GENERATIVE MODELING BEYOND PIXELS WITH ANY PRETRAINED VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many pretrained transformers include task heads that decode their features into structured outputs. VGGT and Depth Anything V2 (DAv2), for example, produce depth maps, while VGGT also produces camera poses and point clouds. We show that the feature space of such an encoder can be modeled directly. This turns the encoder into a text-conditioned generative model whose samples are decoded by the heads it already has. Our framework, MIRAGE, compresses encoder tokens with a hybrid VAE onto a product of spheres with log-normal radii, and trains a diffusion transformer on these latents using rectified flow that follows geodesics on each sphere and a straight path in log-radius. Because generation targets the encoder's own feature space, REPA alignment can use the encoder itself as its target, and no external representation model is needed. For long videos, a temporal compressor and per-frame flow times let a single fixed-window model either generate a clip in parallel or roll forward on clean past context, producing videos of arbitrary length. We evaluate four geometric encoders: VGGT, PAGE-4D, Depth Anything V2, DUSt3R, and provide initial audio experiments. A single model generates all outputs supported by each encoder's heads. Against raw-feature diffusion baselines at two scales, MIRAGE's latents are 16-64$\times$ smaller, remains within 1-6% in depth-FVD, lowers depth-gradient divergences across all tested encoders, and samples 2$\times$ faster with 4-13$\times$ less sampling memory beyond model weights. Its compact latent also makes rolling generation of long videos practical.

1 INTRODUCTION

Pretrained transformers have become general-purpose representation models, mapping raw observations into feature spaces that support a range of downstream predictions (Oquab et al., 2023). In many architectures, downstream task heads operate on these representations to decode them directly into structured outputs. VGGT, for example, predicts depth, camera parameters, point maps, and point tracks from a shared transformer representation. This suggests treating the learned feature space itself as the domain of generation. Diffusion and flow models transform noise into data (Liu et al., 2022; Lipman et al., 2022), while latent diffusion performs this transformation in a learned autoencoder latent before decoding the result back to the original signal (Rombach et al., 2022). Recent work has brought pretrained representations closer to the generative process. REPA aligns DiT features with pretrained semantic representations (Yu et al., 2024), while RAEs use pretrained encoders as generative latents (Zheng et al., 2026). However, in the case of REPA, the pretrained representation is simply an intermediate way to help the model learn to produce the final observation, while RAE is restricted to the case of image generation on ImageNet. Follow-up works have primarily focused on image or video generation.

Pursuing this alternative, we propose MIRAGE. Given a pretrained transformer whose feature space already supports one or more frozen task heads, MIRAGE directly models the distribution of those features. Generated representations can then be passed to the same heads used for real inputs. A single generative model can therefore produce, for example, depth, camera parameters, and dense geometry through the corresponding pretrained heads, without requiring a separate generator for each output. More generally, MIRAGE inherits its output space from the pretrained representation model on which it is built. The only requirement is that the heads process the transformer's features. Making this work requires a latent space that both preserves pretrained features and supports effective generative modeling. Since the geometry of the features is not necessarily well described by

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

the Euclidean latent spaces assumed by standard diffusion models, we compress encoder tokens into a hybrid product-of-spheres latent representation. Respecting this geometry during generation, we apply rectified flow along spherical geodesics for the directional components and Euclidean flow in log-radius (Lipman et al., 2022; Liu et al., 2022; Chen & Lipman, 2024). To keep generated features compatible with the frozen heads, we use *self-representation alignment*, a variant of REPA (Yu et al., 2024) in which the target encoder itself provides the alignment signal.

For long sequences, we further extend the framework beyond the DiT's fixed training window. A temporal compressor expands the effective temporal context, while per-frame flow-time modulation, following the Diffusion Forcing formulation (Chen et al., 2024), allows the same DiT to operate in both parallel and rolling modes. During rolling generation, previously generated context is held at or near the clean end of the flow while a new noisy slot is denoised through attention to that context, avoiding explicit chunk-boundary autoregression.

We evaluate MIRAGE on four pretrained geometric encoders, VGGT (Wang et al., 2025), PAGE-4D (Zhou et al., 2026), Depth Anything V2 (Yang et al., 2024), and DUSt3R (Wang et al., 2024), so our experiments cover heads that produce geometry: depth, camera parameters, and point maps. We also provide preliminary results on audio in Appendix H, showing that it generalizes beyond just vision. Across these different feature spaces, MIRAGE generates samples that the corresponding frozen heads decode directly. We construct baselines which generate the features, as RAEs do for images (Zheng et al., 2026), and we train it as a baseline on PAGE-4D, Depth Anything V2, and DUSt3R. On these encoders, MIRAGE generates in a latent space 16-64 times smaller than the features and stays within 1-6 percent of this baseline in depth-FVD, a difference that a paired bootstrap over test clips finds significant in only one of five comparisons. Its depth-gradient KL is 1.7-5.3 times lower on each encoder, with significant local-structure gains over both baselines on PAGE-4D and Depth Anything V2. It also samples twice as fast with 4-13 times less memory, while its rolling formulation supports generation beyond the native training window.

Our main contributions are threefold: (1) we propose MIRAGE, a framework for generative modeling directly in the task-aware representation space of a pretrained transformer, with generated features interpreted by its frozen task heads; (2) we use a geometry-aware generative formulation combining a hybrid product-of-spheres latent representation, manifold-respecting rectified flow over directional and radial components, and self-representation alignment to the encoder being modeled; and (3) we incorporate a long-form video extension through temporal compression and per-frame flow-time modulation, enabling rolling generation beyond the DiT's fixed training window.

