DINO-Foresight: Looking into the Future with DINO

Efstathios Karypidis[1,3] **Ioannis Kakogeorgiou**[1] **Spyros Gidaris**[2] **Nikos Komodakis**[1,4,5]

[1]Archimedes, Athena Research Center, Greece [2]valeo.ai
[3]National Technical University of Athens [4]University of Crete [5]IACM-Forth

Abstract

Predicting future dynamics is crucial for applications like autonomous driving and robotics, where understanding the environment is key. Existing pixel-level methods are computationally expensive and often focus on irrelevant details. To address these challenges, we introduce DINO-Foresight, a novel framework that operates in the semantic feature space of pretrained Vision Foundation Models (VFMs). Our approach trains a masked feature transformer in a self-supervised manner to predict the evolution of VFM features over time. By forecasting these features, we can apply off-the-shelf, task-specific heads for various scene understanding tasks. In this framework, VFM features are treated as a latent space, to which different heads attach to perform specific tasks for future-frame analysis. Extensive experiments show the very strong performance, robustness and scalability of our framework. Project page and code at https://dino-foresight.github.io/

1 Introduction

Predicting future states in video sequences is a key challenge in computer vision and machine learning, with important applications in autonomous systems like self-driving cars and robotics (Finn et al., 2016; Dosovitskiy and Koltun, 2017). These systems must navigate dynamic environments safely, yet predicting future states remains difficult—especially in complex scenarios involving multi-object interactions over long time horizons.

Recent approaches focus on generating realistic RGB future frames using latent generative modeling (Rombach et al., 2022). These methods first compress RGB data into a latent space, such as continuous (Rombach et al., 2022) or discrete (Esser et al., 2021) Variational Autoencoder (VAE) representations. Then, they train generative models—like diffusion models (Zheng et al., 2024; Gao et al., 2024; Ho et al., 2022a,b; Harvey et al., 2022; Brooks et al., 2024), autoregressive models (Hu et al., 2023; Hong et al., 2023; Kondratyuk et al., 2024; Wang et al., 2024), or masked video generation (Yu et al., 2023, 2024)—to predict future states in this compressed space. While this reduces dimensionality and improves training stability (Rombach et al., 2022), VAE latents often lack semantic alignment, making them hard to interpret or use in downstream scene understanding tasks (see Table 2). Moreover, these methods must model both low-level appearance and high-level semantics, even though decision-making systems (e.g., self-driving cars) primarily need semantic scene understanding—what objects exist and where they are. Latent generative approaches may waste capacity on irrelevant low-level details, compromising temporal semantic accuracy.

In contrast, vision foundation models (VFMs) have revolutionized scene understanding with robust, transferable features (Oquab et al., 2024; Radford et al., 2021; Venkataramanan et al., 2025; Bardes et al., 2024). This raises a key question: *can VFM features serve as a semantically rich, high-dimensional latent space for precise future prediction?*

In this work, we propose precisely this idea. Instead of predicting future low-level VAE latents, we forecast the temporal evolution of VFM features directly. This shift brings several important advantages: **(a) Beyond low-level latents.** It leverages semantically meaningful representations,

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-5466

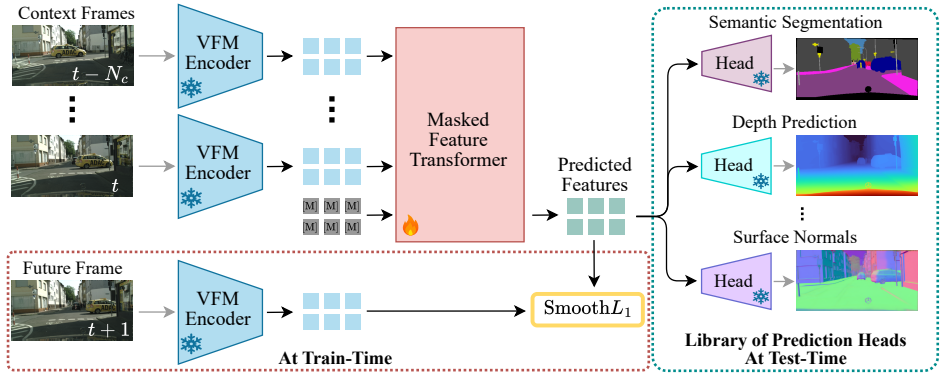


Figure 1: **Forecasting VFM Features for Future Frames.** At the core of our approach is the prediction of future VFM feature evolution. To this end, we train a masked transformer model in a self-supervised manner to forecast these features from context frames, minimizing SmoothL1 loss between predicted and actual future features. By forecasting these rich and versatile features, task-specific prediction heads—such as semantic segmentation, depth, and surface normals—can be effortlessly employed at test time, enabling modular and efficient multi-task scene understanding.

inheriting strong scene understanding. **(b) Dynamic semantics over raw frames.** It avoids full-frame synthesis, letting the model focus on meaningful dynamics, reducing complexity and sidestepping challenges like multimodal pixel distributions. **(c) Scalable multi-task support and modular integration.** Forecasted features align with downstream tasks, enabling plug-and-play integration with pretrained classifiers, segmenters or task-specific heads—without the need to retrain the core feature predictor (see Figure 1). This represents a significant departure from prior semantic feature prediction methods (Luc et al., 2017), which face significant challenges with multi-task scalability. Such approaches either require training a separate model for each task (Nabavi et al., 2018; Chiu et al., 2020; Terwilliger et al., 2019) or involve predicting features from multiple task-specific models simultaneously (Hu et al., 2020; Karypidis et al., 2025), resulting in more complex and less scalable architectures.

In this work we show that forecasting high-dimensional VFM features is not only feasible but also achieves strong performance, offering a new path toward semantically grounded future prediction.

In summary, our contributions are: **(1)** We introduce `DINO-Foresight`, a self-supervised method for semantic future prediction that builds on the key idea of forecasting VFM features—our core contribution. Unlike latent generative methods that rely on low-level VAE features, our approach delivers precise future scene understanding without modeling unnecessary appearance details, enabling a unified model for diverse scene understanding tasks. **(2)** To realize this idea, we design an efficient masked feature transformer (see Figure 1) that propagates multi-layer, high-resolution features critical for achieving strong task performance. **(3)** Experimental results demonstrate a unique advantage of our approach - our single model successfully handles multiple future-frame understanding tasks (semantic segmentation, instance segmentation, depth prediction, and surface normal prediction) where previous approaches required multiple specialized models. Furthermore, as we show in Appendix Subsection A.2 intermediate features within our masked transformer can further improve downstream task performance, highlighting its potential as a self-supervised learning strategy that enhances the already strong VFM features.

2 Related Work

Video Prediction is an extensively studied problem. Early approaches based Convolutional LSTMs (Nabavi et al., 2018; Xu et al., 2018; Wang et al., 2018; Castrejon et al., 2019; Lee et al., 2021; Wu et al., 2021; Gao et al., 2022) struggled to maintain both visual quality and temporal consistency. Subsequent developments introduced generative adversarial networks and variational autoencoders (Yan et al., 2021; Vondrick et al., 2016b; Castrejon et al., 2019; Lee et al., 2018; Babaeizadeh et al., 2018), alongside diffusion models (Ho et al., 2022a,b; Gao et al., 2024; Harvey et al., 2022), to enhance spatial-temporal coherence and improve the quality of predictions. Furthermore, transformer-

https://doi.org/10.52202/085713-5466

based models have also been adapted to videos, utilizing auto-regressive and masked modeling objectives to capture video dynamics (Yu et al., 2024, 2023; Gupta et al., 2023; Wang et al., 2024).

Latent Generative Models Current state-of-the-art video prediction methods (Gupta et al., 2023; Yu et al., 2023, 2024; Gao et al., 2024) build upon latent generative approaches (Rombach et al., 2022). These frameworks employ an autoencoder to compress RGB frames into a latent space, then train generative models to predict future states in this compressed representation. While sharing superficial similarities with our approach—including the use of latent spaces and masked transformers in some cases (Gupta et al., 2023; Yu et al., 2023)—these methods differ fundamentally. Their latent spaces primarily encode low-level visual information, requiring reconstruction back to RGB space and forcing the model to simultaneously handle both appearance details and semantic changes.

Our key innovation lies in forecasting VFM features that natively encode high-level semantic information, enabling direct application of task-specific prediction heads. As demonstrated in Table 2, conventional VAE latent spaces cannot match this capability. Our experimental results in Table 1 further show these methods' limitations in predicting semantic scene evolution, highlighting the distinct advantages of VFM feature forecasting for scene understanding tasks. Concurrently to our work, DINO-WM (Zhou et al., 2025), also leverages DINOv2 features for world modeling with action-conditioned planning in simulated environments, while our work targets multi-task dense semantic forecasting in real world scenarios.

Semantic Future Prediction An emerging approach for future-frame prediction focuses on forecasting intermediate features from an encoder rather than raw RGB values (Nabavi et al., 2018; Lin et al., 2021; Saric et al., 2020; Sun et al., 2019; Chen and Han, 2019; Jin et al., 2017; Luc et al., 2018; Šarić et al., 2019; Hu et al., 2021; Vondrick et al., 2016a; Zhong et al., 2023). This strategy models abstract encoder representations, which task-specific heads use for downstream tasks. Early methods in this paradigm include F2F (Luc et al., 2018), which regresses Mask-RCNN's feature pyramid, and F2MF (Saric et al., 2020), which improves feature prediction using flow-based warping. APANet (Hu et al., 2021) aggregates multi-level features via an auto-path mechanism for instance segmentation, while PFA (Lin et al., 2021) enhances global structures and suppresses local details for more predictable features. Recently, Futurist (Karypidis et al., 2025) introduced a multi-modal semantic forecasting approach for semantic segmentation and depth maps. While effective, these methods often rely on task- or dataset-specific encoders, limiting practicality and scalability. To address this, we use VFM encoders, which, due to large-scale pre-training, perform well across diverse tasks and generalize effectively to new scenes without retraining.

Vision Foundation Models (VFMs) VFMs have transformed computer vision, achieving strong performance across a range of visual tasks. Trained on large-scale datasets, these models learn rich, transferable visual representations. Notable examples include DINOv2 (Caron et al., 2021; Oquab et al., 2024), a self-supervised model based on self-distillation; Franca (Venkataramanan et al., 2025), a fully-open VFM for scalable self-supervised representation learning; CLIP and its variants (Radford et al., 2021; Fang et al., 2023; Sun et al., 2023; Fang et al., 2024), which align visual representations with natural language; SAM (Segment Anything Model) (Kirillov et al., 2023), a foundation model for image segmentation and V-JEPA(Bardes et al., 2024), a video representation learning method via feature prediction in latent space. In this work, we explore the potential of VFM features for semantic future prediction tasks, connecting static visual understanding with dynamic prediction.

Multi-Task Learning Multi-Task Learning (MTL) is a learning paradigm that enables simultaneous training of models on multiple related tasks (Maninis et al., 2019; Misra et al., 2016; Neseem et al., 2023; Vandenhende et al., 2022), promoting shared representations and improving performance across tasks. Traditional MTL frameworks often use parameter sharing (Kendall et al., 2018; Sener and Koltun, 2018; Bekoulis et al., 2018) or task interaction allowing exchange of information (Bragman et al., 2019; Chen et al., 2023a,b; Misra et al., 2016; Ruder et al., 2019). Other approaches employ a strategy that incrementally increases the model's depth during training, enabling the network to learn task-specific representations in a more resource-efficient way Aich et al. (2023); Choi and Im (2023); Guo et al. (2020); Lu et al. (2017); Zhang et al. (2022). Recently, the emergence of large-scale pretrained models has led to the introduction of adapter-based multi-task fine-tuning approaches (Liang et al., 2022; Liu et al., 2022). In our work, we leverage VFM features to provide a unified, scalable and modular framework for future prediction. Our approach enables seamless integration of multiple tasks without retraining or complex adaptations.

 https://doi.org/10.52202/085713-5466

3 Methodology

Our semantic future prediction approach builds on forecasting VFM features – powerful representations that excel at scene understanding tasks and generalize well to unseen environments. We realize this through a masked transformer model trained via self-supervision to predict the temporal evolution of VFM features. This forecasted feature space serves as a latent representation that can flexibly connect to various off-the-shelf task-specific heads for future-frame analysis.

