A Frame is Worth One Token: Efficient Generative World Modeling with Delta Tokens

Supplementary Material

Appendix

Table of contents:

A. Additional Implementation Details

DINO-world reimplementation. An official DINO-world codebase has not been released, so all DINO-world baselines in this paper use our own reimplementation following the protocol described in DINO-world [2]. Specifically, spatial and temporal identity are injected through axial rotary positional embeddings (3D RoPE [17]) applied to the query and key projections, rotating the first $20{+}20{+}20$ dimensions per head and leaving the final 4 unrotated. Furthermore, spatial predictions of frame $t{+}1$ are computed using a block-causal attention mask during training, ensuring queries only attend to past frames while allowing efficient parallelization.

DeltaTok tokenizer. Our *DeltaTok* tokenizer is a simple continuous auto-encoder [9], not a variational auto-encoder (VAE) [12]. It compresses the patch tokens from the DINOv3 [16] ViT-B [5] VFM, which uses a patch size of 16×16. For simplicity, both the tokenizer encoder and decoder follow the ViT-B configuration, though the formulation places no restrictions on scaling. They reuse the DINOv3 Transformer block implementation from Hugging Face Transformers [19], including 2D RoPE for spatial position encoding, but skip the patch embedding layer because the tokenizer operates on VFM output patch tokens rather than pixels. Two learned frame embeddings are added to the encoder inputs to distinguish previous-frame tokens from current-frame tokens. All linear and embedding weights are initialized with truncated normal ($\sigma{=}0.02$), linear biases are set to zero, and Layer Scale [18] values are initialized to 10^{-5}. In the tokenizer decoder, we omit the final layer normalization so that the small initial Layer Scale values make the decoder behave approximately as an identity map at initialization.

We train the tokenizer on sampled frame pairs for 50K iterations with a mean squared error (MSE) loss, using AdamW [13] with linear warmup to 10^{-3} over 5K steps and a constant learning rate thereafter, weight decay of 10^{-4}, a batch size of 1024, and gradient clipping at 10^{-2}.

	Num. samples	Duration (s)	FPS
DINO-world [2]	∼66M	5–60	10–60
Ours	∼4M	11	16

Table A. **Training data statistics.** For DINO-world [2], we report the duration range and FPS from their paper. For ours, we report the mean duration and all videos have a fixed FPS.

DeltaWorld predictor. For simplicity, the future predictor also uses the ViT-B configuration, though the formulation places no restrictions on scaling. Because each frame is represented by a single token rather than an $H \times W$ grid, neither the block-causal attention mask nor the three-dimensional RoPE used in DINO-world is needed. We therefore simplify the block-causal mask to a standard causal (diagonal) mask, and the 3D RoPE to a 1D variant that again rotates the first 60 dimensions of each head and leaves the final 4 unrotated. All predictors are trained with AdamW [13], weight decay $4{\times}10^{-1}$, and smooth L1 loss with $\beta{=}0.1$.

Training augmentations. We use random resized crops with a scale range of 0.6–1.0 and an aspect-ratio range of 3:4–4:3 applied to the original frames. The resulting crop coordinates are applied consistently to all frames in the sequence, and the crop is then resized to a square, introducing a small amount of aspect-ratio distortion.

Training data statistics. Similar to the experimental setting of DINO-world [2], all models (DINO-world, DeltaTok, and DeltaWorld) are trained on a large collection of videos spanning diverse domains. The training data used for DINO-world is not publicly released; Table A compares ours with what is reported in DINO-world. Our dataset comprises videos at mostly $640{\times}360$ resolution, spanning a wide range of scenarios similar in spirit to the DINO-world corpus.

Task heads. Following DINO-world [2], linear segmentation and depth heads are trained on frozen VFM features from the training split of each evaluation dataset. For segmentation on VSPW [14] and Cityscapes [4], the head uses a batch normalization layer followed by a linear layer projecting to 124 and 19 semantic classes, respectively. For depth estimation on KITTI [8], we adopt the DINOv3 [16] depth head architecture. Specifically, a batch normalization layer and a linear layer produce 256 logits per pixel. These logits are rectified and shifted by $\epsilon = 0.1$, normalized across the 256 bins to form a discrete depth distribution, and then

mapped to a continuous depth by taking the expectation over 256 uniformly spaced bins between 10^{-3} and 80 m. We evaluate on the Eigen split [6], cropping frames and depth maps to 352×1216 and restricting depth evaluation to valid pixels within the Garg region [7].

B. Additional Evaluation Details

Sequences. We extract evaluation sequences following prior work [2]. We use a time stride of 0.2 s for VSPW [14] and KITTI [8], and 0.1875 s for Cityscapes [4]. For VSPW, we select every 20th frame for evaluation and extract non-overlapping subsequences to keep the total number of sequences manageable.

Input preprocessing. Training uses square inputs, while evaluation datasets contain rectangular images. Therefore, during evaluation, frames are resized so that the shorter side matches the input size used in each experiment (512 in the main setting and 256 in the ablation setting). For KITTI, the Eigen crop (see above) is applied before resizing and the aspect ratio is clamped to 1:2. Labels remain at their original resolution. We then take two potentially overlapping left/right square crops from the resized frames. The labels are split horizontally into two non-overlapping halves, which define the regions used for evaluation. Task predictions from each frame crop are upsampled and cropped to match the corresponding label half.

