019. 2

[13] Kaiming Li, Georgia Gkioxari, Alexander Kirillov, Ross Girshick, and Piotr Dollár. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.03220*, 2024. 2

[14] Pei Li et al. Uniad: Unified autonomous driving perception and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[15] Samuel Li, Pujith Kachana, Prajwal Chidananda, Saurabh Nair, Yasutaka Furukawa, and Matthew Brown. Rig3r: Rig-aware conditioning and discovery for 3d reconstruction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 3

[16] Zhenxin Li, Kailin Li, Kevin Ziglar, Sergio Zuniga, Jiachen Chen, and Jose M. Alvarez. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.07122*, 2024. 2, 10

[17] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12037–12047, 2025. 6, 1

[18] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022. 7

[19] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16695–16705, 2025. 3

[20] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[21] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 5

[22] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 3

[23] Oriane Siméoni et al. Dinov3: Self-supervised learning for vision at unprecedented scale. *arXiv preprint arXiv:2508.10104*, 2025. 2

[24] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 7

[25] Qijian Tian, Xin Tan, Yuan Xie, and Lizhuang Ma. Drivingforward: Feed-forward 3d gaussian splatting for driving scene reconstruction from flexible surround-view input. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7374–7382, 2025. 3

[26] 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)*, pages 5294–5306, 2025. 2, 3, 4

[27] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. Pi3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025. 4

[28] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347 [cs.CV]*, 2025. v2 revised Sept. 9 2025. 2, 3

[29] Penghao Wu, Li Chen, Hongyang Li, Xiaosong Jia, Junchi Yan, and Yu Qiao. Policy pre-training for autonomous driving via self-supervised geometric modeling. *arXiv preprint arXiv:2301.01006*, 2023. 2

[30] Penghao Wu, Li Chen, Hongyang Li, Xiaosong Jia, Junchi Yan, and Yu Qiao. Policy pre-training for autonomous driving via self-supervised geometric modeling. In *International Conference on Learning Representations (ICLR)*, 2023. 6, 8, 2

[31] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[32] Honghui Yang, Sha Zhang, Di Huang, Xiaoyang Wu, Haoyi Zhu, Tong He, Shixiang Tang, Hengshuang Zhao, Qibo Qiu, Binbin Lin, Xiaofei He, and Wanli Ouyang. Unipad: A universal pre-training paradigm for autonomous driving, 2024. 2

[33] Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, Jun Zhang, Andreas Geiger, Yu Qiao, and Hongyang Li. Generalized predictive model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. Dataset: OpenDV–YouTube, the largest real-world driving video dataset with 1,700+ hours from 244 cities in 40 countries. 5

[34] Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual point cloud forecasting enables scalable autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[35] Haiming Zhang, Wending Zhou, Yiyao Zhu, Xu Yan, Jiantao Gao, Dongfeng Bai, Yingjie Cai, Bingbing Liu, Shuguang Cui, and Zhen Li. Visionpad: A vision-centric pre-training paradigm for autonomous driving, 2025. 3

[36] Jimuyang Zhang, Ruizhao Zhu, and Eshed Ohn-Bar. Selfd: Self-learning large-scale driving policies from the web. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17316–17326, 2022. 2

[37] Qihang Zhang, Zhenghao Peng, and Bolei Zhou. Learning to drive by watching youtube videos: Action-conditioned contrastive policy pretraining. In *European Conference on Computer Vision (ECCV)*, pages 111–128. Springer, 2022. 2

Learning to Drive is a Free Gift: Large-Scale Label-Free Autonomy Pretraining from Unposed In-The-Wild Videos

Supplementary Material

This supplementary material provides additional results and implementation details. We include full training configurations in Sec. A, planning fine-tuning and baseline descriptions in Sec. B, and extended qualitative visualizations: segmentation in Sec. C, motion in Sec. D, depth in Sec. E, and point clouds in Sec. F.

A. Training Details

For reproducibility, we share specific training details of our model. We train LFG on top of the pretrained π^3, keeping the DINOv2 encoder frozen, as well as the confidence, camera, and point decoders, including automatic mixed precision (bfloat16) to speed up training.

To obtain motion masks from **Grounded SAM2** and **CoTracker3**, we first query **Grounded DINO** using the object priors *car*, *vehicle*, and *person*, which yields an initial set of candidate instance masks. Each mask is then processed with **CoTracker3**, using a grid size of 80 and a motion threshold of 0.1 in the normalized geometric space. An object is classified as dynamic if it exhibits motion in the majority of frames.

For the segmentation loss, we apply class-specific weighting across seven categories to address inherent frequency imbalances in driving scenes. Specifically, we assign weights of 0.5 to *road*, 1.2 to *vehicle*, 1.6 to *person*, 1.8 to both *traffic light* and *traffic sign*, 0.3 to *sky*, and 0.2 to *background/buildings*. These weights remain fixed throughout training and were found to provide a stable and effective balance across diverse urban environments.

We apply VGGT-style photometric augmentations during training. Color jittering perturbs brightness, contrast, and saturation by $\pm 40\%$ (0.4) and hue by $\pm 10\%$ (0.1). Random grayscale is applied with probability 0.1. Additionally, we apply random Gaussian blur with probability 0.2, using a sigma sampled uniformly from $[0.1, 2.0]$. We resize all images to $(294, 518)$, and train on the prior 3 images, predicting outputs for the next 3 images, but additionally train the motion head (final stage) on both the 3 prior images and 6 prior images. We vary the time between each image, randomly sampling from $2, 5, 10$Hz.