Figure 1 provides an overview of our approach, with the key components detailed in the following sections: **Section 3.1** describes how target features are generated from multi-layer VFM features. **Section 3.2** presents our formulation of VFM feature forecasting as a masked feature modeling problem and details the model architecture. **Section 3.3** covers efficient training techniques for high-resolution VFM feature prediction. Finally, **Section 3.4** introduces our modular framework for multi-task future-frame analysis and details how prediction heads are trained and integrated.

3.1 Hierarchical Target Feature Construction

In Figure 2, we provide an overview of how the target feature space for the feature prediction model is constructed. Below, we outline the main steps involved.

Multi-Layer VFM Feature Extraction Scene understanding models often benefit from processing features across multiple layers of an image encoder (Ranftl et al., 2021; Long et al., 2015; Lin et al., 2017; Cheng et al., 2022; Zhao et al., 2017; Chen et al., 2017), especially when the encoder is frozen, as in our approach. To fully leverage the pretrained VFMs, we propagate features extracted from multiple layers of the VFM. We ablate the impact of multi-layer features in Appendix Subsection A.4.

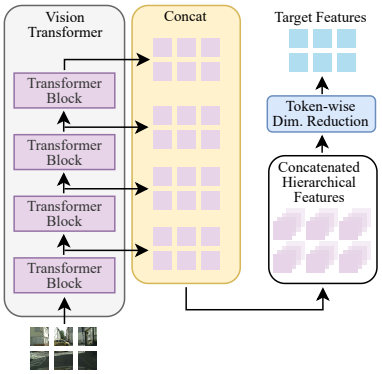


Figure 2: **Hierarchical Target Feature Construction for the Feature Prediction Model.** Our framework constructs a feature space by extracting and concatenating multi-layer features from a frozen ViT encoder, capturing semantic information at varying abstraction levels. PCA is applied to reduce dimensionality, creating compact features.

Our framework uses VFMs based on the Vision Transformer (ViT) architecture, though the approach can be extended to other architectures. Given a sequence of N image frames $\mathbf{X} \in \mathbb{R}^{N\times H'\times W'\times 3}$, let $\mathbf{F}^{(l)} \in \mathbb{R}^{N\times H\times W\times D_{enc}}$ represent the features extracted from layer l of the ViT model. Here, D_{enc} is the feature dimension, and $H \times W$ is the spatial resolution, which are consistent across layers in ViT-based models. To form the target feature space on which the feature prediction model operates, we concatenate the features from L layers along the channel dimension, resulting in $\mathbf{F}_{\text{concat}} \in \mathbb{R}^{N\times H\times W\times L\cdot D_{enc}}$. These concatenated features capture rich semantic information from the input images at multiple levels of abstraction.

Dimensionality Reduction The concatenated features $\mathbf{F}_{\text{concat}}$ have high dimensionality, so we apply dimensionality reduction to simplify the feature prediction task while retaining essential information. In this work, we use Principal Component Analysis (PCA) to reduce the dimensionality, transforming $\mathbf{F}_{\text{concat}}$ into a lower-dimensional representation $\mathbf{F}_{\text{PCA}} \in \mathbb{R}^{N\times H\times W\times D}$, where $D \ll L \cdot D_{enc}$. These PCA-reduced features, $\mathbf{F}_{\text{PCA}}$, serve as the target features on which the future feature prediction operates, i.e., $\mathbf{F}_{\text{TRG}} = \mathbf{F}_{\text{PCA}}$. In Appendix Subsection A.1, we show that this PCA-based compression retains essential information and enhances downstream performance.

3.2 Self-Supervised Future Feature Prediction with Masked Transformers

Masked Feature Transformer Architecture Inspired by previous video generation models (Chang et al., 2022; Gupta et al., 2023; Karypidis et al., 2025), we implement the future feature prediction task using a self-supervised masked transformer architecture. The task involves predicting future frames in a video sequence consisting of N frames, where N_c are context frames and N_p are future

https://doi.org/10.52202/085713-5466

frames to be predicted, such that $N = N_c + N_p$. Given the target features $\mathbf{F}_{\text{TRG}} \in \mathbb{R}^{N \times H \times W \times D}$ for these N frames, the future-frame tokens are masked, and the transformer must predict these missing tokens by processing all the tokens from the entire sequence, i.e., all the $N \cdot H \cdot W$ tokens.

The architecture begins with a token embedding stage, where each token is projected from D dimensions into the transformer's hidden dimension D_{dec} through a linear layer. During training, the tokens corresponding to the future frames are replaced with a learnable D_{dec}-dimensional [MASK] vector. During inference, these [MASK] tokens are appended after the context frames. Each token also receives a position embedding to retain both temporal and spatial information across the sequence.

The tokens are then passed through a series of transformer layers. Standard self-attention layers in transformers have quadratic time complexity with respect to the number of tokens, making them computationally expensive for high-resolution, multi-frame sequences. To address this, we follow the approach from recent video transformers (Arnab et al., 2021; Gupta et al., 2023; Karypidis et al., 2025) and decompose the attention mechanism into temporal and spatial components. Temporal attention is applied across tokens with the same spatial position in different frames, capturing the dynamics over time. Spatial attention, on the other hand, operates within individual frames, focusing on spatial interactions. Thus, each transformer layer consists of a temporal Multi-Head Self-Attention (MSA) layer, a spatial MSA layer, and a feedforward MLP layer. After passing through the transformer layers, a linear prediction layer maps the output token embeddings from the hidden dimension D_{dec} back to the feature dimension D, producing the predicted feature map $\tilde{\mathbf{F}}_{\text{TRG}} \in \mathbb{R}^{N \times H \times W \times D}$.

Training Objective We frame the future-frame prediction as a continuous regression problem and optimize a self-supervised training objective based on the SmoothL1 loss between the predicted features $\tilde{\mathbf{F}}_{\text{TRG}}$ and the ground truth features $\mathbf{F}_{\text{TRG}}$ at the masked locations. The loss is defined as:

$$\mathcal{L}_{\text{MFM}} = \mathop{\mathbb{E}}_{x \in \mathcal{X}} \left[\sum_{p \in \mathcal{P}} L\left(\mathbf{F}_{\text{TRG}}(p), \tilde{\mathbf{F}}_{\text{TRG}}(p) \right) \right], \tag{1}$$

where $\mathcal{X}$ denotes the training dataset, $\mathcal{P} = [N_c + 1 : N] \times [1 : H] \times [1 : W]$ represents the set of positions across the $H \times W$ spatial dimensions of the N_p future frames, and $\mathbf{F}_{\text{TRG}}(p)$ and $\tilde{\mathbf{F}}_{\text{TRG}}(p)$ are the ground truth and predicted feature vectors at position p, respectively. $L(\cdot, \cdot)$ computes the SmoothL1 loss between two feature vectors:

$$L(x, y) = \sum_{d=1}^{D} \begin{cases} 0.5 \frac{(x_d - y_d)^2}{\beta}, & \text{if } |x_d - y_d| < \beta, \\ |x_d - y_d| - 0.5\beta, & \text{otherwise.} \end{cases} \tag{2}$$

In our experiments, we set $\beta = 0.1$.

3.3 Compute-Efficient Training Strategies for High-Resolution Feature Forecasting

Using high-resolution features is crucial for pixel-wise scene understanding tasks, such as segmentation or depth prediction, where low-resolution features struggle to capture small objects or fine spatial structures (Ranftl et al., 2021; Cheng et al., 2022; Chen et al., 2017; Touvron et al., 2019). To achieve good performance on these tasks, we aim to forecast VFM features extracted from frames with a spatial resolution of $H' \times W' = 448 \times 896$. For a ViT with a patch size of 14×14, as used in DINOv2 (Oquab et al., 2024) and EVA-CLIP (Fang et al., 2024), this results in feature maps with a resolution of $H \times W = 32 \times 64$, corresponding to 2048 tokens per frame.

However, training ViTs on such high-resolution inputs is computationally expensive in terms of both time and memory (Dosovitskiy et al., 2020). To address this challenge while maintaining efficient training, we explore the following strategies:

Low-Resolution Training with High-Resolution Inference In this approach, we train on frames with a lower resolution of 224×448, resulting in features with a resolution of 16×32. During testing, we use high-resolution frames (448×896) and adapt the position embeddings through interpolation. However, this strategy leads to suboptimal performance due to a distribution shift between the training and test data, which causes inaccurate feature forecasting.

Sliding-Window Approach for High-Resolution Inference Inspired by sliding-window techniques used in segmentation tasks (Strudel et al., 2021), this strategy trains the model with cropped feature

 https://doi.org/10.52202/085713-5466

maps. The ViT encoder extracts features from high-resolution frames (448×896), producing high-resolution tokens (e.g., 32×64 for a patch size of 14×14). During training, we sample local crops of size 16×32, taken from the same spatial locations across frames. The model is trained on these cropped features. At inference time, the model processes the high-resolution features in a sliding-window manner using the same crop dimensions. This approach efficiently handles large inputs while avoiding the computational cost of full-resolution training.

Two-Phase Training with Resolution Adaptation This strategy employs a two-phase training process (Oquab et al., 2024; Touvron et al., 2019; Kolesnikov et al., 2020). First, the model is trained on low-resolution frames (224×448) for several epochs, focusing on learning broad feature forecasting. Then, the model is fine-tuned on high-resolution (448×896) for a small number of epochs. This adaptation phase improves the model's ability to handle high-resolution features without incurring the computational cost of training from scratch at the higher resolution. As shown in our experiments, both strategies are effective, but the two-phase approach yields better feature forecasting performance. This is likely because the masked transformer has access to a larger spatial context when propagating VFM features in future frames.

3.4 Modular Framework for Future-Frame Multi-Task Predictions

Our framework supports a library of interchangeable task-specific prediction heads that operate on the preficted future freames, enabling flexible multi-task future understanding. The design is modular: each head operates independently, allowing tasks to be added or removed without retraining the core feature prediction model.

We focus on four pixel-wise prediction tasks: semantic segmentation, instance segmentation, depth prediction, and surface normals prediction. However, our framework is general and can support other scene understanding tasks, such as object detection and panoptic segmentation. For semantic segmentation, depth prediction, and surface normals prediction, we use the Dense Prediction Transformer (Ranftl et al., 2021) (DPT) architecture, which is well-suited to our setup. DPT leverages multi-layer features from ViT-based encoders—consistent with the features predicted by our model—and progressively refines them into high-resolution predictions using convolutional fusion. For the instance segmentation task, we use a Mask2Former (Cheng et al., 2022) head.

Prediction heads can be trained directly on frozen VFM features and then applied "off-the-shelf" to future-frame features predicted by the model. Additionally, they can be trained to account for the PCA stage by applying PCA compression and decompression to the multi-layer features. This approach is useful for cases where prediction heads are trained on annotated 2D images, without requiring video data, and then added to the library for future-frame predictions.

4 Experiments

4.1 Experimental setup

Data. We assess our approach using the Cityscapes (Cordts et al., 2016) and nuScenes (Caesar et al., 2020) datasets, both offering video sequences from urban driving environments. The Cityscapes dataset includes 2,975 training sequences, 500 for validation, each with 30 frames captured at 16 fps and a resolution of 1024 × 2048 pixels. The 20th frame in each sequence is annotated for semantic segmentation with 19 classes. The nuScenes dataset comprises of 700 training scenes and 150 validation scenes, captured at a frame rate of 12 Hz, with each sequence extending for 20 seconds.