2 Related Work

Latent diffusion. Latent diffusion (Rombach et al., 2022) trains the generator in the latent space of a pretrained autoencoder, and rectified flow and flow matching (Liu et al., 2022; Albergo et al., 2025; Lipman et al., 2022) regress the velocity of an interpolant between noise and data, as in large transformer models such as Stable Diffusion 3 (Esser et al., 2024). We keep the rectified-flow objective, but our VAE compresses the features of a pretrained encoder E instead of pixels, so its latent retains the task-relevant structure that E was trained to expose.

Generation in semantic feature spaces. RAEs (Zheng et al., 2026) train a diffusion transformer (DiT) directly on the features of a frozen encoder such as DINOv2 and learn a decoder back to pixels, and REPA (Yu et al., 2024) aligns diffusion features with an external encoder. MIRAGE differs in three ways. The encoders we model come with task heads, so every sample can be decoded into each output the encoder supports without training an inverse model. It generates in a compact latent, 16 to 64 times smaller than the raw features, instead of in the features themselves. Its alignment target is the encoder being modeled and not an external one (§3.6).

Flow matching on manifolds. Riemannian flow matching (Chen & Lipman, 2024) uses geodesic interpolants and regresses the network onto the geodesic velocity in the tangent space. Hyperspherical VAEs with Power Spherical posteriors (Cao & Aziz, 2020) give a well-behaved uniform prior. Our latent space (§3.2) combines them. Each direction is generated with Riemannian flow on its sphere and each radius with Euclidean flow in log-radius, so no step of training or sampling leaves the product manifold.

Long video generation. Autoregressive rollout and sequential chunk sampling accumulate errors at chunk boundaries. Diffusion Forcing (Chen et al., 2024) trains with an independent noise level per

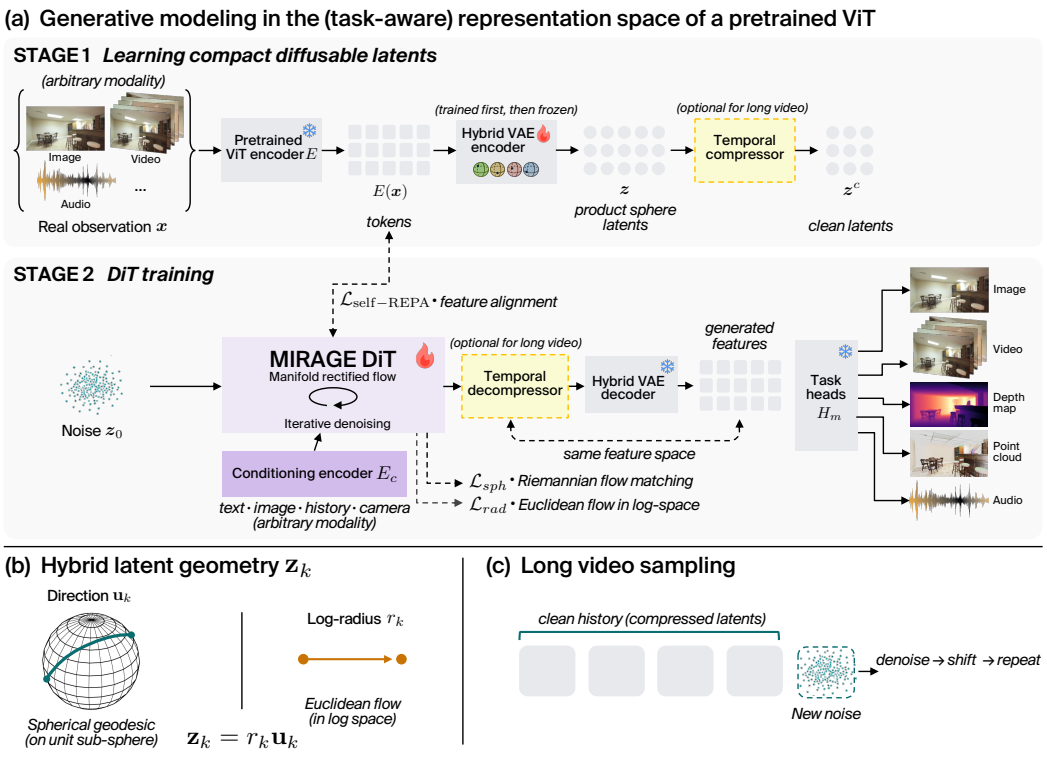


Figure 1: Overview of MIRAGE. **Top:** a frozen pretrained transformer E maps each input element to tokens $E(\boldsymbol{x}_n)$, and the frozen hybrid VAE encodes them into latents $\boldsymbol{z}$ on the product-of-spheres manifold $\mathcal{M}$. For long sequences, an optional frozen temporal compressor shortens the latent sequence. **Bottom:** the DiT generates latents with a spherical flow on the directions and a Euclidean flow in log-radius, conditioned on the encoders $\{E_c\}$ and aligned to the features of E by self-REPA. Generated latents pass through the optional decompressor and the VAE decoder back into the token space of E, where any frozen task head H_m reads them to produce its output.

frame, so that one model can denoise a whole window in parallel or extend a sequence from clean past frames. DFoT (Song et al., 2025) applies this to video diffusion transformers and conditions on a variable amount of clean history. We use the same per-frame noise levels in a feature space and add a temporal compressor, so that the fixed DiT window covers four times as many raw frames.