B. Planning Fine-tuning and Baseline Details

We fine-tune all models on the NAVSIM planning benchmark using only the front-view camera over three consecutive frames to predict 4s future ego trajectories. Unless otherwise specified, all baseline vision encoders are kept frozen and we only train lightweight causal attention adapters and a shared anchor-based trajectory decoder.

Common planning head. For all methods (ours and baselines), we employ the same anchor-based trajectory decoder. Following DiffusionDrive [17], we adopt $K = 20$ trajectory anchors obtained by K-means clustering over ground-truth futures; however we omit the diffusion component and any iterative refinement to keep the architecture simple. After causal temporal aggregation from vision encoder's embedding, the decoder attends over trajectory anchors and across modes, and in a single forward pass predicts (i) confidence scores for each of the K modes and (ii) coordinate offsets for each waypoint along each mode. At test time, the highest-confidence mode is selected as the final plan. All models predict 8 waypoints at 0.5s intervals (a 4s horizon) and are trained with a combination of focal loss (classification over modes) and L1 regression loss on waypoints.

Temporal aggregation. For each front-view frame, the pretrained encoder produces high-dimensional autonomy tokens encoding ego motion and scene context. To exploit temporal structure, we apply a small causal self-attention module across the three input frames' embeddings. The resulting aggregated features are passed into the trajectory decoder. With our method (LFG), since the encoder has already been pretrained with temporal reasoning, we use the last set of future autonomy tokens directly, which provides a temporally consistent representation for the planning head to condition on.

Baselines and training protocol. We evaluate three frozen encoders: PPGeo (geometric pre-training), DINOv3 (self-supervised ViT), and Pi3 (4D self-supervised learning). Each is followed by the same causal temporal adapter and shared anchor-based planning head. Our method (LFG) uses the pretrained temporal autoregressive encoder described in the main paper (also kept frozen), along with a lightweight multi-modal trajectory decoder. All models are optimized with AdamW and a cosine learning-rate schedule, and are trained under identical data-scaling regimes using 1%, 10%, and 100% of NAVSIM training data with learning rate 1e-4 to study data efficiency.

For DiffusionDrive, we follow the publicly released implementation (code available on GitHub) which uses three

Table A1. **Comparing PPGeo with different pretraining data-source.** PPGeo* indicates that the model is pretrained on the *same OpenDV dataset* used by LFG.

Method	Input	1%	10%	100% Data
PPGeo	1Cam	61.5	65.6	74.6
PPGeo*	1Cam	59.8	70.0	76.4
LFG (Ours)	1Cam	**66.3**	**81.4**	**85.2**

front-view cameras plus LiDAR input and their corresponding hyper parameters.

For PPGeo, in Tab. 5 we use the publicly released ResNet-34 encoder from the PPGeo repository[1] pretrained with geometric self-supervision [30]. The original PPGeo encoder is pretrained using the YouTube driving video dataset introduced in the ACO project[2]. To isolate the impact of pre-training data source, we evaluate a variant, PPGeo*, where we replicate the same geometric pre-training procedure but restrict the pre-training corpus to exactly the data used by LFG. As shown in Tab. A1, PPGeo* slightly improves performance at higher label fractions but still under-performs LFG by a wide margin, highlighting that LFG's 4-D temporal pre-training paradigm provides inductive biases that align more directly with downstream planning.

NAVSIM metrics The NAVSIM benchmark uses a composite score called the Predictive Driver Model Score (PDMS) to evaluate planning performance. PDMS is computed in two phases: (i) two hard-multiplier subscores **No at-fault Collisions (NC)** and **Drivable Area Compliance (DAC)** that immediately zero the scenario score if violated; (ii) a weighted average of three performance subscores **Ego Progress (EP)**, **Time-to-Collision (TTC)**, and **Comfort (C)** — reflecting route progress, safety margin, and motion smoothness.

Formally,:

$$\text{PDMS} = \big(\text{NC} \times \text{DAC}\big) \times \frac{5\,\text{EP} + 5\,\text{TTC} + 2\,C}{5 + 5 + 2}$$

Here:

- NC = 1 if no at-fault collision, = 0.5 if a collision with a static object, = 0 otherwise.
- DAC = 1 if the ego vehicle remains within the drivable area for the entire rollout, = 0 if it leaves.
- EP is the ratio of actual route progress achieved to a safe upper bound (clipped to [0,1]).
- TTC = 1 if the minimum time-to-collision along the 4s horizon exceeds a fixed threshold, else = 0.
- C = 1 if all vehicle kinematic thresholds (acceleration, jerk) remain within comfort bounds, else = 0.

All metrics are evaluated via a non-reactive 4-second rollout in the benchmark simulator in the test set 12k samples.

C. Segmentation Visualizations

We show segmentation visualizations on the OpenDV dataset, with sample unposed images, and the teacher SegFormer model outputs as in Fig. A5, on a 5hz scene. We find that LFG performs very competitively with its SegFormer teacher on the current frames, and future predicts the motion of the moving bus as it is about to pass the ego vehicle. LFG, however, suffers from a smoothing effect in the later frames. We posit that training LFG on more steps and the entire OpenDV dataset will improve this, as well as an edge aware point map loss to improve crispness of future frame predictions.