Implementation details. By default, we use DINOv2-Reg with ViT-B/14 as the default VFM visual encoder for our method. For the masked feature transformer we built upon (Besnier and Chen, 2023) implementation. We use 12 layers with a hidden dimension of $d = 1152$ and sequence length $N = 5$ (with $N_c = 4$ context frames and $N_p = 1$ future frame). For end-to-end training, we use the Adam optimizer (Kingma and Ba, 2015) with momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.99$, and a learning rate of 6.4×10^{-4} with cosine annealing. Training is conducted on 8 A100 40Gb GPUs with an effective batch size of 64. We train DPT (Ranftl et al., 2021) heads for the semantic segmentation, depth prediction, and surface normals estimation tasks, and a Mask2Former (Cheng et al., 2022) head for the instance segmentation task. More implementation details in Appendix Subsection D.

Table 1: **Comparison with state-of-the-art on multiple forecasting tasks.** Methods that do not handle a task are marked with '-'. For approaches requiring separate prediction models per task (e.g., PFA), we show multiple entries (semantic/instance). ALL: mIoU of all classes. MO: mIoU of movable objects. VISTA$_{ft}$ is the VISTA model fine-tuned on Cityscapes. Compared approaches include: 3Dconv-F2F-RGB (Chiu et al., 2020), Dil10-S2S (Luc et al., 2017), F2F (Luc et al., 2018), ConvLSTM (Chiu et al., 2020), FeatReproj3D (Vora et al., 2018), Bayesian S2S (Bhattacharyya et al., 2019), 3Dconv-F2F-SEG (Chiu et al., 2020), DeformF2F (Šarić et al., 2019), LSTM AM S2S (Chen and Han, 2019), APANet (Hu et al., 2021), LSTM M2M (Terwilliger et al., 2019), IndRNN-Stack (Graber et al., 2021), DiffAttn-Fuse (Graber et al., 2022), F2MF (Saric et al., 2020), and PFA (Lin et al., 2021), Futurist (Karypidis et al., 2025) and VISTA (Gao et al., 2024).

Method	Semantic Segmentation				Instance Segmentation				Depth		Surface Normals	
	Short		Mid		Short		Mid		Short	Mid	Short	Mid
	ALL	MO	ALL	MO	AP50	AP	AP50	AP	δ_1	δ_1	11.25°	11.25°
3Dconv-F2F-RGB	57.0	-	40.8	-	-	-	-	-	-	-	-	-
Dil10-S2S	59.4	55.3	47.8	40.8	-	-	-	-	-	-	-	-
F2F	-	61.2	-	41.2	39.9	19.4	19.4	7.7	-	-	-	-
ConvLSTM	60.1	-	-	-	-	-	-	-	-	-	-	-
FeatReproj3D	61.5	-	45.4	-	-	-	-	-	-	-	-	-
Bayesian S2S	65.1	-	51.2	-	-	-	-	-	-	-	-	-
3Dconv-F2F-SEG	65.5	-	50.5	-	-	-	-	-	-	-	-	-
DeformF2F	65.5	63.8	53.6	49.9	-	-	-	-	-	-	-	-
LSTM AM S2S	65.8	-	51.3	-	-	-	-	-	-	-	-	-
CPConvLSTM	-	-	-	-	44.3	22.1	25.6	11.2	-	-	-	-
APANet	-	64.9	-	51.4	46.1	23.2	29.2	12.9	-	-	-	-
LSTM M2M	67.1	65.1	51.5	46.3	-	-	-	-	-	-	-	-
IndRNN-Stack	67.6	60.8	58.1	52.1	-	-	-	-	-	-	-	-
DiffAttn-Fuse	67.9	61.2	58.1	51.7	-	-	-	-	-	-	-	-
F2MF	69.6	67.7	57.9	54.6	-	-	-	-	-	-	-	-
PFA (semantic)	71.1	69.2	60.3	56.7	-	-	-	-	-	-	-	-
PFA (instance)	-	-	-	-	48.7	24.9	**30.5**	**14.8**	-	-	-	-
Futurist	**73.9**	**74.9**	**62.7**	**61.2**	-	-	-	-	**96.0**	**91.9**	-	-
Oracle	77.0	77.4	77.0	77.4	66.2	40.4	66.2	40.4	89.1	89.1	95.3	95.3
VISTA$_{ft}$	64.9	62.1	53.9	51.0	33.1	17.7	19.8	9.0	86.4	82.8	93.0	90.0
DINO-Foresight (ours)	71.8	71.7	59.8	57.6	**50.5**	**26.6**	27.3	12.6	88.6	85.4	**94.4**	**91.3**

Evaluation Metrics. To evaluate our method's performance we use the following metrics: For **semantic segmentation**, we use mean Intersection over Union (mIoU) in two ways: (1) mIoU (ALL), which includes all semantic classes, and (2) MO-mIoU (MO), which considers only movable object classes like person, rider, car, truck, bus, train, motorcycle, and bicycle. For **instance segmentation**, we measure performance using average precision at a 0.50 IoU threshold (AP50) and also the mean average precision over IoU thresholds from 0.50 to 0.95. For **depth prediction**, we use mean Absolute Relative Error (AbsRel) and depth accuracy (δ_1). For **surface normals**, we compute the mean angular error (m↓) and the percentage of pixels with angular errors below 11.25° (11.25°↑). Definitions of depth and surface normals metrics are provided in Appendix Subsection E.

Evaluation Scenarios Following prior work (Luc et al., 2017; Nabavi et al., 2018), we evaluate our model on Cityscapes for **short-term prediction** (3 frames, 0.18s) and **mid-term prediction** (9 frames, 0.54s). On nuScenes, which has more static scenes (i.e., with less movement) than Cityscapes, we use **mid-term prediction** (9 frames, 0.75s) and **long-term prediction** (18 frames, 1.5s). Further details in Appendix Subsection F.

Baselines We evaluate our method against three baselines. The first, the Oracle baseline, directly accesses the target future frame, establishing an upper performance bound. The second, Copy-Last, copies the most recent context frame to predict the target frame, providing a lower performance bound. Both baselines use DINOv2-Reg ViT-B encoder with DPT heads for semantic segmentation, depth prediction, and surface normals estimation, and a Mask2Former head for instance segmentation. The third baseline leverages VISTA (Gao et al., 2024), a state-of-the-art world model that uses video latent diffusion to generate future RGB frames from three context frames. This is a large-scale model, comprising 2.5 billion parameters and trained on 1,740 hours of driving videos, which we fine-tune on Cityscapes and nuScenes using the same frame-rate as our model. VISTA generates future RGB frames (with action-free conditioning), which are processed by a DINOv2-Reg encoder with DPT heads (identical to other baselines) for semantic segmentation, depth and surface normals prediction.

https://doi.org/10.52202/085713-5466

Table 2: **Comparison of VFM encoders across tasks**. For each encoder (DINOv2, EVA2-CLIP, SAM), we show performance on segmentation (ALL, MO), depth estimation (δ_1 accuracy, AbsRel error), and surface normal prediction (m, percentage within 11.25°).

Encoder	Method	Segmentation				Depth				Surface Normals			
		Short		Mid		Short		Mid		Short		Mid	
		ALL↑	MO↑	ALL↑	MO↑	δ_1 ↑	AbsR↓	δ_1↑	AbsR↓	m↓	11.25°↑	m↓	11.25°↑
DINOv2	Oracle	77.0	77.4	77.0	77.4	89.1	.108	89.1	.108	3.24	95.3	3.24	95.3
	Copy Last	54.7	52.0	40.4	32.3	84.1	.154	77.8	.212	4.41	89.2	5.39	84.0
Oquab et al. (2024)	Prediction	**71.8**	**71.7**	**59.8**	**57.6**	**88.6**	**.114**	**85.4**	**.136**	**3.39**	**94.4**	**4.00**	**91.3**
EVA2-CLIP	Oracle	71.0	69.5	71.0	69.5	85.2	.123	85.2	.123	3.37	94.5	3.37	94.5
	Copy Last	51.9	47.7	38.5	29.5	81.2	.161	75.6	.216	4.52	88.5	5.44	83.6
Fang et al. (2024)	Prediction	66.3	64.2	54.5	49.6	85.1	.122	82.5	.145	3.56	93.4	4.18	90.1
SAM	Oracle	69.8	63.9	69.8	63.9	84.8	.143	84.8	.143	3.01	96.0	3.01	96.0
	Copy Last	49.4	41.8	36.8	26.0	78.3	.211	73.4	.267	4.84	87.4	5.77	82.4
Kirillov et al. (2023)	Prediction	65.3	59.3	52.5	43.9	81.3	.178	77.6	.209	3.80	92.8	4.49	89.2
	Oracle	47.3	34.7	47.4	35.2	61.5	.251	61.5	.251	5.3	86.1	5.3	86.1
VAE	Copy Last	37.1	25.6	28.5	16.5	60.7	.252	59.2	.286	5.8	83.2	6.3	80.1
Rombach et al. (2022)	Prediction	33.4	17.9	24.7	9.8	64.1	.281	61.4	.394	6.5	80.5	8.0	73.2

4.2 VFM feature forecasting results

Comparison with State-of-the-Art Table 1 compares our method with state-of-the-art approaches for semantic/instance segmentation, depth estimation, and surface normal forecasting. The results highlight our key advantage: a single feature prediction model that achieves competitive or superior performance across multiple scene understanding tasks. In contrast, prior works either require separate prediction models per task (Luc et al., 2018) or handle at most two tasks simultaneously (Karypidis et al., 2025). This demonstrates the flexibility and practicality of our VFM feature forecasting approach.

Unified Representations for Multiple Tasks: VFM Features vs. RGB Pixels An alternative to forecasting VFM features is predicting future frames directly in RGB space, which also supports performing multiple downstream tasks through standard scene understanding models. For comparison, we fine-tune `VISTA` (Gao et al., 2024) to generate five future frames from three context frames (8 frames in total). We process these synthesized frames (both short-term and mid-term) with DINOv2-Reg ViT-B and DPT heads (similar to our method's setup) for fair evaluation. Despite `VISTA`'s large-scale training and model size (2.5B parameters), it achieves lower performance on semantic segmentation, depth estimation, and surface normal prediction. Extended evaluations on nuScenes (provided Appendix Table 7) show similar trends for depth and surface normal estimation. Our approach is also far more computationally efficient: mid-term forecasting on Cityscapes' 500 validation scenes takes approximately 5 minutes versus `VISTA`'s 8.3 hours (both on a single A100 GPU). This highlights our method's advantages of operating in VFM feature space – achieving accurate semantic prediction while being significantly more resource-efficient.

Comparison of VFM Visual Encoders In Table 2, we evaluate our method using three VFM encoders to extract features for our feature prediction model: DINOv2 with registers (Darcet et al., 2024; Oquab et al., 2024) (self-supervised), EVA2-CLIP (Fang et al., 2024) (vision-language contrastive), and SAM (Kirillov et al., 2023) (supervised instance segmentation). For each, we use the ViT-B variant. We also include `Copy-Last` and `Oracle` baselines for comparison. The results show that: **(1)** DINOv2 consistently outperforms other encoders across all tasks, achieving the best results for both short- and mid-term predictions. **(2)** This aligns with expectations, as the DINOv2-based `Oracle` also performs best in all cases. **(3)** Our model effectively predicts future-frame features for all VFMs, significantly improving over the `Copy-Last` baseline. Based on these findings, we select DINOv2 as our default VFM encoder.

Future Prediction: VFM features vs VAE-based latents Additionally, in Table 2, we evaluate using VAE latents (Rombach et al., 2022) (used in latent generative models) instead of VFM features, training DPT prediction heads on these latents. Results are significantly worse, as expected, since these latents lack high-level information, and even DPT oracles perform poorly. This highlights the advantage of forecasting VFM features over VAE latents.