Pretrained transformers with task heads. VGGT (Wang et al., 2025) predicts depth, point maps, and camera pose from a set of frames, and DINOv2 (Oquab et al., 2023) supports linear heads for segmentation and depth. MIRAGE assumes only a shared interface of token-emitting encoders and task heads, making the encoder a framework parameter rather than a design choice.

3 Methodology

MIRAGE turns a pretrained transformer with task heads into a generative model whose samples live in the transformer's own feature space. We compress the transformer's features into a latent space shaped to their geometry and train a diffusion model in that space. When the encoder E has task heads $\{H_m\}$, a single generated sample decodes into every modality the heads produce, and all of these outputs are computed from the same generated features. In addition, if the heads were trained jointly and are consistent, then the outputs of MIRAGE will be consistent too. The framework relies only on E and $\{H_m\}$: E maps inputs to token sequences, and each H_m maps a token sequence to a task output. The same recipe applies whether the model generates a single element ($N = 1$), a short ordered sequence such as video frames or views of a scene that fits in one DiT forward pass ($1 < N \leq N_{\max}$), or a longer sequence, which uses the extension of §4. Fig. 1 summarizes the overview of MIRAGE.

3.1 Framework and Notation

Let $E : \mathcal{X} \to \mathbb{R}^{L \times D_E}$ be a frozen pretrained transformer that maps an input element $\boldsymbol{x} \in \mathcal{X}$ to L tokens of dimension D_E. When features are taken from F layers of E, they are concatenated per token, so that $D_E = FD_\ell$ with D_ℓ the width of one layer. Let $\{H_m\}_{m=1}^M$ be the frozen task heads of E, each of which decodes tokens into one output modality such as depth or RGB. The DiT also reserves one summary token per element. For VGGT and PAGE-4D it carries the latent of the encoder's camera token, a Euclidean vector generated jointly with the patch latents. For the other encoders it holds a single extra token.

Given $N \geq 1$ elements $\boldsymbol{x}_{1:N}$ (of the given input type), we (i) apply E to each element, (ii) encode each token sequence with the hybrid VAE into latents on a manifold $\mathcal{M}$ (§3.2), and (iii) train a velocity network v_θ to generate the joint latent sequence $\boldsymbol{z}_{1:N}$ with rectified flow on $\mathcal{M}$ (§3.3), conditioned on an ordered set of conditioning encoders $\{E_c\}$ (§3.5). At inference, the VAE decodes generated latents into the token space of E, and the heads $\{H_m\}$ turn them into outputs.

3.2 Product-of-Spheres Latent Representation and Hybrid VAEs

The raw tokens of E usually have very large dimensionality, which makes diffusing them directly expensive (Table 2). Bond et al. (2026) introduced S^{2}VAE, which compresses transformer features onto a product of unit spheres with Power Spherical posteriors (Cao & Aziz, 2020) and reconstructs them better than a Gaussian VAE. A product of unit spheres discards the norm of each sub-vector, however, so we restore a controlled amount of magnitude information with a radius per sphere. Each D-dimensional latent token therefore lives on the manifold

$$\mathcal{M} = \prod_{k=1}^{K} (\mathbb{S}^{d-1} \times \mathbb{R}_{>0}), \quad D = K \cdot d, \quad \boldsymbol{z} = \bigoplus_{k=1}^{K} r_k \boldsymbol{u}_k, \quad \boldsymbol{u}_k \in \mathbb{S}^{d-1},\ r_k \in \mathbb{R}_{>0}. \tag{1}$$

where $\mathbb{S}^{d-1}$ is the unit sphere in $\mathbb{R}^d$, and a latent $\boldsymbol{z} \in \mathcal{M}$ decomposes into K sub-latents, each a unit direction scaled by a radius. The radius reintroduces magnitude without returning to an unconstrained Gaussian latent, which Bond et al. (2026) found performs worse than the product of spheres.

The VAE encoder outputs a Power Spherical distribution for each direction and a log-normal distribution for each radius. Both have closed-form KL divergences to their priors, the uniform distribution on the sphere and a standard normal on $\log r_k$. The radius KL is weighted at a small fraction of the direction KL, so that magnitude information survives the bottleneck (App. A). We call this model the hybrid VAE. It is trained on the features of E, with a feature reconstruction loss and the auxiliary terms listed in Table A4, and then frozen for all diffusion training.

3.3 Rectified Flow on a Product-of-Spheres Manifold

Standard rectified flow (Liu et al., 2022; Lipman et al., 2022; Albergo et al., 2025) draws noise $\boldsymbol{z}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and data $\boldsymbol{z}_1 \sim p_{\text{data}}$, forms the interpolant $\boldsymbol{z}_t = (1-t)\boldsymbol{z}_0 + t\boldsymbol{z}_1$, and trains a velocity network with the loss $\mathbb{E}_{t,\boldsymbol{z}_0,\boldsymbol{z}_1}[\|v_\theta(\boldsymbol{z}_t, t) - (\boldsymbol{z}_1 - \boldsymbol{z}_0)\|^2]$. This interpolant ignores the structure of $\mathcal{M}$. The straight line between two points on a sphere passes through its interior, so every intermediate latent leaves the manifold that the VAE was built to enforce. We thus split the flow into a directional part and a radial part, each following the geometry of its component. In both parts the source distribution matches the VAE prior, uniform on each sphere and standard normal in log-radius.