D. Motion Visualizations

We demonstrate motion visualization on the OpenDV dataset, seen in Fig A2, with current frames to emphasize the performance trained from pseudo ground truth data, on 10Hz, but we show 3 frames spaced apart every other frame. LFG correctly predicts the moving cars in frame from only 2D images, with a small amount of frames. Future work entails demonstrating LFG's performance for constructing dynamic Gaussian Splats, where the motion masks can be freely obtained.

E. Depth Visualizations

We show depth visualizations of LFG compared to π^3 on validation images on our dataset, at a frequency of 5Hz, on Fig. A3. LFG performs comparable to π^3 on the seen frames, and while sharp edges are slowly lost in the future frames, LFG is able to understand dynamic and static objects, and the relative positioning of the other vehicles over time. Future work will crispen the point maps, and more results, including on motion and semantic results, are shown at the end of the supplementary.

F. Full Point Visualizations

[1] https://github.com/OpenDriveLab/PPGeo
[2] https://github.com/metadriverse/ACO

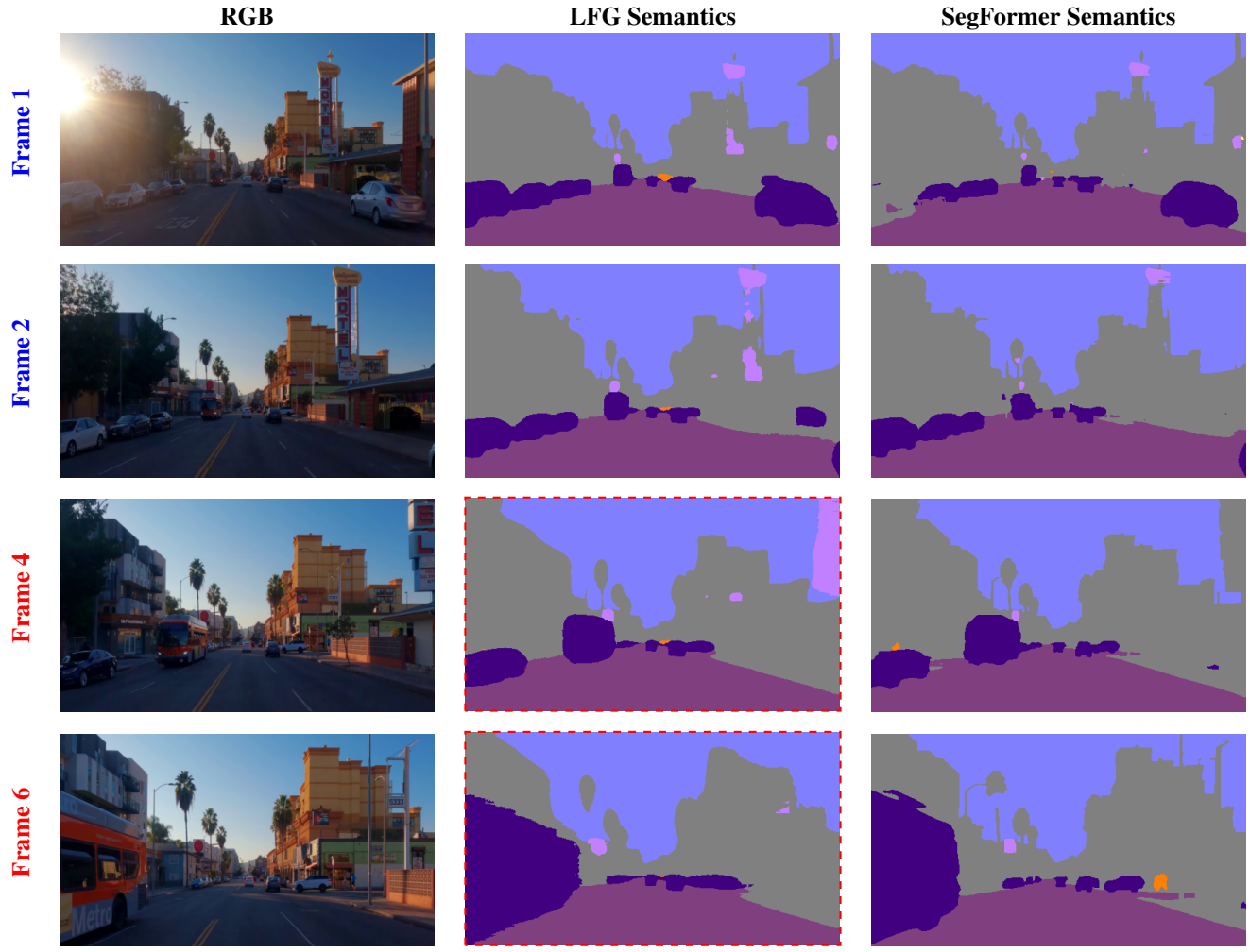

Figure A1. Qualitative comparison of semantic segmentation across RGB, LFG, and SegFormer for current frames 1 and 2 (with ground-truth input) and future frames 4 and 6. Dashed red outlines denote predicted frames with *no ground-truth image input*, produced solely from the model's future tokens.