Cosmos. Cosmos (Predict1) [1] can only be evaluated under its native inference constraints, and we follow a similar protocol to DINO-world [2]. Specifically, Cosmos requires a fixed context of 9 input frames and generates a rollout of 24 future frames in a single forward pass. Frames are resized so that the height is 512 pixels while preserving the aspect ratio, and padded to 640×1024 as required by the Cosmos input format. For KITTI, the Eigen crop (see above) is applied before resizing to 512×1024, which squashes the aspect ratio to 1:2; for all other datasets, no cropping is applied before generation. After generation, we remove the padding and apply the same left/right cropping protocol as above before re-encoding each predicted crop with DINOv3, ensuring consistent evaluation with other models.

***Best* and *mean* evaluation.** We generate 20 independent rollouts per sequence, unless noted otherwise. The *best* score is computed on the rollout whose DINOv3 features have the lowest feature-space loss to the ground truth at the last predicted timestep. The *mean* score averages the 20 DINOv3 features at the last predicted timestep and then applies the task head once on the averaged features; we do not take the average of the scores from the 20 future predictions. This evaluation protocol is applied per crop, identically to DeltaWorld and Cosmos. For the discriminative DINO-world baseline, we report the score of its single deterministic prediction.

Discriminative DINO-world [2]	
Backbone (4 frames)	$4 \times 47.185 = 188.74$
Predictor (4-frame context)	84.88
Predictor (5-frame context)	96.96
Predictor (6-frame context)	109.04
Generative DeltaWorld (Ours)	
Shared once	
Backbone (4 frames)	$4 \times 47.185 = 188.74$
DeltaTok encoder (4 frames)	$4 \times 96.930 = 387.72$
Per sample (repeated K times)	
Predictor (4-frame context)	0.26
Predictor (5-frame context)	0.28
Predictor (6-frame context)	0.31
DeltaTok decoder (step 1)	46.12
DeltaTok decoder (step 2)	46.12
DeltaTok decoder (step 3)	46.12

Table B. **GFLOPs breakdown.** In DeltaWorld, the backbone and DeltaTok encoder run once, while the predictor and DeltaTok decoder are applied per generated sample. Using a three-step rollout and a four-frame context (mid-horizon), ViT-B components, and 256×256 crops.

FLOPs. All GFLOPs are computed for square inputs and doubled, since evaluation uses two square-crop forward passes as described above. Cosmos is the exception, as it does not use square crops. Additionally, for Cosmos we exclude the fixed-cost GFLOPs associated with the tokenizer and KV pre-filling, which we expect to be small relative to the autoregressive decoding and iterative diffusion. For step (2) in Table 2, GFLOPs include applying the tokenizer decoder at each intermediate rollout step, not only the final one.

Training time and memory. In Table 2, we measure the training time per optimization iteration and steady-state GPU memory on a single node with 8 NVIDIA H200 GPUs, using BF16 mixed precision and `torch.compile` (default mode). Despite generating $K=16$ candidate futures, BoM training in step (1) requires similar memory to the discriminative baseline, because the candidate selection pass uses detached parameters (no activation storage for backpropagation) and only the best candidate is re-run with gradients. Delta compression in step (3) is slightly slower than frame compression in step (2) because its tokenizer encoder processes both the current and previous frame's patch tokens.

Model	Time	Mem	VSPW ↑	Cityscapes ↑
Copy last (lower bound)	–	–	*41.8*	*37.9*
DINO-world† [2]	1.0×	1.0×	**44.8**	45.4
↳ **Delta compression**	**0.5**×	**0.2**×	44.6	**46.9**
Present (upper bound)	–	–	*52.0*	*59.3*

Table C. **Delta tokens in the discriminative DINO-world [2].** *Delta tokens* also perform well within a discriminative world model. Time and Mem report per-iteration training time and GPU memory relative to the discriminative baseline. Reporting mid-horizon ($\sim$0.6 s) mIoU using 256 × 256 crops. †Our reimplementation.

		Seg. mIoU ↑		Depth δ_1 ↑	
	Tokens	Short	Mid	Short	Mid
Copy last (lower bound)		*54.7*	*40.4*	*84.1*	*77.8*
DINO-Foresight† [11]	10240	71.8	59.8	**88.6**	85.4
↳ **Delta compression**	**5**	**72.1**	**60.0**	88.5	**85.6**
Present (upper bound)		*77.0*	*77.0*	*89.1*	*89.1*

Table D. **Delta tokens in the discriminative DINO-Foresight [11].** Results on Cityscapes [4] show that *delta tokens* transfer effectively to a different discriminative architecture, matching performance with 2048× fewer tokens. The token count indicates the total number of tokens used by the world model. Using 448×896 frames. †Numbers reported in the DINO-Foresight paper [11].

Efficiency breakdown. Table B shows how GFLOPs are distributed across the model components for both the discriminative DINO-world [2] and our generative DeltaWorld. Although the predictor dominates compute in DINO-world, its cost becomes negligible in DeltaWorld with a short context of four to six delta tokens, with most per-sample compute instead coming from the DeltaTok decoder. Crucially, however, unlike the predictor in DINO-world, the decoder's compute cost does not increase with context length. Even with the small predictor size and the benchmark's short context length, the decoder remains more efficient than the predictor in DINO-world. Furthermore, the DeltaTok encoder overhead in DeltaWorld is shared across all generated samples. This makes DeltaWorld noticeably cheaper per generated sample and enables efficient multi-sample generation, while the future predictor remains lightweight and flexible for scaling, *e.g.*, in context or predictor size.