Directions: Riemannian flow matching. Each sub-latent has a direction $\boldsymbol{u}_k \in \mathbb{S}^{d-1}$. We interpolate between the noise and data directions along the great circle that connects them (SLERP),

$$\boldsymbol{u}_k(t) = \frac{\sin((1-t)\Omega_k)}{\sin \Omega_k}\, \boldsymbol{u}_{k,0} + \frac{\sin(t\Omega_k)}{\sin \Omega_k}\, \boldsymbol{u}_{k,1}, \quad \Omega_k = \arccos(\boldsymbol{u}_{k,0}^\top \boldsymbol{u}_{k,1}), \tag{2}$$

which stays on $\mathbb{S}^{d-1}$ for all $t \in [0, 1]$. Its time derivative $\dot{\boldsymbol{u}}_k(t)$ lies in the tangent space at $\boldsymbol{u}_k(t)$. We project the network's predicted ambient vector for each sub-sphere onto this space using $\Pi_{\boldsymbol{u}}(\boldsymbol{w}) = \boldsymbol{w} - (\boldsymbol{w}^\top \boldsymbol{u})\boldsymbol{u}$ to obtain the directional velocity $\boldsymbol{v}_k^{\text{sph}}(\boldsymbol{z}_t, t, \boldsymbol{c})$. The directional loss is

$$\mathcal{L}_{\text{sph}} = \mathbb{E}_{t,\boldsymbol{u}_0,\boldsymbol{u}_1}\left[\sum_{k=1}^{K} \|\boldsymbol{v}_k^{\text{sph}}(\boldsymbol{z}_t, t, \boldsymbol{c}) - \dot{\boldsymbol{u}}_k(t)\|_2^2\right]. \tag{3}$$

At sampling time, each Euler step moves along the sphere with the exponential map, so the trajectory always stays on $\mathbb{S}^{d-1}$.

Radii: Euclidean flow in log-space. Each sub-latent also has a radius $r_k > 0$. A Gaussian source would place mass on negative radii, so we generate $\log r_k \in \mathbb{R}$ instead, where standard Euclidean rectified flow applies:

$$\log r_k(t) = (1-t)\log r_{k,0} + t \log r_{k,1}, \tag{4}$$

$$\mathcal{L}_{\text{rad}} = \mathbb{E}_{t,r_0,r_1}\left[\sum_{k=1}^{K}\left(v_k^{\text{rad}}(\boldsymbol{z}_t, t, \boldsymbol{c}) - (\log r_{k,1} - \log r_{k,0})\right)^2\right]. \tag{5}$$

At sampling time, we integrate $\log r_k$ with Euler steps and exponentiate the result. At each step, the network receives the current direction and log-radius of each sub-latent separately (App. B).

The flow loss is $\mathcal{L}_{\text{flow}} = \mathcal{L}_{\text{sph}} + \mathcal{L}_{\text{rad}}$, plus a Euclidean rectified-flow term for the summary latent. A single forward pass predicts all velocities from shared conditioning and a shared flow-time input. Training times are drawn uniformly from $[0, 1]$.

3.4 DiT Architecture

The velocity network v_θ is a diffusion transformer over the $N \cdot L$ latent tokens of a sequence. Each block factorizes attention into a spatial pass over the L tokens of each element and a temporal pass over the N elements at each token position. This reduces the cost of attention from $O(N^2L^2)$ to $O(NL^2 + LN^2)$. Spatial attention uses rotary position embeddings over the 2D patch grid, and a learned per-element position embedding added to every token encodes temporal order. A block applies spatial attention, temporal attention, cross-attention to the conditioning tokens, and a SwiGLU MLP in turn, and each of these is modulated by adaptive layer normalization whose input is the sum of the pooled conditioning and a per-element flow-time embedding.

The flow time is a vector with one entry per element rather than a scalar. During training, we draw a shared time t_{s} and independent per-element times $\tilde{t}_{1:N}$, and mix them with a curriculum coefficient α as $t_n = (1-\alpha)\, t_{\text{s}} + \alpha\, \tilde{t}_n$. Training begins at $\alpha = 0$, which is standard rectified flow with one time per sample, and α ramps to 1 over a warmup period. At $\alpha = 1$, each element has its own noise level as in Diffusion Forcing (Chen et al., 2024). This enables the rolling sampler described in §4.

3.5 Modality-Agnostic Multi-Stream Conditioning

Conditioning follows the multi-encoder design of Stable Diffusion 3 (Esser et al., 2024) and comes from an ordered list of encoders $\{E_c\}_{c=1}^{C}$. Encoder E_c maps its input $\boldsymbol{y}_c$ to tokens $E_c(\boldsymbol{y}_c) \in \mathbb{R}^{N_c \times D_c}$ and, optionally, to a pooled vector $\bar{E}_c(\boldsymbol{y}_c) \in \mathbb{R}^{D_c}$. The DiT learns a token projection W_c^{tok} and a pooled projection W_c^{pool} for each stream. Cross-attention uses the concatenation of projected token sequences $\boldsymbol{c}_{\text{tok}} = [W_1^{\text{tok}} E_1(\boldsymbol{y}_1), \ldots, W_C^{\text{tok}} E_C(\boldsymbol{y}_C)]$, and the summed pooled projections $\sum_c W_c^{\text{pool}} \bar{E}_c(\boldsymbol{y}_c)$ are added to the flow-time embedding that drives adaptive layer normalization.