Table 3: **Continuous vs. Discrete VFM Representations**. Comparison of continuous DINOv2 features (our approach) against 4M's DINOv2 tokenizer with discrete codes. Unlike other tables and for fair comparison with the 4M tokenized variant, we use the DINOv2 ViT-B w/o Reg model and extract features from the last layer only. Results on semantic segmentation forecasting.

Method	Continuous				Discrete (4M tokenized)			
	Short		Mid		Short		Mid	
	ALL↑	MO↑	ALL↑	MO↑	ALL↑	MO↑	ALL↑	MO↑
Oracle	72.9	74.0	72.9	74.0	70.2	71.4	70.2	71.4
Copy Last	54.7	51.9	40.5	32.2	53.7	51.0	40.0	31.6
Prediction	68.9	69.3	57.3	55.0	61.7	60.9	53.7	51.0

Table 4: **Strategies for Training-Efficient High-Resolution Feature Forecasting**.

Resolutions (Train→Test)	Adaptation Approach	Short-term ALL	Short-term MO	Mid-term ALL	Mid-term MO
Oracle					
(a) 224→224	N/A	68.24	66.41	68.24	66.41
(b) 448→448	N/A	76.97	77.40	76.97	77.40
Forecasting					
(c) 224→224	N/A	64.50	62.63	55.49	52.62
(d) 224→448	Pos. interp.	64.34	64.29	48.31	44.60
(e) 224→448_{224}	Sliding win.	71.26	71.11	58.75	56.78
(f) (224&448)→448	Two-phase	**71.81**	**71.71**	**59.78**	**57.65**

Discrete vs. Continuous VFM Representations Recent work in generative modeling has explored both discrete and continuous representations for image and video generation (Chang et al., 2022; Yu et al., 2023; Yan et al., 2021; Razavi et al., 2019; Li et al., 2024; Tschannen et al., 2024). Recent findings favor continuous representations (Li et al., 2024; Tschannen et al., 2024), showing that removing vector quantization can improve generation quality while retaining the benefits of sequence modeling. To further investigate this in the context of VFM feature forecasting, we employ 4M's pretrained DINOv2 tokenizer (Bachmann et al., 2024), which encodes DINOv2 features (without register tokens) into discrete codes from a vocabulary of size 8192. We train a quantized variant of DINO-Foresight (ViT-B14, single-layer features) using a cross-entropy loss to predict these discrete codes, which are decoded back to DINOv2 features at inference time. Results are reported in Table 3. While the discretized variant achieves comparable oracle performance to the continuous VFM feature case, our continuous VFM feature forecasting approach yields superior future semantic prediction results. These findings suggest that preserving the rich, continuous representations from VFMs—without quantization—offers clear advantages for dense semantic forecasting tasks.

Training-Efficient Strategies for High-Resolution Feature Forecasting In Table 4, we compare the resolution-adaptation strategies from Section 3.3, reporting results for future semantic segmentation. *High-resolution features are essential*: comparing the low-resolution Oracle baseline (model (a)) with the high-resolution Oracle baseline (model (b)) highlights the importance of high-resolution features for strong segmentation performance. Consequently, forecasting low-resolution features (model (c)) results in significantly poorer segmentation than models predicting high-resolution features (models (e) and (f)). Adapting a model trained on low-resolution features for high-resolution inputs by simply adjusting position embeddings during inference (model (d)) leads to suboptimal results, even underperforming compared to low-resolution forecasting. The other two adaptation strategies—Sliding Window (model (e)) and two-phase training with resolution increase (model (f))—achieve considerably better results, demonstrating their effectiveness. The two-phase approach is simpler and yields the best performance, so we adopt it as our default for high-resolution feature forecasting.

Additional Ablations and Analysis We conduct comprehensive ablation studies in the appendix to further validate our approach. First, in Appendix Subsection A.3, we demonstrate strong zero-shot generalization by training on Cityscapes and evaluating on nuScenes without fine-tuning. The resulting performance is only slightly worse than models trained directly on nuScenes, while surpassing

https://doi.org/10.52202/085713-5466

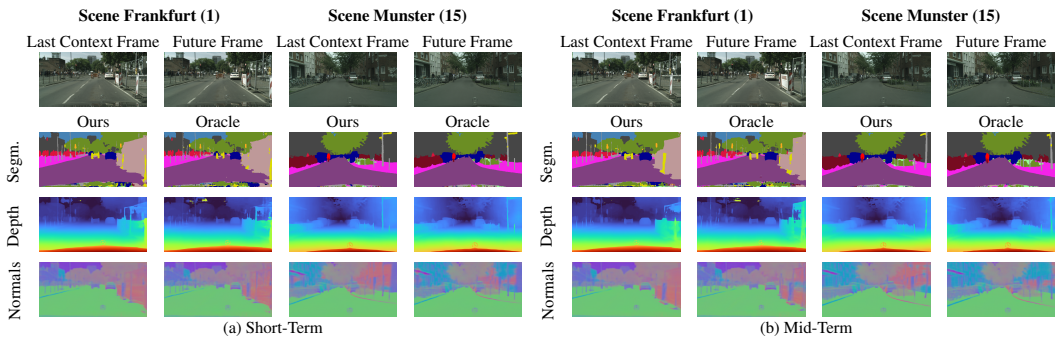


Figure 3: **Future predictions for semantic segmentation, depth, and surface normals.** Noisy segmentations at the bottom of the image (in both predicted and Oracle results) are due to unannotated regions in Cityscapes that are ignored during DPT training. This artifact affects only segmentation, not the predicted features, as evident in the clear depth and surface normal predictions.

all baselines. Second, in Appendix Subsection A.5, we investigate the impact of different masking strategies (random vs. full masking) and loss functions (L1, MSE, SmoothL1, and SmoothL1 with cosine similarity). Third, in Appendix Subsection A.6, we analyze scalability across model sizes (Small: 115M, Base: 258M, Large: 460M parameters) and data scale (Cityscapes alone vs. Cityscapes+nuScenes combined), demonstrating consistent performance improvements with increased capacity and data diversity. These results highlight the promising robustness and scalability of our VFM feature forecasting approach.

4.3 Qualitative results.

In Figure 3, we present qualitative results from our method applied to semantic segmentation, depth estimation, and surface normal prediction tasks, with both short-term and mid-term future predictions. Our single VFM feature prediction model produces meaningful outputs across all tasks, demonstrating the benefits of leveraging the feature space of large-scale pre-trained VFMs for future prediction.

5 Conclusion

In this work, we introduced `DINO-Foresight`, a self-supervised framework for semantic future prediction that shifts the paradigm from forecasting low-level latent representations to predicting high-dimensional, semantically rich VFM features. This shift offers several key advantages: it enhances scene understanding by leveraging structured semantic information, avoids the complexity of full-frame synthesis, and enables scalable, modular integration with downstream tasks.

To realize this approach, we designed a masked feature transformer that efficiently propagates high-resolution, multi-layer VFM features over time. Our experiments show that forecasting such features is not only feasible but also highly effective—demonstrating strong performance across diverse future-frame understanding tasks including semantic segmentation, instance segmentation, depth estimation, and surface normal prediction. Unlike prior methods that rely on multiple task-specific models, our single model handles all tasks seamlessly, validating the scalability and versatility of our framework.

Overall, this work lays the foundation for a new class of unified and modular future prediction systems, grounded in semantic reasoning rather than low-level reconstruction.

Acknowledgements This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program. Hardware resources were granted with the support of GRNET. Also, this work was performed using HPC resources from GENCI-IDRIS (Grants 2023-A0141014182, 2023-AD011012884R2, and 2024-AD011012884R3).

References

Abhishek Aich, Samuel Schulter, Amit K. Roy-Chowdhury, Manmohan Chandraker, and Yumin Suh. Efficient controllable multi-task architectures. In *ICCV*, 2023.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021.

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *ICLR*, 2018.

Roman Bachmann, Oğuzhan Fatih Kar, David Mizrahi, Ali Garjani, Mingfei Gao, David Griffiths, Jiaming Hu, Afshin Dehghan, and Amir Zamir. 4m-21: An any-to-any vision model for tens of tasks and modalities. In *NeurIPS*, 2024.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. Adversarial training for multi-context joint entity and relation extraction. In *NeurIPS*, 2018.

Victor Besnier and Mickael Chen. A pytorch reproduction of masked generative image transformer. *arXiv preprint arXiv:2310.14400*, 2023.

Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Bayesian prediction of future street scenes using synthetic likelihoods. In *ICLR*, 2019.

Felix JS Bragman, Ryutaro Tanno, Sebastien Ourselin, Daniel C. Alexander, and Jorge Cardoso. Stochastic filter groups for multi-task CNNs: Learning specialist and generalist convolution kernels. In *ICCV*, 2019.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1:8, 2024.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *CVPR*, 2019.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Tianlong Chen, Xuxi Chen, Xianzhi Du, Abdullah Rashwan, Fan Yang, Huizhong Chen, Zhangyang Wang, and Yeqing Li. Adamv-moe: Adaptive multi-task vision mixture-of-experts. In *ICCV*, 2023a.

Xin Chen and Yahong Han. Multi-timescale context encoding for scene parsing prediction. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 2019.

Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik G. Learned-Miller, and Chuang Gan. Mod-squad: Designing mixtures of experts as modular multi-task learners. In *CVPR*, 2023b.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

Hsu-kuang Chiu, Ehsan Adeli, and Juan Carlos Niebles. Segmenting the future. *IEEE Robotics and Automation Letters*, 2020.

 https://doi.org/10.52202/085713-5466

Wonhyeok Choi and Sunghoon Im. Dynamic neural network for multi-task learning searching across diverse network topologies. In *CVPR*, 2023.

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 *CVPR*, 2016.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024.

Alexey Dosovitskiy and Vladlen Koltun. Learning to act by predicting the future. In *ICLR*, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.

Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, 2023.

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 2024.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *NeurIPS*, 2016.

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 *NeurIPS*, 2024.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. Simvp: Simpler yet better video prediction. In *CVPR*, 2022.

Spyros Gidaris, Andrei Bursuc, Oriane Siméoni, Antonín Vobecký, Nikos Komodakis, Matthieu Cord, and Patrick Perez. MOCA: Self-supervised representation learning by predicting masked online codebook assignments. *Transactions on Machine Learning Research*, 2024.

Rohit Girdhar, Alaaeldin El-Nouby, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. OmniMAE: Single Model Masked Pretraining on Images and Videos . In *CVPR*, 2023.

Colin Graber, Grace Tsai, Michael Firman, Gabriel Brostow, and Alexander Schwing. Panoptic Segmentation Forecasting . In *CVPR*, 2021.

Colin Graber, Cyril Jazra, Wenjie Luo, Liangyan Gui, and Alexander G Schwing. Joint forecasting of panoptic segmentations with difference attention. In *CVPR*, 2022.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent a new approach to self-supervised learning. In *NeurIPS*, 2020.

Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *ICML*, 2020.

Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. In *ICLR*, 2023.

William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Dietrich Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. In *NeurIPS*, 2022.

Jing He, Haodong LI, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Zhang, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. In *ICLR*, 2025.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Greg Heinrich, Mike Ranzinger, Hongxu Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. Radiov2.5: Improved baselines for agglomerative vision foundation models. In *CVPR*, 2025.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *NeurIPS*, 2022b.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. In *ICLR*, 2023.

Anthony Hu, Fergal Cotter, Nikhil Mohan, Corina Gurau, and Alex Kendall. Probabilistic future prediction for video scene understanding. In *ECCV*, 2020.

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.

Jian-Fang Hu, Jiangxin Sun, Zihang Lin, Jian-Huang Lai, Wenjun Zeng, and Wei-Shi Zheng. Apanet: Auto-path aggregation for future instance segmentation prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Xiaojie Jin, Xin Li, Huaxin Xiao, Xiaohui Shen, Zhe Lin, Jimei Yang, Yunpeng Chen, Jian Dong, Luoqi Liu, Zequn Jie, et al. Video scene parsing with predictive feature learning. In *ICCV*, 2017.

Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzalos, and Nikos Komodakis. What to hide from your students: Attention-guided masked image modeling. In *ECCV*, 2022.

Ioannis Kakogeorgiou, Spyros Gidaris, Konstantinos Karantzalos, and Nikos Komodakis. Spot: Self-training with patch-order permutation for object-centric learning with autoregressive transformers. In *CVPR*, 2024.

Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Advancing semantic future prediction through multimodal visual sequence transformers. In *CVPR*, 2025.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *CVPR*, 2023.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, 2020.

Dan Kondratyuk, Lijun Yu, Xiuye Gu, Jose Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, Krishna Somandepalli, Hassan Akbari, Yair Alon, Yong Cheng, Joshua V. Dillon, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David Minnen, Mikhail Sirotenko, Kihyuk Sohn, Xuan Yang, Hartwig Adam, Ming-Hsuan Yang, Irfan Essa, Huisheng Wang, David A Ross, Bryan Seybold, and Lu Jiang. Videopoet: A large language model for zero-shot video generation. In *ICML*, 2024.

Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

Sangmin Lee, Hak Gu Kim, Dae Hwi Choi, Hyung-Il Kim, and Yong Man Ro. Video prediction recalling long-term motion context via memory alignment learning. In *CVPR*, 2021.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *NeurIPS*, 2024.

Xiwen Liang, Yangxin Wu, Jianhua Han, Hang Xu, Chunjing Xu, and Xiaodan Liang. Effective adaptation in multi-task co-training for unified autonomous driving. In *NeurIPS*, 2022.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

Zihang Lin, Jiangxin Sun, Jian-Fang Hu, Qizhi Yu, Jian-Huang Lai, and Wei-Shi Zheng. Predictive feature learning for future segmentation prediction. In *CVPR*, 2021.

 https://doi.org/10.52202/085713-5466

Yen-Cheng Liu, CHIH-YAO MA, Junjiao Tian, Zijian He, and Zsolt Kira. Polyhistor: Parameter-efficient multi-task adaptation for dense vision tasks. In *NeurIPS*, 2022.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *CVPR*, 2017.

Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. In *ICCV*, 2017.

Pauline Luc, Camille Couprie, Yann Lecun, and Jakob Verbeek. Predicting future instance segmentation by forecasting convolutional features. In *ECCV*, 2018.

Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *CVPR*, 2019.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016.

Seyed Shahabeddin Nabavi, Mrigank Rochan, and Yang Wang. Future semantic segmentation with convolutional lstm. In *BMVC*, 2018.

Marina Neseem, Ahmed Agiza, and Sherief Reda. AdaMTL: Adaptive Input-dependent Inference for Efficient Multi-Task Learning. In *CVPRW*, 2023.

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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *CVPR*, 2021.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *NeurIPS*, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *AAAI*, 2019.

Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, Jitendra Malik, Yanghao Li, and Christoph Feichtenhofer. Hiera: a hierarchical vision transformer without the bells-and-whistles. In *ICML*, 2023.

Josip Šarić, Marin Oršić, Tonći Antunović, Sacha Vražić, and Siniša Šegvić. Single level feature-to-feature forecasting with deformable convolutions. In *41st DAGM German Conference on Pattern Recognition*, 2019.

Josip Saric, Marin Orsic, Tonci Antunovic, Sacha Vrazic, and Sinisa Segvic. Warp to the future: Joint forecasting of features and feature motion. In *CVPR*, 2020.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *NeurIPS*, 2018.

Sophia Sirko-Galouchenko, Spyros Gidaris, Antonin Vobecky, Andrei Bursuc, and Nicolas Thome. Dip: Unsupervised dense in-context post-training of visual representations. In *ICCV*, 2025.

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *CVPR*, 2021.

Jiangxin Sun, Jiafeng Xie, Jian-Fang Hu, Zihang Lin, Jianhuang Lai, Wenjun Zeng, and Wei-Shi Zheng. Predicting future instance segmentation with contextual pyramid convlstms. In *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

Adam Terwilliger, Garrick Brazil, and Xiaoming Liu. Recurrent flow-guided semantic forecasting. In *WACV*, 2019.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022.

Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Herve Jegou. Fixing the train-test resolution discrepancy. In *NeurIPS*, 2019.

Michael Tschannen, Cian Eastwood, and Fabian Mentzer. Givt: Generative infinite-vocabulary transformers. In *ECCV*, 2024.

Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Shashanka Venkataramanan, Valentinos Pariza, Mohammadreza Salehi, Lukas Knobel, Spyros Gidaris, Elias Ramzi, Andrei Bursuc, and Yuki M Asano. Franca: Nested matryoshka clustering for scalable visual representation learning. *arXiv preprint arXiv:2507.14137*, 2025.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016a.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016b.

Suhani Vora, Reza Mahjourian, Soeren Pirk, and Anelia Angelova. Future semantic segmentation using 3d structure. *arXiv preprint arXiv:1811.11358*, 2, 2018.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *CVPR*, 2023.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *ICLR*, 2018.

Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, 2022.

Haixu Wu, Zhiyu Yao, Jianmin Wang, and Mingsheng Long. Motionrnn: A flexible model for video prediction with spacetime-varying motions. In *CVPR*, 2021.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

Jingwei Xu, Bingbing Ni, Zefan Li, Shuo Cheng, and Xiaokang Yang. Structure preserving video prediction. In *CVPR*, 2018.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024a.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *NeurIPS*, 2024b.

Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *CVPR*, 2023.

 https://doi.org/10.52202/085713-5466

Lijun Yu, Jose Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A Ross, and Lu Jiang. Language model beats diffusion - tokenizer is key to visual generation. In *ICLR*, 2024.

Lijun Zhang, Xiao Liu, and Hui Guan. Automtl: A programming framework for automating efficient multi-task learning. In *NeurIPS*, 2022.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *ECCV*, 2024.

Zeyun Zhong, David Schneider, Michael Voit, Rainer Stiefelhagen, and Jürgen Beyerer. Anticipative feature fusion transformer for multi-modal action anticipation. In *WACV*, 2023.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *ICML*, 2025.

Appendix

A Additional Results

A.1 Impact of Dimensionality Reduction

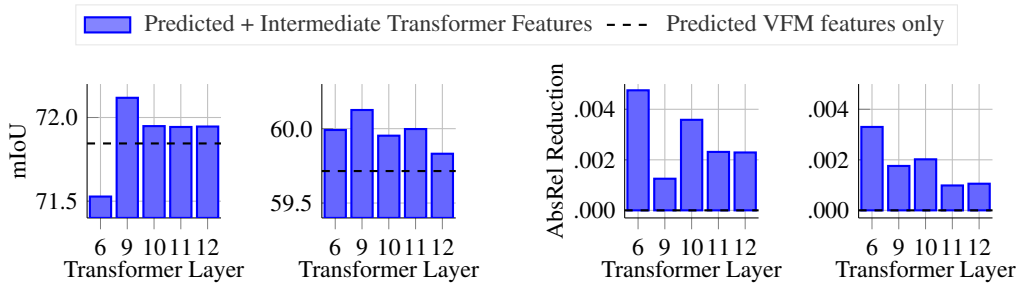
In our work, we examine a PCA-based dimensionality reduction method and find that compressing features in this way does not compromise performance on semantic segmentation and depth prediction downstream tasks (Table 5). In fact, reducing the dimensionality simplifies the modeling process and leads to improved performance. Specifically, for semantic segmentation forecasting, PCA enhances short-term predictions—particularly for moving objects—while its effect on mid-term predictions is negligible. Similarly, for depth forecasting, dimensionality reduction consistently boosts performance across all metrics for both short- and mid-term predictions.

Table 5: **Impact of Dimensionality Reduction**. Reduction is performed using PCA. Results on semantic segmentation and depth forecasting.

Dim. Reduction	Segmentation				Depth			
	Short		Mid		Short		Mid	
	ALL↑	MO↑	ALL↑	MO↑	δ_1 ↑	AbsR↓	δ_1↑	AbsR↓
✗	71.3	70.4	**59.9**	**57.6**	87.9	.122	84.8	.147
✓	**71.8**	**71.7**	59.8	**57.6**	**88.6**	**.114**	**85.4**	**.136**

A.2 Emerging Visual Representations in the Future-Frame Masked Feature Transformer

Self-supervised representation learning has achieved remarkable progress, with numerous studies focusing on extracting robust visual features from unlabeled images and videos (Bardes et al., 2024; Tong et al., 2022; Girdhar et al., 2023; Ryali et al., 2023; Wang et al., 2023; Chen et al., 2020; Caron et al., 2021; He et al., 2022; Grill et al., 2020; Wei et al., 2022; Kakogeorgiou et al., 2024, 2022; Gidaris et al., 2024; Venkataramanan et al., 2025; Sirko-Galouchenko et al., 2025). Inspired by these advancements, we investigate the potential of our future-frame masked feature transformer as a self-supervised method for enhancing VFM visual features. Specifically, we train DPT heads for semantic segmentation and depth prediction, using not only the features predicted by the masked feature transformer but also additional features extracted from intermediate transformer layers. We examine features from the 6th, 9th, 10th, 11th, and 12th (last) layers of the transformer to assess whether these intermediate representations can further improve the strong VFM features predicted by our masked transformer.



(a) Short-term, Segm. (b) Mid-term, Segm (c) Short-term, Depth (d) Mid-term, Depth

Figure 4: **Impact of Intermediate Transformer Features on Future Segmentation and Depth Prediction.** Results are shown for semantic segmentation and depth prediction heads using two feature sets: only the VFM features predicted by the masked feature transformer (dashed line) and combined features from both predicted and intermediate transformer layers (blue bars). We evaluate DPT heads trained on features from the 6th, 9th, 10th, 11th, and 12th layers. For segmentation (barplots (a) and (b)), we report mIoU across all classes. For depth (barplots (c) and (d)), we show the reduction in AbsRel metric (higher is better) when adding intermediate layer features.

https://doi.org/10.52202/085713-5466

Results, shown in Figure 4, indicate that incorporating intermediate transformer features from the masked transformer enhances segmentation performance, with one exception (6th-layer features for short-term segmentation). Notably, the best segmentation results are achieved using features from the 9th layer. Although the improvements are modest, this aligns with expectations given the strength of the predicted VFM features alone. Similar results are observed for future depth prediction, where intermediate features also led to performance gains, with the best depth results achieved using 6th-layer features (see Figure 4). While exploring self-supervised learning was not the primary aim of our work, we find these results intriguing, as they suggest that future prediction methods hold promise as self-supervised visual representation learners. We hope this work can spark further research into this direction.

A.3 Additional Comparisons with VISTA

In Table 6, we present a comprehensive evaluation comparing our method against VISTA (Gao et al., 2024) for semantic segmentation, instance segmentation, depth and surface normal estimation on the Cityscapes dataset. We extend this evaluation to the nuScenes (Caesar et al., 2020) dataset in Table 7 for depth and surface normals. Notably, DINO-Foresight demonstrates strong zero-shot generalization capabilities. When trained only on Cityscapes and directly evaluated on nuScenes without fine-tuning, performance degradation is minimal, while still significantly outperforming all baselines. The results show that DINO-Foresight consistently surpasses the performance of VISTA on both datasets.

Table 6: **Comparison across tasks on Cityscapes**. We use DINOv2 encoder and show performance on segmentation (ALL, MO), instance segmentation (AP50, AP), depth estimation (δ_1 accuracy, AbsRel error), and surface normal prediction (m, percentage within 11.25°). VISTA$_{ft}$ is the VISTA model fine-tuned on Cityscapes.