C. Delta Tokens in Discriminative Models

Although not the primary focus of this paper, *delta tokens* can also be used in a discriminative world model. Table C shows that replacing per-frame patch tokens with a single delta token in the discriminative DINO-world baseline [2] performs well (-0.2 on VSPW and +1.5 on Cityscapes), while also being more efficient in training time (0.5×) and memory (0.2×).

We also integrate delta tokens into DINO-Foresight [11], a separate discriminative world model with a different architecture, using their official implementation. It is trained and evaluated on Cityscapes [4] and extracts multi-layer DINOv2 [15] features, applying PCA to obtain 1152-dimensional spatial features per patch. We train a DeltaTok variant that compresses these PCA features of two consecutive frames into a single 1152-dimensional delta token at 448×896 resolution, using BDD100K [20] and briefly fine-tuning on Cityscapes [4]. We then retrain the DINO-Foresight world model on Cityscapes to predict these delta tokens instead of spatial PCA features. Since delta tokens collapse the large spatio-temporal sequence to only one token per frame, we can simplify the architecture by replacing the factorized space-time attention with standard self-attention, and skip the high-resolution fine-tuning stage, training directly at the target resolution. As shown in Table D, the delta-compressed variant matches the original while reducing the token count by 2048×, confirming that delta tokens transfer effectively across discriminative world model architectures.

D. Limitations and Future Work

We discuss two limitations of our work and directions for future research.

Distribution modeling. The Best-of-Many (BoM) objective enables efficient, non-iterative generation of diverse futures by mapping stochastic noise queries to distinct futures [3]. However, unlike diffusion models [10], whose denoising objective provides a principled connection to the data distribution, BoM lacks an explicit distributional objective. Consequently, the model's coverage of the predictive distribution is limited by the number of noise queries K explored during training, with no mechanism encouraging diverse utilization of the query space, and no guarantee that the distribution over sampled futures approximates the true probability of each outcome. That said, in practice different queries tend to produce distinct futures, suggesting the query space may serve as a form of implicit action conditioning. This could open a path toward explicit action-conditional generation, as similar queries may produce similar futures across different scenes.

Error accumulation. Because delta tokens encode temporal differences, reconstructing absolute feature maps requires repeatedly decoding delta tokens conditioned on previous features. During tokenizer reconstruction, errors may compound across steps, potentially leading to feature drift. A natural mitigation is to have the tokenizer operate on its own reconstructions, computing delta tokens sequentially relative to previously decoded frames, rather than in parallel from ground truth input frames. In DeltaWorld, the predic-

tor may introduce an additional source of error, which may further compound during multi-step autoregressive rollouts, a well-known challenge in autoregressive video generation. Existing approaches to mitigate error accumulation in autoregressive generation may apply.

E. Additional Qualitative Examples

In Figure A we show short-horizon Cityscapes [4] predictions from DINO-world [2], Cosmos-12B [1], and DeltaWorld. All three models predict the car moving out of the frame, but both DINO-world and Cosmos fail to maintain the bicycle wheel, DINO-world also loses the sign post, and Cosmos misses some of the people in the background.

In Figure B we show mid-horizon KITTI [8] predictions, comparing *mean* and *best* samples for Cosmos-12B and DeltaWorld. Both models track the car's motion, but DeltaWorld's *best* sample is more accurate than Cosmos's: for example, it provides a more accurate depth estimate on the passing train. Cosmos also yields *mean* and *best* samples that are very similar, reflecting lower variation across its outputs.

In Figures C and D we show mid-horizon autoregressive rollouts from DeltaWorld across all three evaluation datasets, visualized through task head outputs (Figure C) and RGB reconstructions (Figure D). Since DeltaWorld operates in DINOv3 feature space, we use the decoder from Representation Autoencoder (RAE) [21], trained on DINOv3 ViT-B, to decode predicted features back into pixels for the RGB visualization.

In Figures E and F we visualize the diversity of mid-horizon autoregressive rollouts from DeltaWorld across all three evaluation datasets by showing multiple samples for the same input context, again as task head outputs (Figure E) and RGB reconstructions (Figure F). Each group of three rows shares the same four context frames but shows three different rollouts.

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos World Foundation Model Platform for Physical AI. *arXiv preprint arXiv:2501.03575*, 2025. 2, 4, 6, 7

[2] Federico Baldassarre, Marc Szafraniec, Basile Terver, Vasil Khalidov, Francisco Massa, Yann LeCun, Patrick Labatut, Maximilian Seitzer, and Piotr Bojanowski. Back to the Features: DINO as a Foundation for Video World Models. In *ICML*, 2025. 1, 2, 3, 4, 6

[3] Apratim Bhattacharyya, Bernt Schiele, and Mario Fritz. Accurate and Diverse Sampling of Sequences Based on a "Best of Many" Sample Objective. In *CVPR*, 2018. 3

[4] 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. 1, 2, 3, 4, 6, 8, 10

[5] Alexey Dosovitskiy. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021. 1