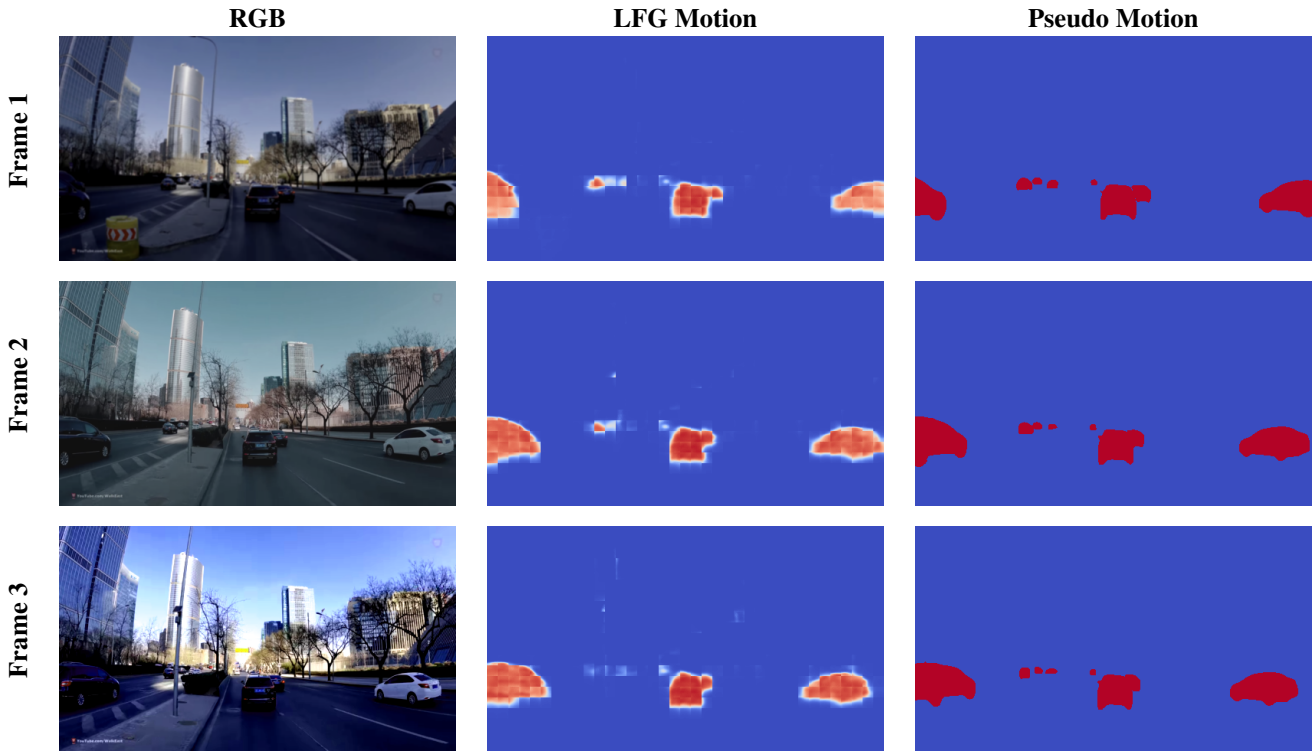

Figure A2. Qualitative comparison of motion predictions (LFG vs Pseudo GT motion) with corresponding RGB frames. We show results on non-future frames to demonstrate the motion map precision on a few images. The ego vehicle is moving on a road with three nearby vehicles moving.