Our default configuration uses three streams: CLIP-L and T5-large text encoders, and DINOv2-base encoding of the first frame as an image reference. Nothing in the architecture depends on the number or type of streams. Adding, removing, or replacing a modality only requires registering its projections.

3.6 Self-REPA Alignment

REPA (Yu et al., 2024) adds an auxiliary loss that aligns an intermediate representation of the DiT with the features of a pretrained encoder, which substantially speeds up convergence and improves sample quality. The alignment target is usually a strong external encoder such as DINOv2 (Oquab et al., 2023), chosen for the quality of its features on natural images.

Our setting offers a more direct target, the encoder E whose features the DiT is learning to generate. Let $\boldsymbol{h}_{n,i}^{(b)}$ be the hidden state of token i of element n after DiT block b, computed from the noisy input $\boldsymbol{z}_t$, and let g_ϕ be a three-layer MLP projection head with a bottleneck. The self-REPA loss is

the mean per-token cosine distance to layer j of E on the clean input,

$$\mathcal{L}_{\text{self-REPA}} = \mathbb{E}_{\boldsymbol{x},t}\left[1 - \frac{1}{NL}\sum_{n=1}^{N}\sum_{i=1}^{L}\cos\left(g_\phi(\boldsymbol{h}_{n,i}^{(b)}),\, E_{[j]}(\boldsymbol{x}_n)_i\right)\right], \tag{6}$$

where $E_{[j]}(\boldsymbol{x}_n)_i$ is token i of the output of layer j of E applied to $\boldsymbol{x}_n$. We use block $b = 5$ and the last layer of E. The full training objective is $\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda\,\mathcal{L}_{\text{self-REPA}}$, with $\lambda = 0.5$ in all experiments. The alignment target and the generation target then describe the same features, so the auxiliary loss pulls the DiT toward the representation it must ultimately produce. No external encoder is needed either, since E already computes these features at every step to produce the VAE inputs. §5.3 compares other alignment depths.

3.7 Guidance, Normalization, and Parameter Averaging

For classifier-free guidance, all conditioning streams are jointly dropped with probability 0.35 per sample, so guidance with scale s, $v_\theta^{\text{cfg}} = v_\theta^{\text{uncond}} + s(v_\theta^{\text{cond}} - v_\theta^{\text{uncond}})$, strengthens the conditioning as a whole. Direction, log-radius, and summary latents are standardized separately using running estimates of their means and variances at the DiT input and output to account for their different scales. We sample from an exponential moving average of the DiT weights with decay 0.999.

4 Extension to Long Sequences

A DiT trained on a window of $N_{\max}$ elements will fail to generalize to longer sequences directly, even though more sampled noise tokens can be added. We make two additions to address this.

Temporal compressor. A frozen temporal compressor $C : \mathcal{M}^R \to \mathcal{M}$ maps the latents of R consecutive frames to one frame of compressed latents on the same product-of-spheres manifold. C is a small transformer with one learned register token per spatial position. The register attends over the R input frames at its position, and a direction head and a log-radius head map its output back onto $\mathcal{M}$. A paired decompressor C^{-1} maps each compressed frame back to R frames, and the two are trained before the DiT with a reconstruction loss on $C^{-1}(C(\boldsymbol{z}_{1:R}))$ and a KL term on the compressed latents. The DiT then generates compressed latents, which C^{-1} expands before VAE decoding. With $R = 4$, a DiT window of $N_{\max}$ compressed frames spans $4N_{\max}$ raw frames.

Per-frame flow times and rolling. Because the DiT is trained with independent per-element flow times (§3.4), it can denoise some frames while treating others as clean context. The rolling sampler uses this in two phases. The bootstrap phase denoises a full window of $N_{\max}$ frames from noise under a shared schedule. Each rolling step then shifts the window by one frame, places fresh noise in the last slot, and holds the other $N_{\max} - 1$ slots near the clean end of the flow. The DiT denoises only the new slot and sees the past frames through its own attention. The context level t_c is a sampler setting. Values slightly below 1 add a small amount of noise to the history, following the stabilization of Diffusion Forcing, which makes rolling more robust to errors in previously generated frames. Time patterns of the form $(t_c, \ldots, t_c, t)$ have nonzero probability under independent per-element sampling, so rolling queries the model on inputs of the kind it saw during training. With $R = 4$ and $N_{\max} = 8$, the bootstrap produces 32 raw frames and each rolling step adds 4 more.

5 Experiments

5.1 Experimental Setup

Encoders and heads. We use four pretrained encoders. VGGT (Wang et al., 2025), our primary encoder, predicts per-frame depth, camera intrinsics and extrinsics, and dense point maps. PAGE-4D (Zhou et al., 2026) adds camera-refined 3D point maps. Depth Anything V2 (Yang et al., 2024) is a monocular depth model with a DPT head. DUSt3R (Wang et al., 2024) takes pairs of frames and predicts point maps in a shared coordinate frame. For every encoder, the heads it was released with serve as the frozen $\{H_m\}$.