[6] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014. 2

[7] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 2

[8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets Robotics: The KITTI Dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013. 1, 2, 4, 7, 8, 10

[9] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *science*, 313(5786):504–507, 2006. 1

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *NeurIPS*, 2020. 3

[11] Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. DINO-Foresight: Looking into the Future with DINO. In *NeurIPS*, 2025. 3

[12] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2014. 1

[13] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019. 1

[14] Jiaxu Miao, Yunchao Wei, Yu Wu, Chen Liang, Guangrui Li, and Yi Yang. VSPW: A Large-scale Dataset for Video Scene Parsing in the Wild. In *CVPR*, 2021. 1, 2, 8, 10

[15] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning Robust Visual Features without Supervision. *TMLR*, 2024. 3

[16] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. 1

[17] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024. 1

[18] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going Deeper with Image Transformers. In *ICCV*, 2021. 1

[19] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-Art Natural Language Processing. In *EMNLP Demos*, 2020. 1

[20] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *CVPR*, 2020. 3

[21] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion Transformers with Representation Autoencoders. In *ICLR*, 2026. 4

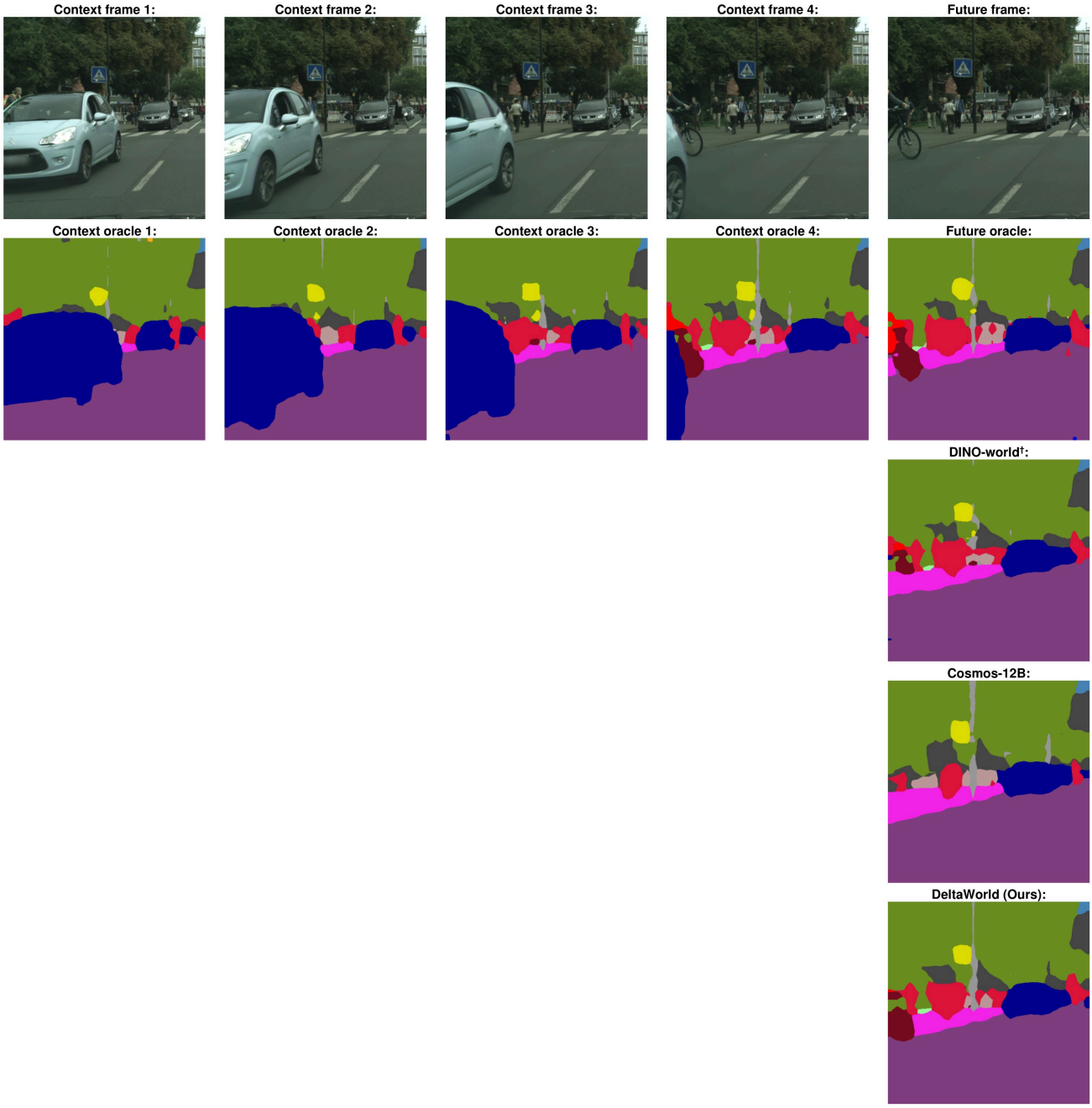

Figure A. **DINO-world† [2] *vs*. Cosmos-12B [1] *vs*. DeltaWorld (Ours).** Given a context of four frames, predict the fifth frame (short-horizon). Second row shows the segmentation head output on the ground-truth frames, while third, fourth, and fifth rows show the segmentation head output for the predicted future frame. In this Cityscapes example [4], DeltaWorld provides an accurate prediction of the scene evolution. Generative models show *mean* features from 20 samples; DINO-world shows its single deterministic prediction. Using 512×512 crops. †Our reimplementation.