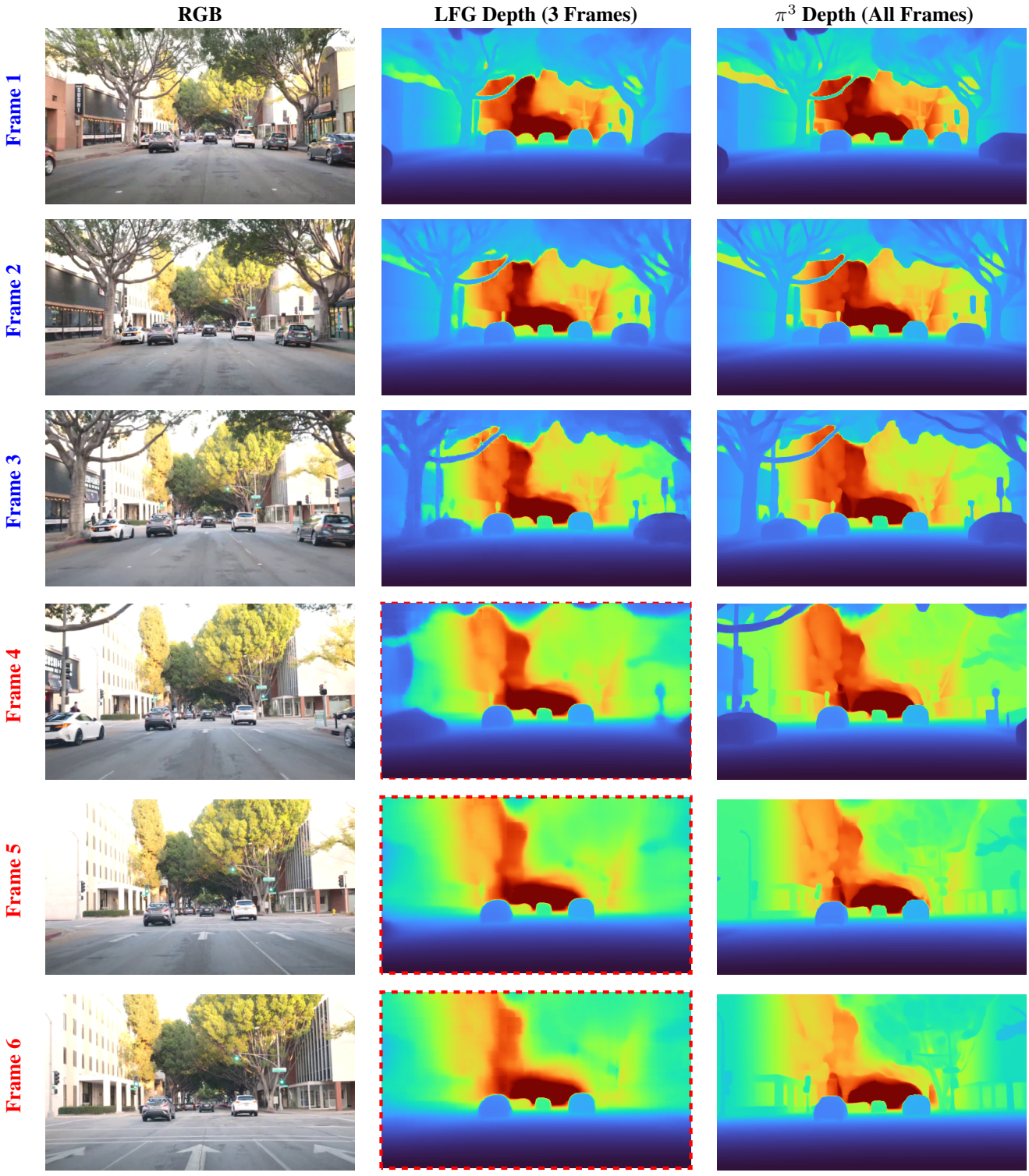

Figure A3. Qualitative comparison of depth prediction for six frames (first three frames are blue, the future three frames are red. LFG is able to decouple static and dynamic objects as it continues along the road, and future work will improve the sharpness of the last frames' predictions. **Dashed red** outlines denote predicted frames with *no ground-truth image input*, produced solely from the model's future tokens.

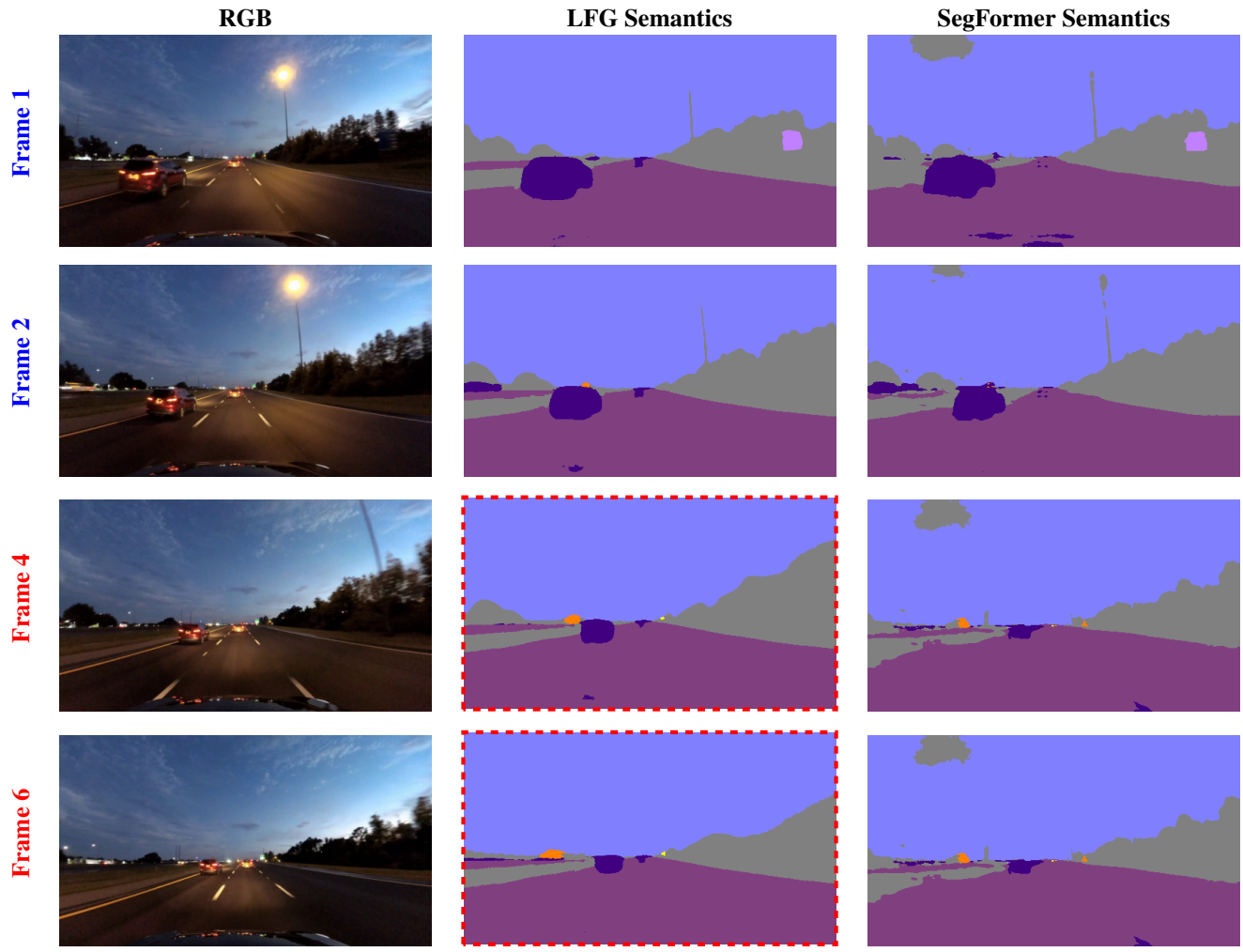

Figure A4. Additional qualitative comparison of semantic segmentation across RGB, LFG, and SegFormer for current frames 1 and 2 (with ground-truth input) and future frames 4 and 6. **Dashed red** outlines denote predicted frames with *no ground-truth image input*, produced solely from the model's future tokens. LFG on current frames enjoys crisper predictions even than its teacher.