Method	Semantic Segm.				Instance Segm				Depth				Surface Normals			
	Short		Mid		Short		Mid		Short		Mid		Short		Mid	
	ALL↑	MO↑	ALL↑	MO↑	AP50↑	AP↑	AP50↑	AP↑	δ_1 ↑	AbsR↓	δ_1↑	AbsR↓	m↓	11.25°↑	m↓	11.25°↑
Oracle	77.0	77.4	77.0	77.4	66.2	40.4	66.2	40.4	89.1	.108	89.1	.108	3.24	95.3	3.24	95.3
Copy Last	54.7	52.0	40.4	32.3	24.7	10.4	9.5	2.8	84.1	.154	77.8	.212	4.41	89.2	5.39	84.0
VISTA$_{ft}$	64.9	62.1	53.9	51.0	33.1	17.7	19.8	9.0	86.4	.124	82.8	.153	3.75	93.0	4.30	90.0
DINO-Foresight	**71.8**	**71.7**	**59.8**	**57.6**	**50.5**	**26.6**	**27.3**	**12.6**	**88.6**	**.114**	**85.4**	**.136**	**3.39**	**94.4**	**4.00**	**91.3**

Table 7: **Comparison across tasks at nuScenes**. We use DINOv2 encoder and show performance on depth estimation (δ_1 accuracy, AbsRel error) and surface normal prediction (m, percentage within 11.25°). VISTA$_{ft}$ is the VISTA model fine-tuned on nuScenes. The last row evaluates zero-shot generalization: DINO-Foresight trained on Cityscapes and directly evaluated on nuScenes.

Method	Depth				Surface Normals			
	Mid		Long		Mid		Long	
	δ_1 ↑	AbsR↓	δ_1 ↑	AbsR↓	m↓	11.25°↑	m↓	11.25°↑
Oracle	82.6	.206	82.6	.206	3.09	97.1	3.09	97.1
Copy Last	73.4	.353	68.4	.468	4.66	88.6	5.31	85.3
VISTA$_{ft}$	74.6	.337	70.8	.421	4.46	90.8	4.96	88.2
DINO-Foresight	**80.7**	**.218**	**76.3**	**.299**	**3.59**	**93.9**	**4.22**	**90.6**
DINO-Foresight (zero-shot)	78.4	.269	72.1	.377	4.03	92.3	4.77	88.4

A.4 Impact of Multi-Layer Features

Scene understanding models benefit from utilizing features from multiple layers of a frozen image encoder. To fully exploit the pretrained DINO features, we integrate representations from several layers into the DPT head. Table 8 presents an ablation comparing multi-layer DINO features to using only the final layer (layer 12) for semantic segmentation on Cityscapes. The results demonstrate that aggregating features from layers 3, 6, 9, and 12 enhances performance, with ALL (all semantic classes) and MO (movable objects classes) scores rising from 72.1/73.4 to 77.0/77.4.

https://doi.org/10.52202/085713-5466

Table 8: **DINO+DPT Features Ablation**. Ablation of multi-layer DINO features as input to the DPT head versus using only the last layer features, evaluated on semantic segmentation on Cityscapes.

Layers	Segmentation	
	ALL	MO
12	72.1	73.4
3,6,9,12	77.0	77.4

A.5 Masking strategy and loss function ablations

We evaluate the impact of masking strategies and loss functions on forecasting performance. As shown in Table 9, full masking (masking all future features) consistently outperforms random masking across all metrics, for semantic segmentation and depth forecasting. This demonstrates that masking all future features forces the model to learn more robust temporal dynamics. Regarding loss functions (Table 10), we find that our framework is robust to loss function choice, with L1, MSE, SmoothL1, and SmoothL1+Cosine achieving comparable performance.

Table 9: **Impact of Masking Strategies on Cityscapes**. We compare random masking versus full masking for semantic segmentation, depth estimation, and surface normal prediction. Full masking demonstrates consistently superior performance across all metrics.

Masking Strategy	Semantic Segmentation				Depth				Surface Normals			
	Short		Mid		Short		Mid		Short		Mid	
	ALL↑	MO↑	ALL↑	MO↑	δ_1 ↑	AbsR↓	δ_1↑	AbsR↓	m↓	11.25°↑	m↓	11.25°↑
Random Masking	70.5	70.2	58.0	55.7	88.2	.121	84.7	.148	3.48	94.0	4.11	90.8
Full Masking	**71.8**	**71.7**	**59.8**	**57.6**	**88.6**	**.114**	**85.4**	**.136**	**3.39**	**94.4**	**4.00**	**91.3**

Table 10: **Loss Function Comparison on Cityscapes**. We evaluate different loss functions (L1, MSE, SmoothL1, SmoothL1+Cosine) for semantic segmentation, depth estimation, and surface normal prediction. Results demonstrate that our framework is robust to loss function choice, with all variants achieving comparable performance.

Loss Function	Semantic Segmentation				Depth				Surface Normals			
	Short		Mid		Short		Mid		Short		Mid	
	ALL↑	MO↑	ALL↑	MO↑	δ_1 ↑	AbsR↓	δ_1↑	AbsR↓	m↓	11.25°↑	m↓	11.25°↑
L1	71.7	71.7	59.7	57.6	88.6	.118	**85.6**	.138	3.41	94.3	4.00	91.3
MSE	71.7	**71.9**	**60.0**	**57.8**	88.6	.117	85.4	.138	3.40	**94.4**	**3.99**	**91.4**
SmoothL1+Cos	71.7	71.4	59.8	57.5	**88.7**	.116	85.5	.137	3.40	**94.4**	**3.98**	**91.4**
SmoothL1	**71.8**	71.7	59.8	57.6	88.6	**.114**	85.4	**.136**	**3.39**	**94.4**	4.00	91.3

A.6 Model Size and Data Scale Scalability

We investigate how performance scales with model size and training data diversity. As shown in Table 11, we evaluate three model variants—Small (115M), Base (258M), and Large (460M) parameters—by modifying the hidden dimension and number of attention heads while keeping dataset size and training duration fixed. Results demonstrate consistent performance improvements with increased model capacity, particularly for mid-term predictions, indicating that larger models better capture complex temporal dynamics in VFM features. Regarding data scale (Table 12), we combine Cityscapes and nuScenes datasets with equal-probability sampling during training and evaluate on Cityscapes. The model trained on combined datasets achieves consistent improvements across all tasks compared to training on Cityscapes alone, with gains particularly pronounced for mid-term semantic segmentation. These findings demonstrate that our framework effectively scales with both increased model capacity and diverse training data, motivating further exploration with larger models and datasets to fully realize the potential of forecasting VFM features for multi-task scene understanding.

https://doi.org/10.52202/085713-5466

Table 11: **Model Size Scalability on Cityscapes**. We evaluate three model sizes—Small (115M), Base (258M), and Large (460M) parameters—across semantic segmentation, depth estimation, and surface normal prediction. Results demonstrate consistent performance improvements with increased model capacity.

Model Variant	Hidden Dim	Att Heads	Semantic Segm.				Depth				Surface Normals			
			Short		Mid		Short		Mid		Short		Mid	
			ALL↑	MO↑	ALL↑	MO↑	δ_1 ↑	AbsR↓	δ_1↑	AbsR↓	m↓	11.25°↑	m↓	11.25°↑
Small (115M)	768	6	71.1	70.8	59.3	57.3	87.7	.125	84.8	.142	3.58	93.7	4.13	90.8
Base (258M)	1152	8	71.8	**71.7**	59.8	57.6	**88.6**	**.114**	85.4	**.136**	**3.39**	**94.4**	**4.00**	91.3
Large (460M)	1536	12	**71.9**	**71.7**	**60.2**	**58.3**	**88.6**	.116	**85.5**	.137	3.40	**94.4**	**4.00**	**91.5**

Table 12: **Data Scale Scalability**. We compare training on Cityscapes alone versus training on combined Cityscapes+nuScenes data. Results demonstrate that increasing training data diversity leads to improved performance across all tasks and temporal horizons.

Training Data	Semantic Segmentation				Depth				Surface Normals			
	Short		Mid		Short		Mid		Short		Mid	
	ALL↑	MO↑	ALL↑	MO↑	δ_1 ↑	AbsR↓	δ_1↑	AbsR↓	m↓	11.25°↑	m↓	11.25°↑
Cityscapes	71.8	71.7	59.8	57.6	**88.6**	**.114**	85.4	**.136**	**3.39**	**94.4**	4.00	91.3
Cityscapes+nuScenes	**72.3**	**72.2**	**61.0**	**59.4**	**88.6**	.117	**85.7**	**.136**	3.41	**94.4**	**3.95**	**91.6**

A.7 More Visualizations

In Figure 5 and 6, we present additional qualitative results illustrating the prediction of semantic segmentation, depth maps, and surface normals. Specifically, we compare our method, `DINO-Foresight`, against the `Oracle`, which involves using future RGB frames as inputs for different prediction heads, as well as `VISTA` Gao et al. (2024). As illustrated in Figure 5, `DINO-Foresight` maintains superior integrity of motion dynamics and geometric consistency across frames, resulting in more accurate predictions

In Figure 7 and 8, we offer additional qualitative results derived from utilizing `DINO-Foresight` for the prediction of semantic segmentation and depth maps and surface normals over extended time intervals. These outcomes are achieved through the use of autoregressive rollouts. Beginning with a series of four context frames (X_{t-9} to X_t), the model is capable of predicting up to 48 subsequent frames, equivalent to 2.88 seconds, with predictions occurring at an interval of every third frame. Our model consistently delivers accurate predictions over the entire forecasted duration, effectively capturing motion dynamics and maintaining consistency across different modalities. This performance underscores its robustness and versatility, which are related to its capability of predicting the features of a foundation model. As a final remark, it is important to note that the noisy segmentation predictions observed at the bottom of the images, present in both the predicted and `Oracle` results, are attributed to unannotated regions in the Cityscapes dataset that are disregarded during DPT training. This artifact impacts only the segmentation outcomes of DPT head and does not affect the predicted future features, as evidenced by the clear and accurate depth and surface normal predictions.

B Limitations and Future Work

In this work, we introduced `DINO-Foresight`, a simple yet effective method for semantic future prediction based on forecasting VFM features. Our approach delivers strong results while opening several exciting directions for future research.

First, our method uses a straightforward masked transformer with SmoothL1 loss. While forecasting VFM features avoids the challenges of modeling complex pixel distributions, our current implementation is deterministic. However, our framework can easily be extended to capture uncertainty—for example, by adding a diffusion loss (as in MAR (Li et al., 2024)) or modeling tokens with a Gaussian mixture model (as in GIVT (Tschannen et al., 2024)). These extensions would better handle future ambiguity while maintaining the simplicity of our approach.

Although we explored strategies to reduce training compute for high-resolution feature prediction, inference-time compute demands remain unchanged. Future work could address this by adopting

hierarchical transformer architectures (Ryali et al., 2023), which would not only improve efficiency but also enable the model to handle even higher feature resolutions.

Another promising direction is scaling `DINO-Foresight` to larger datasets and models. Our experiments on model size (115M to 460M parameters) and data diversity (Cityscapes+nuScenes) demonstrate consistent performance improvements, particularly for mid-term predictions, suggesting that further scaling to even larger models and more diverse datasets could yield substantial gains in forecasting performance.. Furthermore, the flexibility of our framework allows seamless integration of newer VFM encoders, such as RADIOv2.5 (Heinrich et al., 2025), which combines multiple VFMs into a single, more powerful model, enhancing its multi-task future scene understanding capabilities.

Overall, these research directions highlight the flexibility and growth potential of our approach, paving the way for further advancements in semantic future prediction.

C Broader Impact

Our work enables efficient and scalable semantic future prediction by forecasting semantically rich VFM features. This allows flexible integration with different scene understanding tasks without retraining, making it useful for applications like autonomous driving and robotics. While we do not foresee risks in our approach, we must remain mindful that the pretrained Vision Foundation Models we build upon may carry biases, which could potentially influence our semantic future predictions.