Dataset. We train on RealCam-Vid (Zheng et al., 2025), which combines RealEstate10K (Zhou et al., 2018), DL3DV-10K (Ling et al., 2024), and MiraData9K (Ju et al., 2024). RealEstate10K and

Table 1: **Depth generation on the first 128 RealCam-Vid test clips at a fixed 8-frame window.** DUSt3R uses the first frame pair of each clip. Hist and Grad denote KL and Wasserstein divergences of log-depth and depth-gradient histograms (§5.1). RAE-XL is not reported for DUSt3R. All heads run in fp32. Lower is better, and the best value per encoder is in bold. †The 95 percent paired bootstrap interval of the ratio to RAE-matched excludes 1 (Table A3).

Encoder	Model	Params	Depth-FVD	Hist KL ($\times 10^2$)	Hist W ($\times 10^2$)	Grad KL ($\times 10^3$)	Grad W ($\times 10^3$)
PAGE-4D	RAE-XL	1.07B	**7.35**	4.19	7.01	5.16	0.80
	RAE-matched	661M	7.47	**3.75**	6.31	5.46	0.78
	MIRAGE	489M	7.67	4.15	**5.30**	**1.32**†	**0.26**†
DAv2	RAE-XL	1.04B	6.52	0.77	**2.85**	1.48	2.04
	RAE-matched	634M	**6.28**	**0.75**	2.89	0.95	1.46
	MIRAGE	489M	6.67†	1.11	6.21†	**0.57**	**0.83**†
DUSt3R	RAE-matched	623M	**3.58**	1.26	2.95	0.52	0.17
	MIRAGE	489M	3.63	**0.79**	**1.39**	**0.10**	**0.05**

DL3DV-10K contain mostly static scenes with calibrated cameras. MiraData9K contains dynamic in-the-wild footage on which the geometric encoders often produce unreliable outputs. We exclude MiraData9K for all four encoders to avoid training the DiT to reproduce these encoder failures.

Training. All runs use an effective batch of 64 clips per optimizer step, bf16 mixed precision, gradient checkpointing, and AdamW with learning rate 10^{-4}, 3000 warmup steps, and cosine decay to 10^{-6}. The DiT has 12 layers, hidden size 1152, and 16 heads, for about 489M trainable parameters. Fixed-window models, for MIRAGE and the baselines, are evaluated with EMA weights after 90k optimizer steps, and the long-sequence models after 40k steps of additional fine-tuning from the base model. The temporal compressor uses $R = 4$ and hidden size 256, and is trained for 20k steps and then frozen. Table A4 lists all hyperparameters.

Baselines. We compare against RAE stage-2 DiT (Zheng et al., 2026) at its released XL size (1.04B to 1.07B parameters, depending on the feature width) and at a smaller configuration obtained by halving the depth of its DiT (RAE-matched, 623M to 661M parameters), the closest size to the 489M MIRAGE model. The baseline diffuses directly in the frozen encoder's feature space, without a VAE bottleneck, and is trained on the features of each encoder. It uses the same conditioning streams (CLIP-L, T5-large, and the DINOv2 image reference), training data, and number of optimizer steps as MIRAGE, so the comparison isolates the latent space and the generative model. The one exception is the MIRAGE model for Depth Anything V2, which is fine-tuned on the short RealCam-Vid captions and sampled from them, while its baselines use the long captions.

Evaluation. The fixed-window comparison uses the first 128 clips of the RealCam-Vid test split and the long-form evaluation the first 500, and the test split keeps all three subsets. Each sample is conditioned on the caption and first frame of one test clip, with a fixed seed per clip. The reference for a clip is the real video passed through the same frozen encoder and heads that decode the samples, so MIRAGE and the RAE baselines share one target per encoder. DUSt3R is pairwise, so every DUSt3R model is evaluated on the first frame pair of each clip. Depth-FVD (Unterthiner et al., 2018) applies an RGB Kinetics I3D network to 16-frame depth videos colorized with a magma colormap after per-clip normalization to the 1st to 99th percentile. Depth histogram divergences (KL and Wasserstein) are accumulated over all pixels, on log-depth with 128 bins and on absolute depth gradients with 64 bins, with bin edges shared by the generated and reference sets. For point clouds we report KL and Wasserstein divergences between the distributions of per-voxel point counts and of per-clip bounding-box volumes. Uncertainty comes from a paired bootstrap that resamples test clips 2000 times, with the same resampled clips for every model of an encoder (App. E).

5.2 Results

Generation quality per encoder. Table 1 compares MIRAGE with both RAE baselines on PAGE-4D, Depth Anything V2, and DUSt3R at a fixed 8-frame window. For each encoder it lists the 489M MIRAGE model, the half-depth RAE-matched baseline (623M-661M), and RAE-XL (1.04B-1.07B). Growing the RAE baseline from $\sim$650M-1.05B parameters changes little at this training

Table 2: **Sampling cost per 8-frame sample on a single GPU.** Measurements use batch size 1, 50 Euler steps, and classifier-free guidance, with the first batch excluded. Memory denotes peak allocation beyond the loaded models. Speed-ups and memory ratios are relative to MIRAGE for the same encoder. VGGT has the same sampling speed as PAGE-4D.

Encoder	Model	Params	Channels per token	Seconds per sample	Peak memory (GB)
PAGE-4D	RAE-XL	1.07B	8192	20.00 (3.4×)	16.59 (13.0×)
	RAE-matched	661M	8192	12.16 (2.1×)	15.88 (12.4×)
	MIRAGE	489M	128	5.89	1.28
DAv2	RAE-XL	1.04B	3072	19.76 (3.3×)	7.66 (4.8×)
	RAE-matched	634M	3072	11.90 (2.0×)	6.93 (4.3×)
	MIRAGE	489M	128	5.93	1.60

Table 3: **Long-form generation with the rolling sampler on VGGT.** Each video contains 64 raw frames; results are reported on the first 500 test clips. Metrics and units match Table 1. All task heads are evaluated in FP32. Lower is better for all metrics.