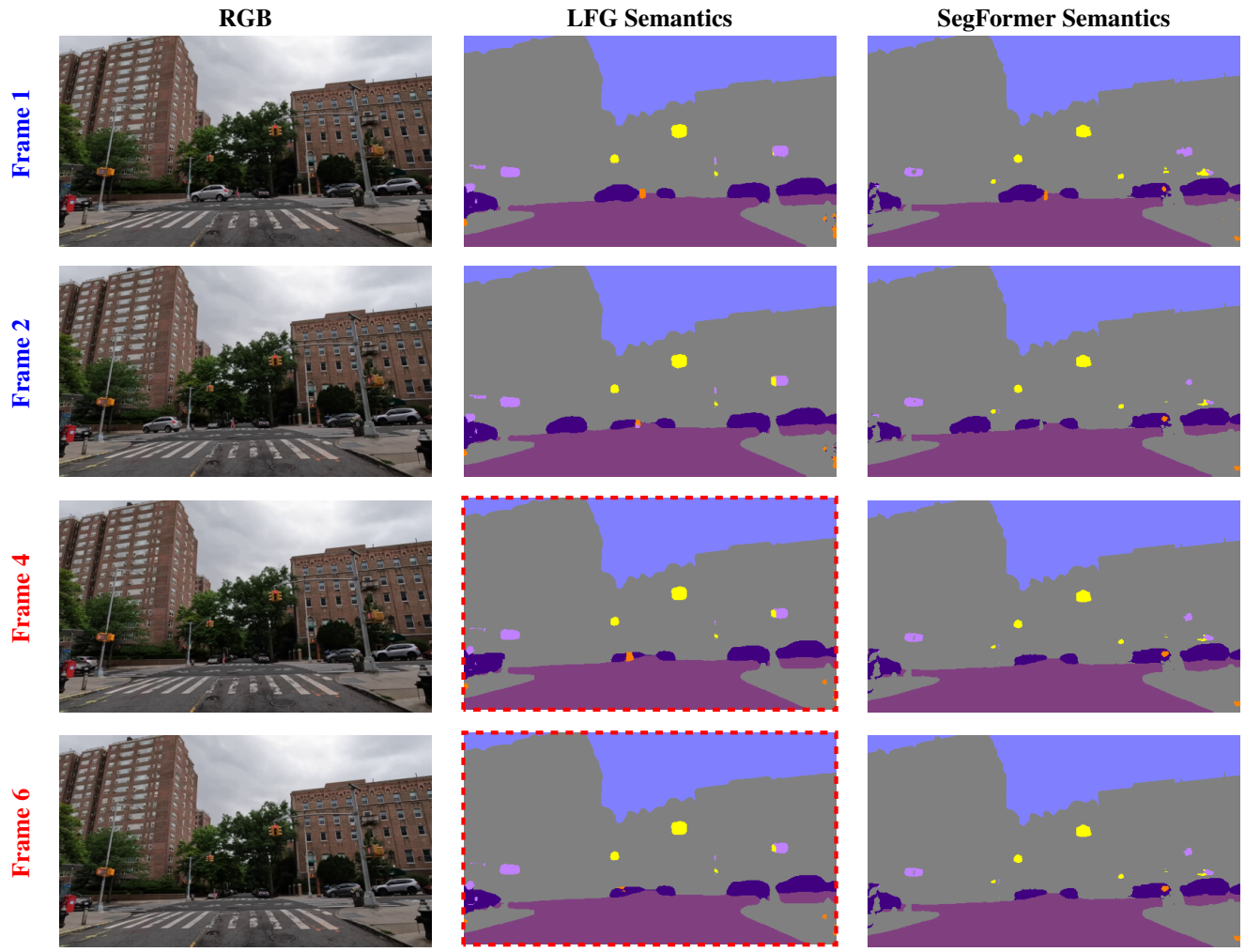

Figure A5. Additional qualitative comparison of semantic segmentation across RGB, LFG, and SegFormer for current frames 1 and 2 (with ground-truth input) and future frames 4 and 6. **Dashed red** outlines denote predicted frames with *no ground-truth image input*, produced solely from the model's future tokens. LFG retains accurate predictions of cars, road, buildings, sky, traffic lights and signs, and even a person.