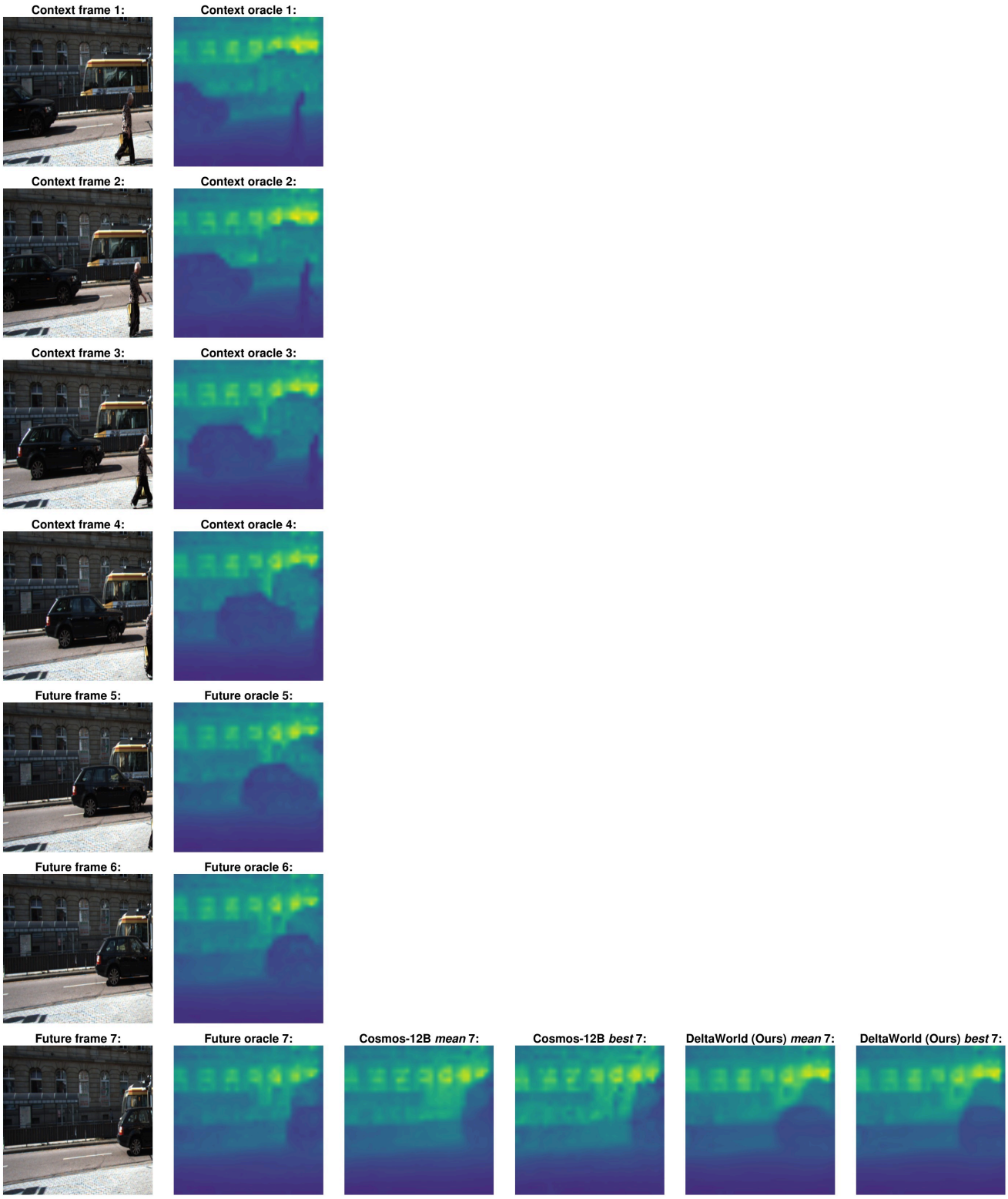

Figure B. **Comparing *mean* and *best* for Cosmos-12B [1] *vs*. DeltaWorld (Ours).** Given a context of four frames, predict the seventh frame autoregressively (mid-horizon). Second column shows the depth head output on the ground-truth frames, third and fourth columns show Cosmos, and fifth and sixth columns show DeltaWorld predictions. In this KITTI example [8], DeltaWorld's *best* sample more closely matches the oracle depth layout. Using 512×512 crops.