Conditioning	Depth-FVD	Hist KL ($\times 10^2$)	Hist W ($\times 10^2$)	Grad KL ($\times 10^3$)	Grad W ($\times 10^3$)
Text and first frame	15.10	102.6	64.3	253.1	3.98
Text, first frame and camera trajectory	12.21	76.8	62.0	20.6	2.27

budget, and RAE-XL is not better than RAE-matched on depth-FVD for Depth Anything V2. The depth-FVD of MIRAGE is within 1-6% of both baselines on all three encoders, with only the 6% gap to RAE-matched on Depth Anything V2 significant under paired bootstrap. On the depth-gradient histograms, which measure local depth structure, MIRAGE has the lowest divergences on every encoder, with gradient KL reduced by 1.7-5.3 times. These gains are significant against both baselines on PAGE-4D and Depth Anything V2, and on DUSt3R 84-87% of bootstrap resamples favor MIRAGE. For depth-value histograms, the only significant difference is the larger Wasserstein distance on Depth Anything V2, explained by the 5% depth reduction introduced by its VAE (App. C). Overall, MIRAGE matches the baselines in depth-FVD and depth distribution within sampling uncertainty, while using a smaller network and latent and improving local depth structure. The appendix reports the VAE ceiling, which decodes real clips through the hybrid VAE without generation.

Sampling cost. Table 2 reports the cost of one 8-frame sample at batch size 1, using 50 Euler steps and classifier-free guidance for all model. MIRAGE samples 2.0-2.1 times faster than RAE-matched and 3.3-3.4 times faster than RAE-XL, and it needs 4-13 times less GPU memory beyond the model weights. The difference comes from what each network processes per token: 128 latent channels for MIRAGE against the 3072 or 8192 channels of the raw features, which also makes the RAE network larger. Decoding through the VAE and the frozen heads adds 0.04-0.05 seconds per sample.

Point-cloud geometry. On DUSt3R, the only encoder whose heads output 3D points, MIRAGE matches the distribution of point density more closely than RAE-matched (App. D).

Long-form generation. Only MIRAGE supports rolling generation. The RAE stage-2 DiT is trained on a fixed 8-frame window and cannot condition on past frames at inference, and concatenating independently sampled 8-frame chunks produces a discontinuity at every chunk boundary. We thus report long-form results for MIRAGE alone, on VGGT with the temporal compressor. Each video has 64 raw frames, 16 compressed frames generated as a bootstrap window of 8 followed by 8 rolling steps. One variant additionally conditions on the camera trajectory of the test clip through a fourth conditioning stream. Table 3 reports the results. Camera conditioning lowers depth-FVD from 15.10 to 12.21 and the gradient KL by a factor of 12. The large histogram divergences of both variants come mostly from depth scale: generated depth is about twice the reference depth over the rollout, and median normalization reduces the histogram KL of the camera-conditioned model from 0.77 to about 0.01.

Table 4: Ablations on VGGT, first 500 test clips, with every model evaluated after 70k training steps. Each row changes one setting of the reference configuration and is retrained from scratch. Every row decodes depth with bf16 heads, so the table reports only depth-FVD and the depth-value histograms, which head precision barely changes (App. C). The long-sequence reference is not evaluated on 500 clips, so that group compares the variants with one another. Lower is better throughout, and the best value per group is in bold.

Setting	Depth-FVD	Hist KL ($\times 10^2$)	Hist W ($\times 10^2$)
Fixed window, 32 frames			
Reference	4.80	1.62	**4.49**
No per-token cross-attention (pooled adaLN only)	6.54	2.05	6.26
No DINOv2 image reference (text only)	5.37	5.35	11.57
No T5 (CLIP-L only)	4.89	5.03	11.14
No self-REPA	5.17	1.21	5.72
Self-REPA at block 9 instead of 5	**4.78**	1.31	5.49
Self-REPA at blocks 5 and 9	4.92	2.00	8.56
Shared flow time (no per-frame times)	4.86	**0.93**	5.51
Long sequence, 64 frames			
No temporal compressor	9.18	5.57	14.82
Constant context probability ($p_{\text{ctx}} = 0.5$)	4.76	1.93	8.98
Anti-static loss (weight 0.05)	4.87	**1.85**	8.84
Attention expander for self-REPA	**4.47**	2.10	**7.39**

5.3 Ablations

Table 4 reports ablations on VGGT. Each model is trained for 70k steps from the reference configuration with one setting changed.

Conditioning. Pooled adaLN conditioning without per-token cross-attention gives the worst fixed-window depth-FVD (6.54 against 4.80). Dropping the DINOv2 or T5 stream changes depth-FVD less but more than triples the depth-histogram KL, so the per-token streams shape the distribution of scene depth.

Self-REPA and flow times. Removing self-REPA raises depth-FVD to 5.17 and the histogram Wasserstein distance to 5.72. Block 9 matches block 5 in depth-FVD, and a second head at block 9 worsens both histogram divergences. A shared flow time matches the reference in depth-FVD (4.86), so per-frame times cost little short-clip quality and enable rolling generation.