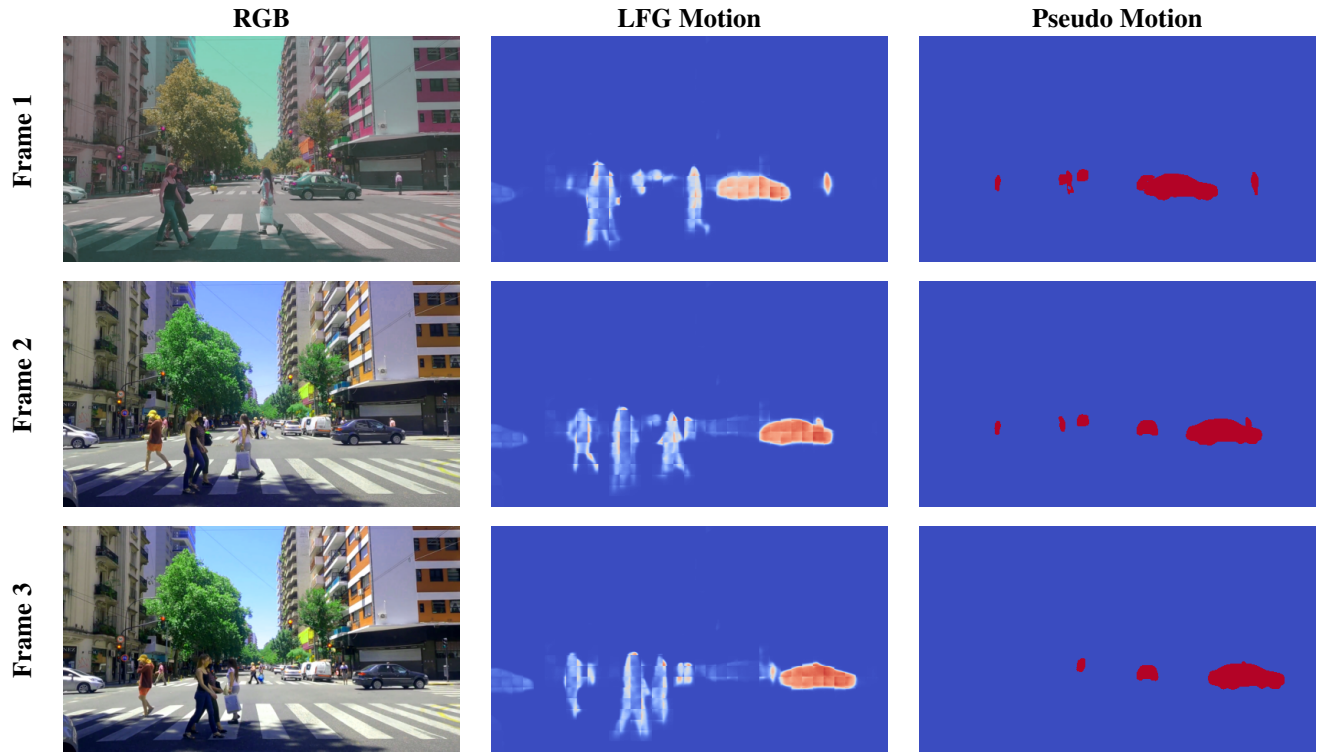

Figure A6. More qualitative comparison of motion predictions (LFG vs Pseudo) with corresponding RGB frames. In this scene, LFG predicts the moving car across the intersection, but also the close pedestrians, demonstrating that the pretrained point decoders of π^3 improve the predictions.