D Implementation Details

We provide implementation details for the heads trained on different downstream tasks. The DPT head is used for semantic segmentation, depth estimation, and surface normal estimation. We adopt the DPT Ranftl et al. (2021) implementation from Depth Anything Yang et al. (2024a,b), setting the feature dimensionality to 256 and configuring `dptoutchannels = [128, 256, 512, 512]`. For all tasks, models are trained for 100 epochs with a batch size of 128 (16×8 GPUs). The learning rate is set to 0.0016, using the AdamW optimizer with linear warmup for the first 10 epochs, and weight decay is 0.0001. For semantic segmentation, we use a polynomial scheduler and cross-entropy loss with 19 classes. For depth estimation, we use a cosine annealing scheduler and cross-entropy loss, with 256 classes. For surface normal estimation, we employ a polynomial scheduler and a loss function combining cosine similarity and L_2 loss with weighted averaging, using 3 classes.

For the Mask2Former head used in instance segmentation, we implement our approach using the official Mask2Former Cheng et al. (2022) and Detectron2 Wu et al. (2019) codebases. The main difference compared to the official Mask2Former configuration for Cityscapes instance segmentation is the input feature maps. In our approach, the four multi-scale feature maps expected by Mask2Former are derived from the PCA features. These PCA features are first projected to 128, 256, 512, and 1024 dimensions and then resized so their spatial resolutions are $\times 4$, $\times 2$, $\times 1$, and $\times 0.5$ relative to the original resolution of the DINOv2 ViT-B outputs. We train using the AdamW optimizer, with a batch size of 64 (8×8 GPUs), learning rate of 0.00032, weight decay of 0.05, and 67,500 iterations, with a polynomial scheduler.

Regarding Vista, we fine-tuned the model with 8 frames in total (3 as context and 5 future frames) for cityscapes, while for Nuscenes we used 9 frames in total (3 as context and 6 future frames) to support long-term forecasting

E Definitions of Evaluation Metrics

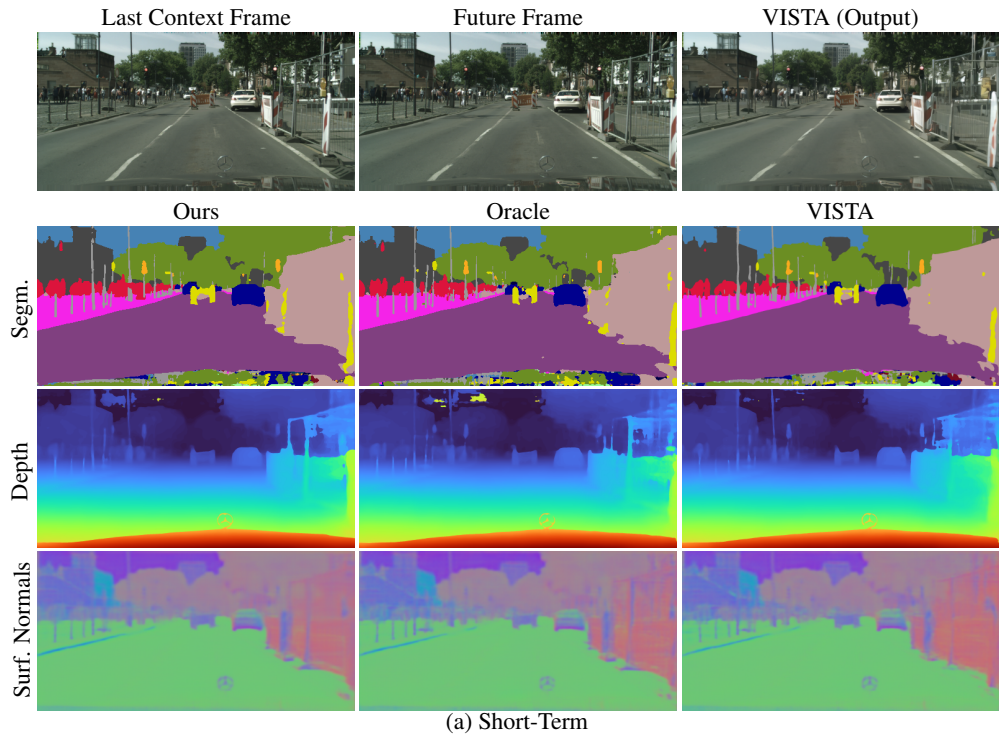
For **depth prediction**, we use two metrics: the mean Absolute Relative Error (AbsRel), defined as $\frac{1}{M}\sum_{i=1}^{M}\frac{|a_i - b_i|}{b_i}$, where a_i and b_i are the predicted and ground truth disparities at pixel i, and M is the number of pixels. We also evaluate depth accuracy using δ_1, the percentage of pixels where $\max\left(\frac{a_i}{b_i}, \frac{b_i}{a_i}\right) < 1.25$. For **surface normal evaluation**, we compute the mean angular error m$\downarrow$ as $\frac{1}{N}\sum_{i=1}^{N}\cos^{-1}\left(\frac{\mathbf{n}_i \cdot \tilde{\mathbf{n}}_i}{\|\mathbf{n}_i\|\,\|\tilde{\mathbf{n}}_i\|}\right)$, where $\mathbf{n}_i$ and $\tilde{\mathbf{n}}_i$ are the predicted and ground truth normals,

 https://doi.org/10.52202/085713-5466

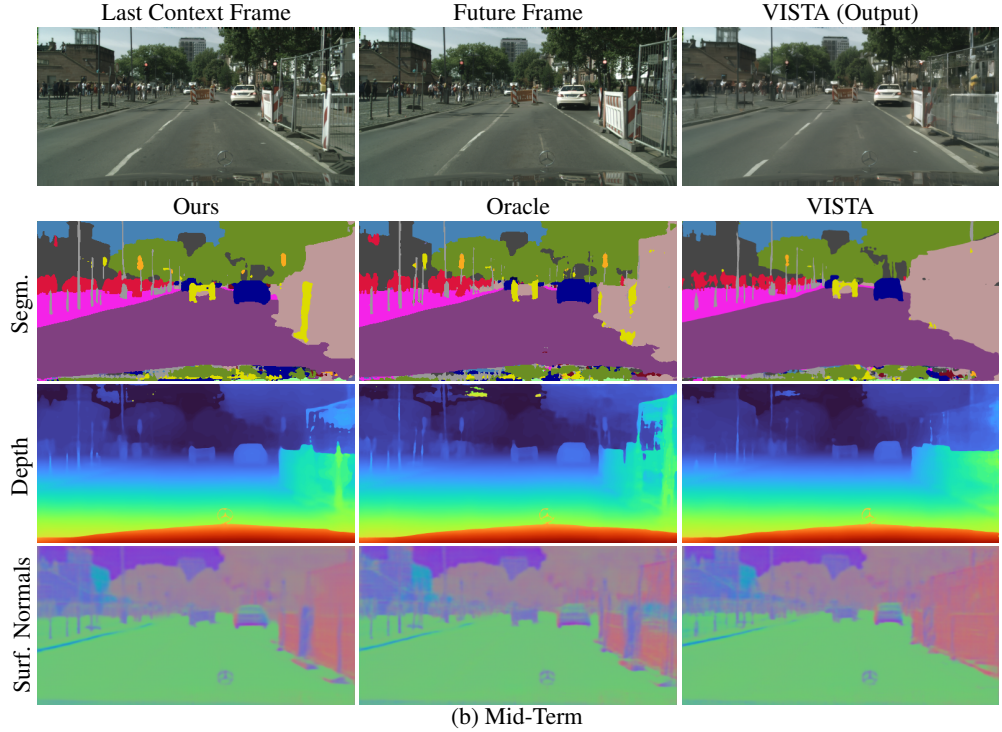
respectively. Furthermore, we measure precision through the percentage of pixels with angular errors below 11.25°, calculated as $\left(\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\theta_i < 11.25°)\right)$

F Details of Evaluation Scenarios

Regarding Cityscapes, the target frame for both short and mid term prediction is 20. We subsample sequences by a factor of 3 before inputting to the model. For short-term prediction, the model uses frames 8, 11, 14, and 17 as context to predict frame 20 (with context length $N_c = 4$ and $N_p = 1$). For mid-term prediction, the model uses frames 2, 5, 8, and 11 as context and predicts frame 20 auto-regressively through frames 14 and 17. We calculate segmentation metrics on the 20th frame using Cityscapes ground truth. For depth and surface normals, we rely on pseudo-annotations from DepthAnythingV2 Yang et al. (2024b) and Lotus He et al. (2025), respectively, due to the lack of true annotations in Cityscapes. Regarding nuScenes, the target frame for both mid and long term prediction is 29. Again, we subsample sequences by a factor of 3 before input to the model. For mid-term prediction, the model uses frames 11, 14, 17, and 20 as context and predicts frame 29 auto-regressively through frames 23 and 26. For long-term prediction, the model uses frames 2, 5, 8, and 11 as context and predicts frame 29 auto-regressively through frames 14,17,20,23 and 26. Again for depth and surface normals, we rely on pseudo-annotations from DepthAnythingV2 Yang et al. (2024b) and Lotus He et al. (2025), respectively, due to the lack of true annotations in nuScenes.