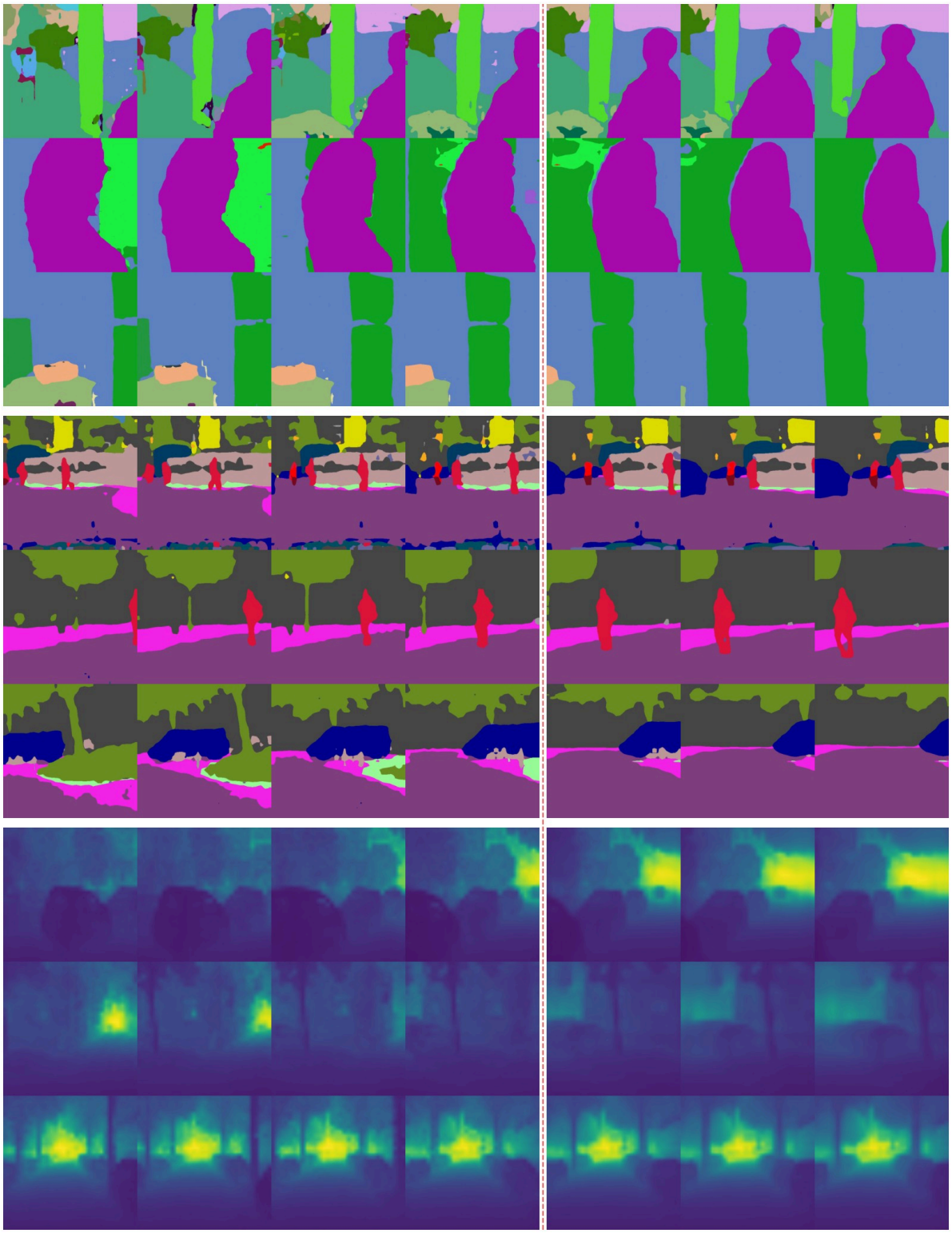

Figure C. **Mid-horizon rollouts (task head visualization).** Each row shows four context frames (left of the dashed line) and an autoregressive rollout from DeltaWorld (right), conditioned on random noise queries, in a single forward pass per step. Top: VSPW [14] segmentation, middle: Cityscapes [4] segmentation, bottom: KITTI [8] depth. Using 512×512 crops.