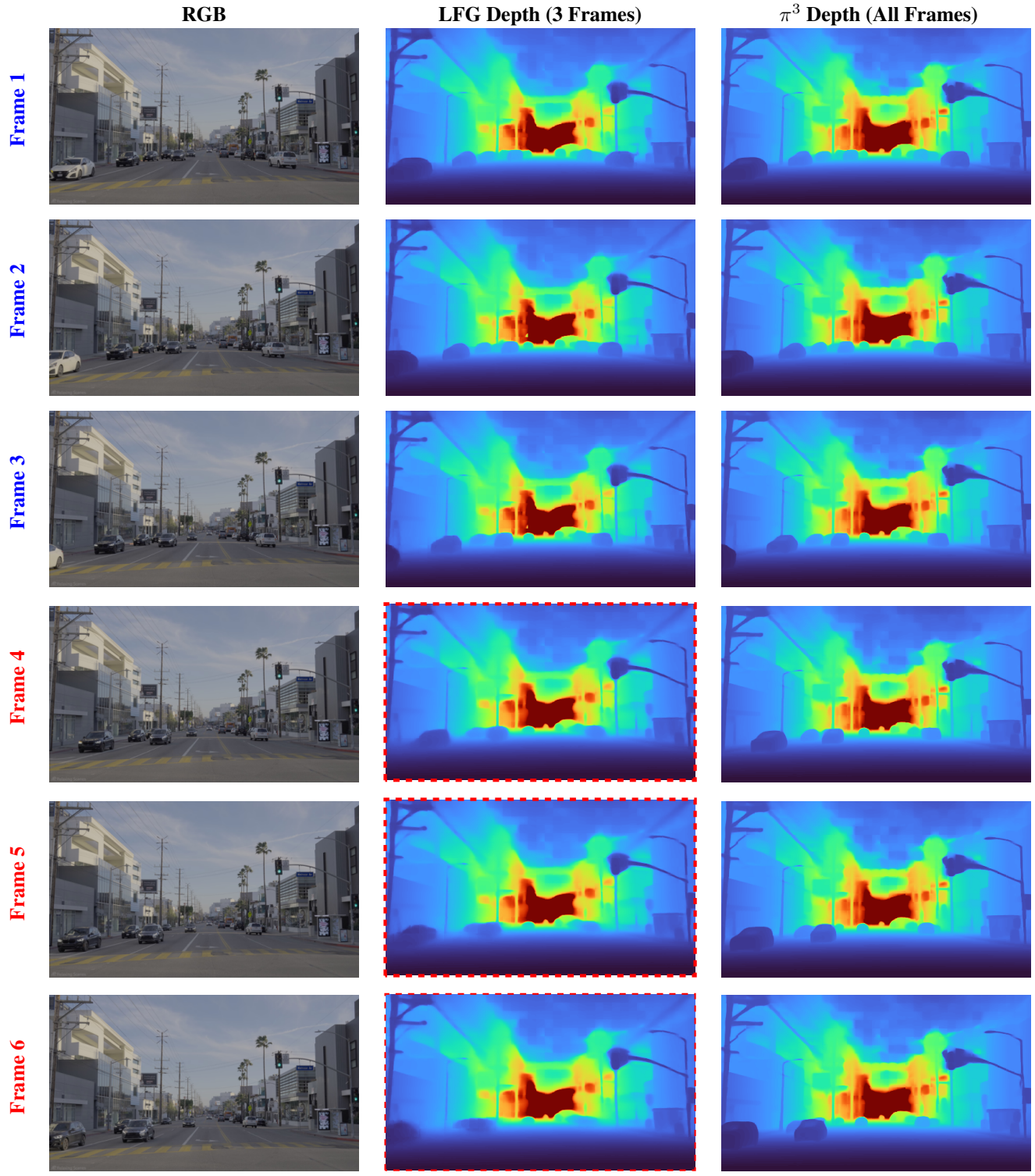

Figure A7. Qualitative comparison of depth prediction for six frames. LFG is able to decouple static and dynamic objects as it continues along the road, and future work will improve the sharpness of the last frames' predictions. **Dashed red** outlines denote predicted frames with *no ground-truth image input*, produced solely from the model's future tokens

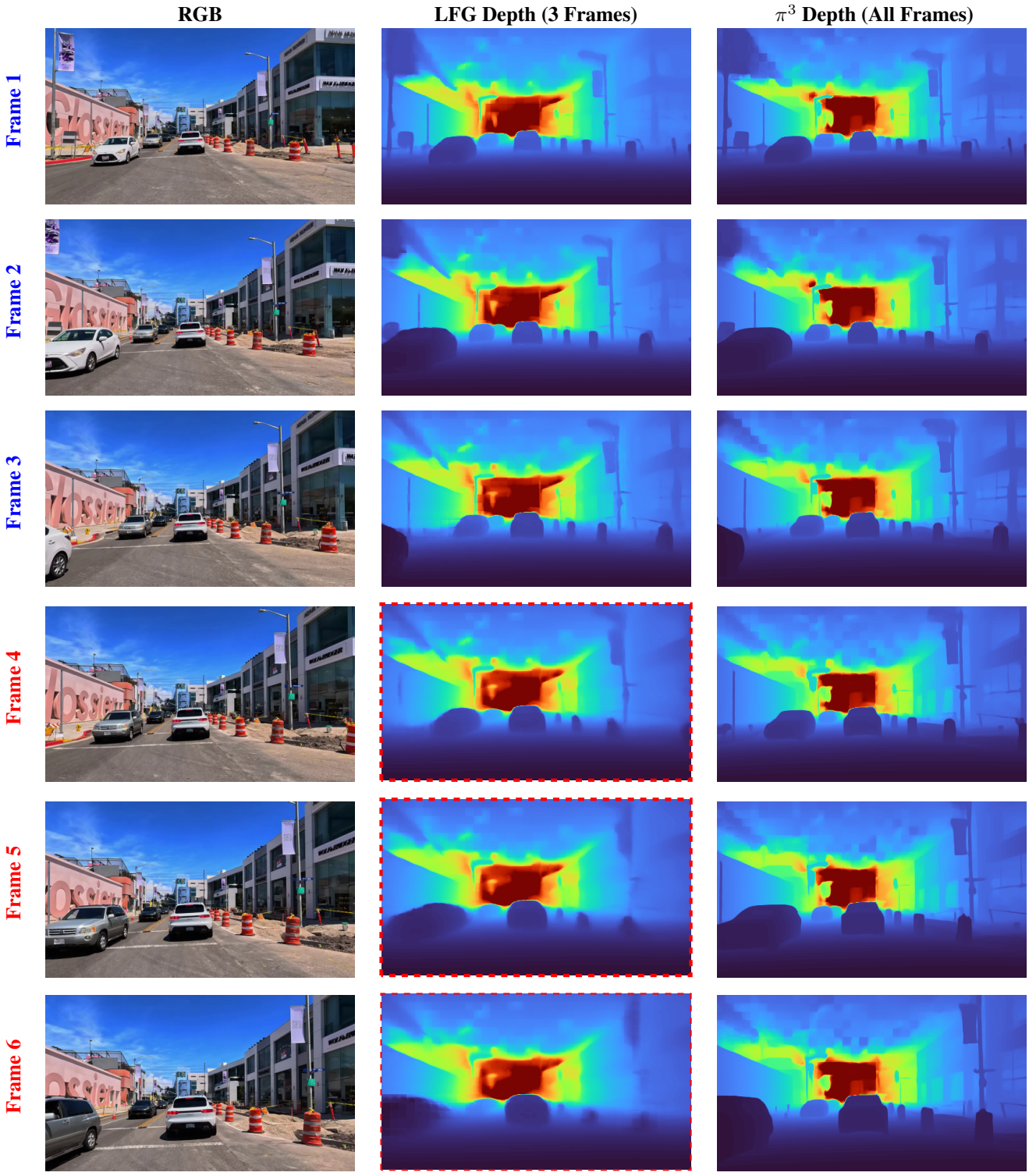

Figure A8. More qualitative comparison of depth prediction for six frames. **Dashed red** outlines denote predicted frames with *no ground-truth image input*, produced solely from the model's future tokens