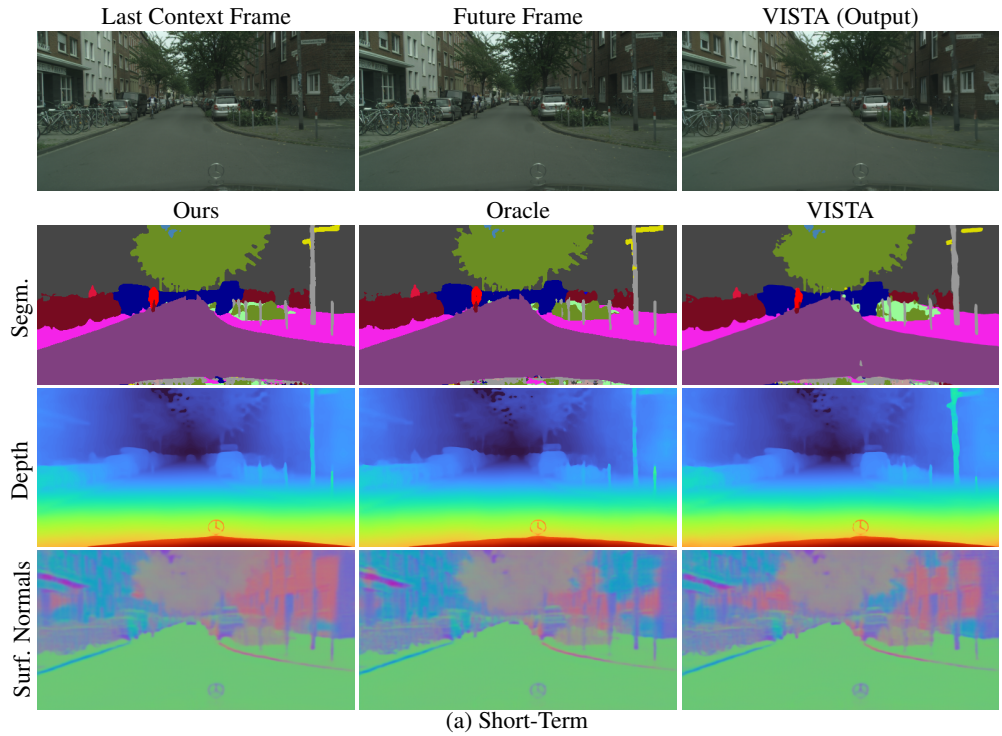
(a) Short-Term



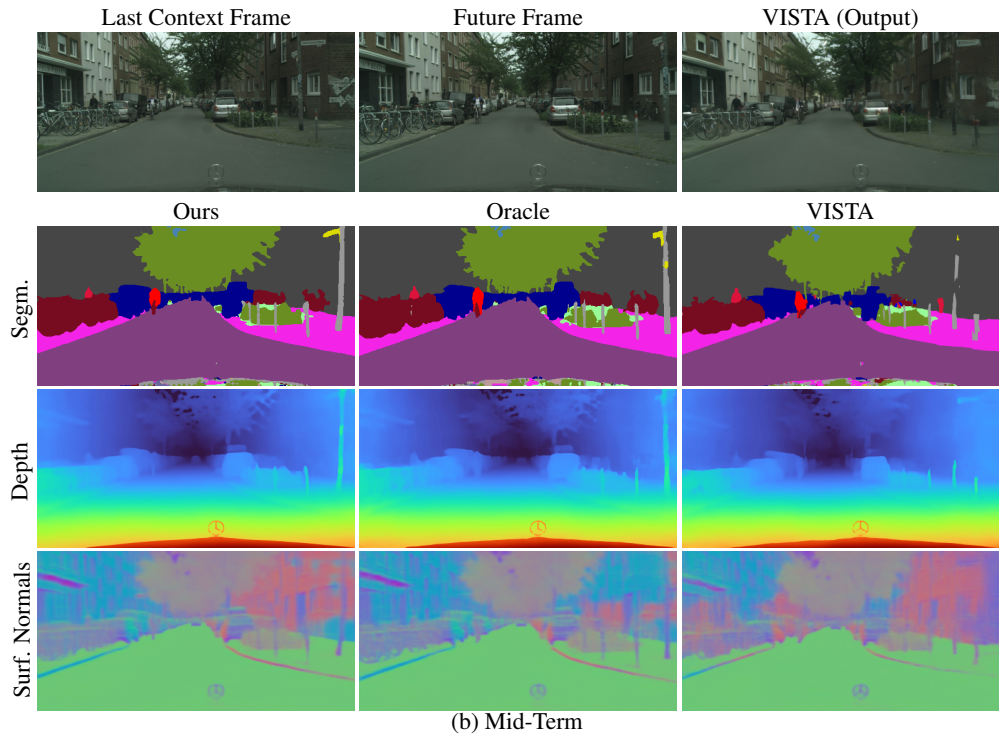
(b) Mid-Term

Figure 5: **Visualization of future predictions for semantic segmentation, depth, and surface normals.** The illustrated scene is Frankfurt (01 (017082-017111)).

https://doi.org/10.52202/085713-5466



(a) Short-Term



(b) Mid-Term

Figure 6: **Visualization of future predictions for semantic segmentation, depth, and surface normals**. The illustrated scene is Munster (15).

https://doi.org/10.52202/085713-5466

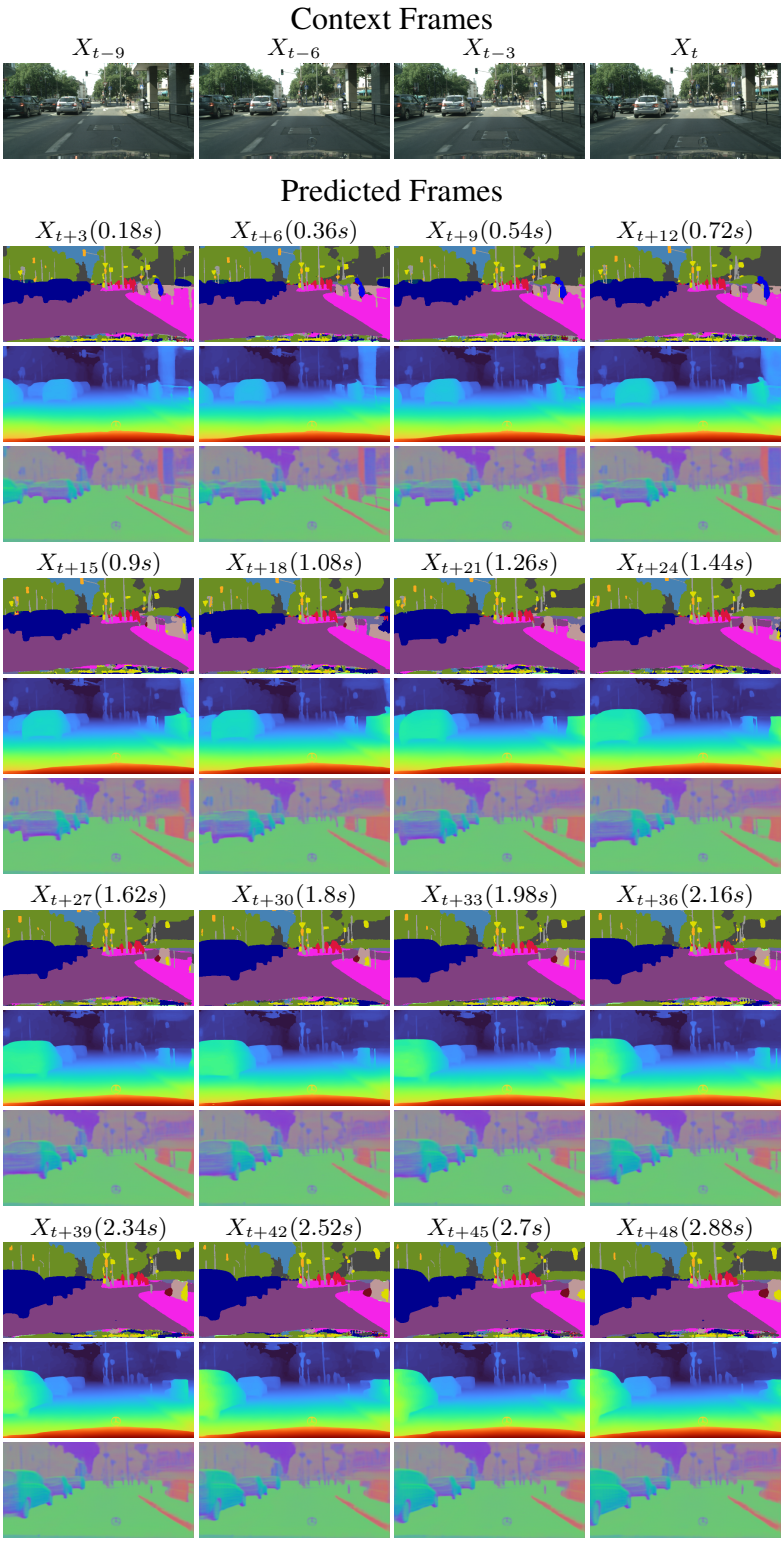


Figure 7: **Long-term semantic segmentation, depth and surface normal predictions.** The illustrated scene is Frankfurt (01 (011791-011820)). `DINO-Foresight` consistently preserves motion dynamics and intricate details in complex scenes over extended time horizons.

https://doi.org/10.52202/085713-5466

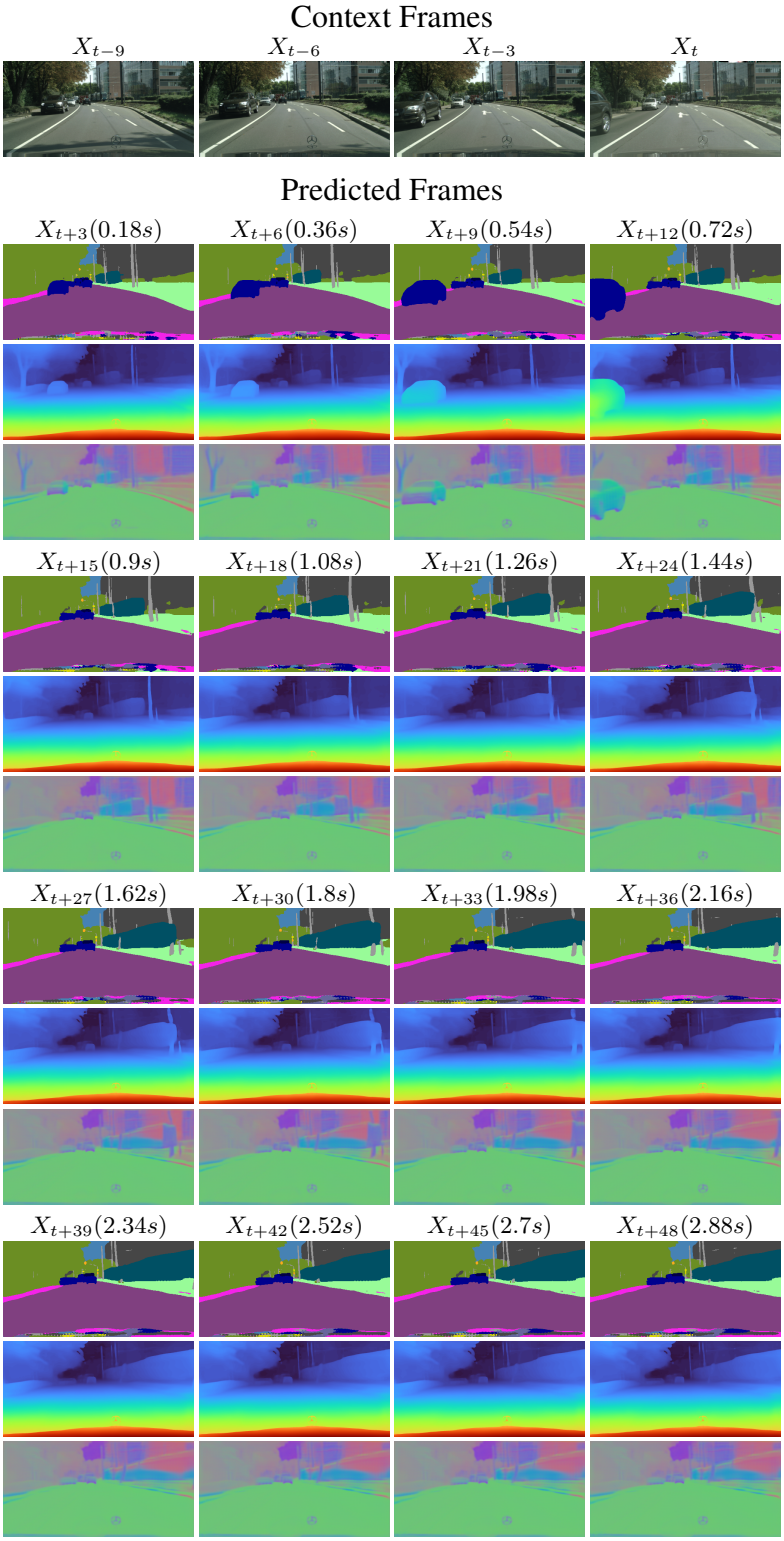


Figure 8: **Long-term semantic segmentation, depth and surface normal predictions**. The illustrated scene is Frankfurt (01 (006570-006599)). `DINO-Foresight` excels in predicting the motion of the nearby car but faces challenges with distant, low-motion objects, highlighting areas for future improvement.

https://doi.org/10.52202/085713-5466

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately describes the proposed method and align with the results presented in Table 1, Table 2 and Table 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations of our work in Appendix Subsection B.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

https://doi.org/10.52202/085713-5466

Justification: We do not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all the implementation details for our experiments in section 4 and in the Appendix Subsection D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will publicly release our code upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: To our knowledge, we have included all required dataset splits, hyperparameters and experimental specifications needed to reproduce the resuls.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Running large generative models multiple times with different random seeds demands computational resources beyond our capacity. Statistical significance testing remains uncommon in the field of large image/video generative models.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

https://doi.org/10.52202/085713-5466

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify the hardware used (8xA100 40G GPUs) for our experiments in section 4 and in the Appendix Subsection D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the code of ethics and made sure that this paper conforms to it in every aspect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the societal impact of our work in Appendix Subsection C.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The proposed method for semantic future prediction do not present significant misuse concerns that would require specialized release safeguards.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We appropriately cite all code implementations, datasets and models utilized in conducting our experiments and evaluating our models.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

 https://doi.org/10.52202/085713-5466

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The source code will be open sourced upon acceptance.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We not do involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We not involve human subjects in our work, therefore IRB approval is not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLM usage doe not impacts the core methodology, scientific rigorousness and originality of this work.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

https://doi.org/10.52202/085713-5466