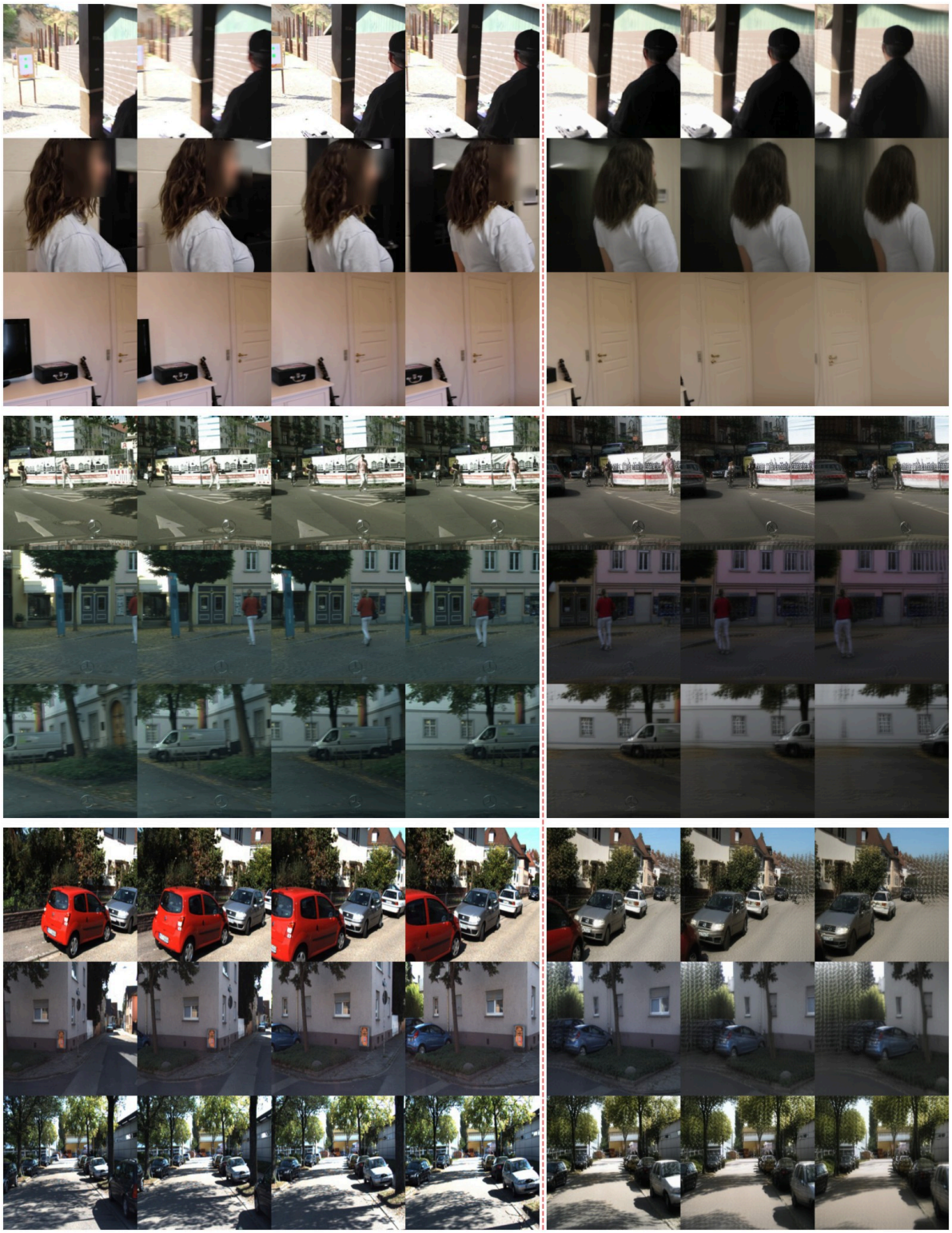

Figure D. **Mid-horizon rollouts (RGB visualization).** Same sequences as Figure C. Context columns (left of the dashed line) show ground-truth RGB frames; future columns show the predicted features decoded into pixels. Using 512×512 crops.

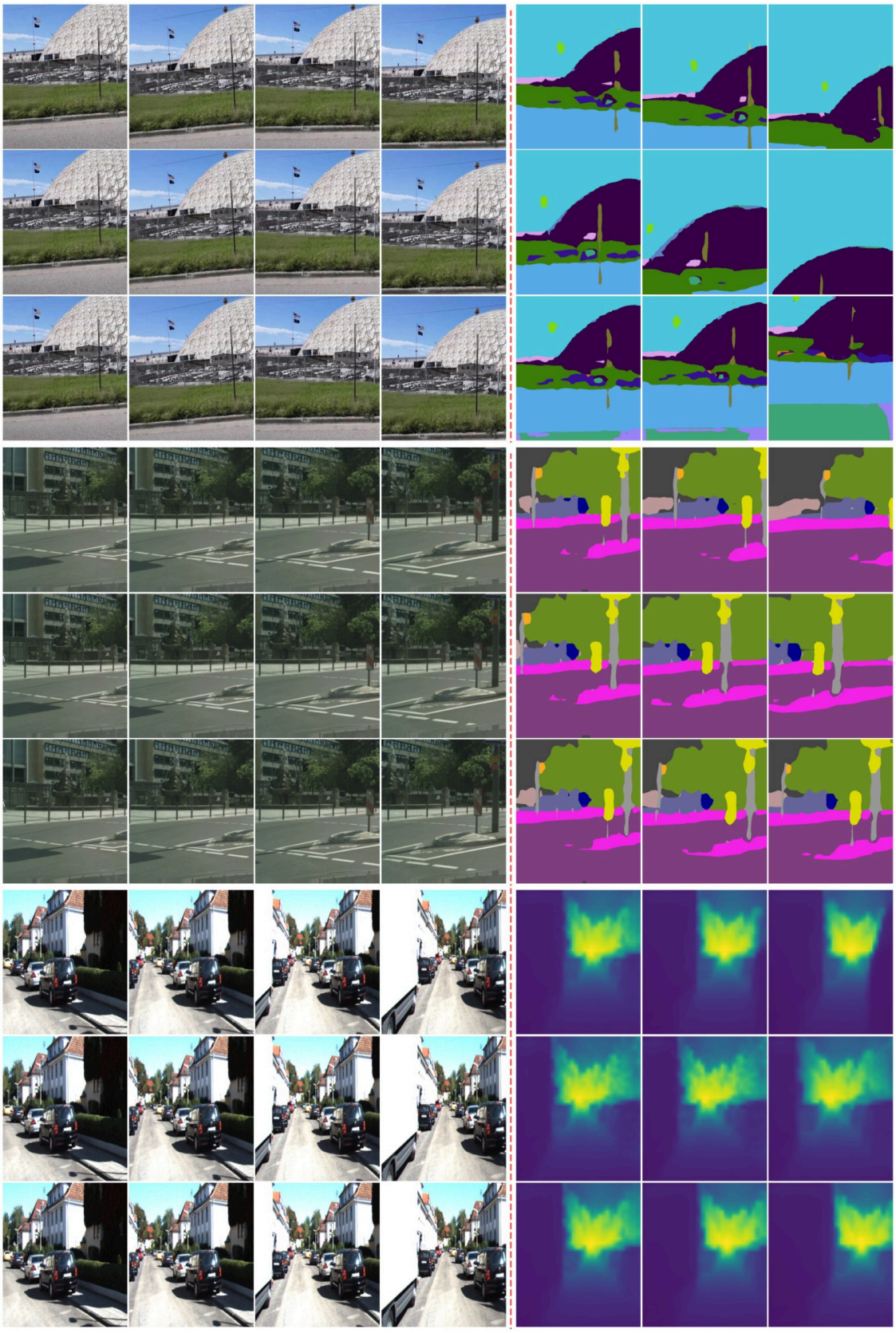

Figure E. **Diverse mid-horizon rollouts (task head visualization).** Each group of three rows shares the same four context frames (left of the dashed line) but shows three different autoregressive rollouts from DeltaWorld, each conditioned on random noise queries, in a single forward pass per step. Top: VSPW [14] segmentation, middle: Cityscapes [4] segmentation, bottom: KITTI [8] depth. Using 512×512 crops.

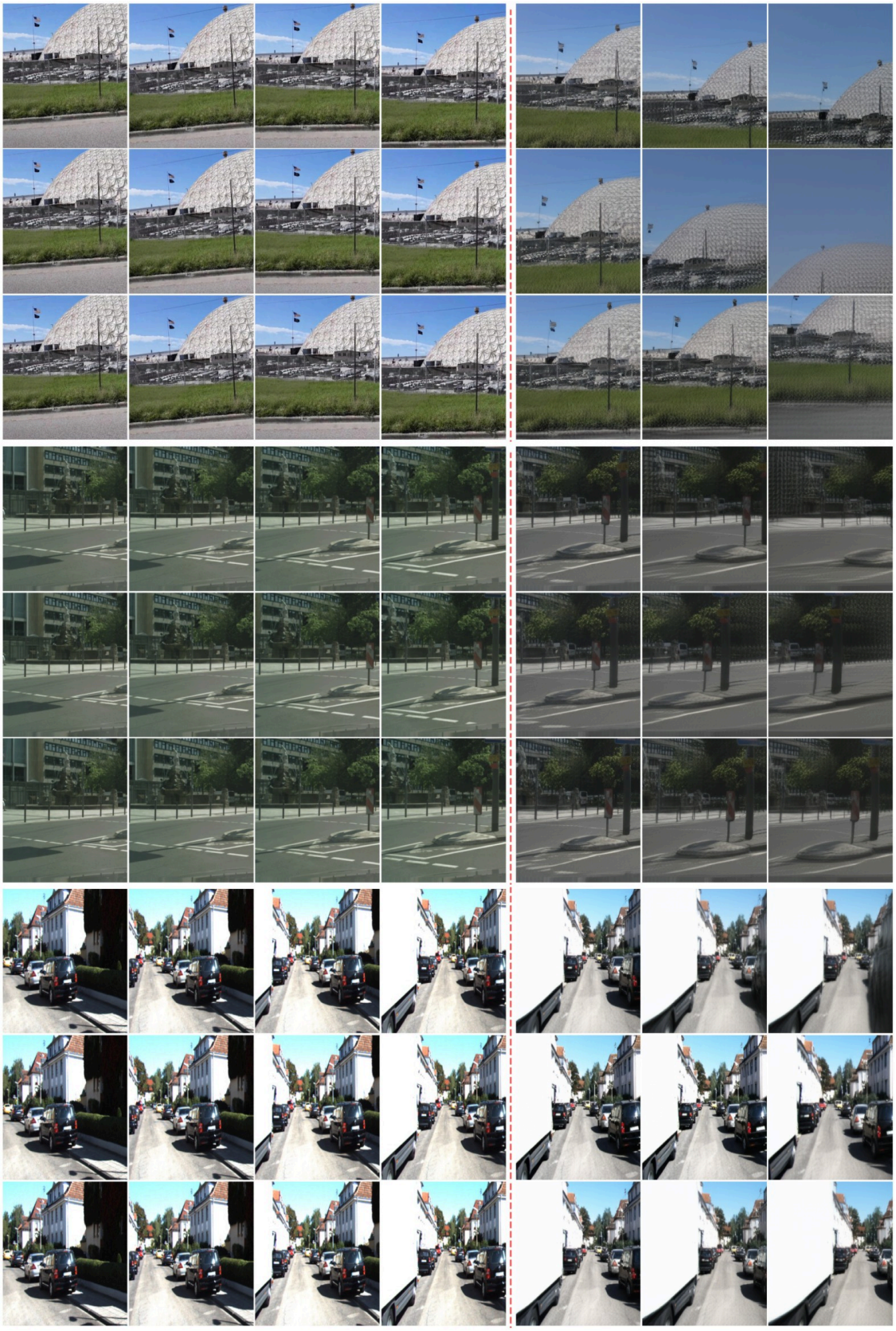

Figure F. **Diverse mid-horizon rollouts (RGB visualization).** Same sequences and samples as Figure E. Context columns (left of the dashed line) show ground-truth RGB frames; future columns show the predicted features decoded into pixels. Using 512×512 crops.