Long sequences. Removing the temporal compressor roughly doubles depth-FVD (9.18, against 4.47 to 4.87 for all other variants) and nearly triples the histogram KL. A constant context probability or an anti-static loss changes depth-FVD little, and an attention expander for self-REPA gives the lowest depth-FVD (4.47).

6 Conclusion

A pretrained transformer with task heads can be turned into a generative model of its own feature space, and one such model produces every output the heads support. MIRAGE does this with a hybrid product-of-spheres VAE, a rectified flow that respects its geometry, self-REPA alignment to the encoder being modeled, and a temporal compressor with per-frame flow times for rolling generation, and the same recipe works across four geometric encoders. Compared with diffusion in the raw encoder features, MIRAGE works in a latent 16 to 64 times smaller, stays within 1 to 6 percent in depth-FVD, matches it in depth distribution within sampling uncertainty, improves local depth structure, and samples twice as fast with 4 to 13 times less memory. Because samples are read by the encoder's own heads, their quality is bounded by those heads and by how well the VAE reconstructs the features, and improvements to either carry over to MIRAGE directly. The main weaknesses we observe are a depth-scale error in long rollouts and a depth reduction of about 5 percent introduced by the Depth Anything V2 VAE. Our experiments cover geometric encoders only, although encoders with semantic or non-visual heads fit the same interface.

AI USE STATEMENT

In this work, we used generative AI tools to help revise the text of this paper and to help write and debug the scripts for evaluation and cluster jobs. The research questions, method, experiments, and conclusions are our own. We checked all text and code produced with this assistance and take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

§3 and §4 define every component of MIRAGE, and App. A and B give the hybrid VAE objective and the full DiT architecture. Table A4 lists the hyperparameters of every training stage. All four encoders and their task heads are publicly released checkpoints, and RealCam-Vid and its subsets are public datasets. §5.1 specifies the training data, the RAE baselines, the test clips used for evaluation, the fixed per-clip sampling seeds, and each metric with its binning, colormap, and feature network. All task heads run in fp32 at evaluation, except in the VGGT ablations, which share bf16 heads across rows, and App. C shows how strongly head precision affects the gradient metrics. App. E describes the bootstrap behind the reported intervals. The code will be released at the end of the review cycle.

REFERENCES

Michael S. Albergo, Nicholas Matthew Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *J. Mach. Learn. Res.*, 26:209:1–209:80, 2025.

Andrew Bond, Ilkin Umut Melanlioglu, Erkut Erdem, and Aykut Erdem. Beyond gaussian bottlenecks: Topologically-aligned encoding of vision-transformer feature spaces. In *ICCV 2026 Beyond Euclidean Workshop*, 2026. URL https://openreview.net/forum?id=2rFwBlgbsl.

Nicola De Cao and Wilker Aziz. The power spherical distribution. *ArXiv*, abs/2006.04437, 2020.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 24081–24125, 2024.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, 2024.

Heinrich Dinkel, Xingwei Sun, Gang Li, Jiahao Mei, Yadong Niu, Jizhong Liu, Xiyang Li, Yifan Liao, Jiahao Zhou, Junbo Zhang, and Jian Luan. Dashengtokenizer: One layer is enough for unified audio understanding and generation, 2026. URL https://arxiv.org/abs/2602.23765.

Patrick Esser, Sumith Kulal, A. Blattmann, Rahim Entezari, Jonas Muller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. Miradata: A large-scale video dataset with long durations and structured captions. *Advances in Neural Information Processing Systems*, 37:48955–48970, 2024.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169. IEEE, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ArXiv*, abs/2210.02747, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *ArXiv*, abs/2209.03003, 2022.

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, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. *ArXiv*, abs/2502.06764, 2025.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *ArXiv*, abs/2104.09864, 2021.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotný. Vggt: Visual geometry grounded transformer. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in neural information processing systems*, 37:21875–21911, 2024.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *ArXiv*, abs/2410.06940, 2024. URL `https://api.semanticscholar.org/CorpusID:273229262`.

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

Guangcong Zheng, Teng Li, Xianpan Zhou, and Xi Li. Realcam-vid: High-resolution video dataset with dynamic scenes and metric-scale camera movements. *arXiv preprint arXiv:2504.08212*, 2025.

Kaichen Zhou, Yuhan Wang, Grace Chen, Gaspard Beaudouin, Fangneng Zhan, Paul Pu Liang, and Mengyu Wang. PAGE-4d: Disentangled pose and geometry estimation for VGGT-4d perception. In *International Conference on Learning Representations (ICLR)*, 2026.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

APPENDIX

The appendix contains the hybrid VAE details (§A), the full DiT architecture (§B), the VAE ceiling and head-precision analysis (§C), point-cloud evaluation (§D), bootstrap intervals (§E), implementation details and hyperparameters (§F), additional qualitative results (§G), and audio experiments (§H).

A HYBRID VAE: LOG-RADIUS EXTENSION

Bond et al. (2026) derive the direction side of the hybrid VAE in full, including the Power Spherical posterior on $\mathbb{S}^{d-1}$, the uniform prior, and their closed-form KL. Here we describe the log-radius extension.

Variational family. Let $\boldsymbol{x} \in \mathcal{X}$ be an input element and $E(\boldsymbol{x}) \in \mathbb{R}^{L \times D_E}$ its encoder output. For each token $i \in \{1, \dots, L\}$ and sub-sphere $k \in \{1, \dots, K\}$, the VAE encoder outputs parameters $(\mu^r_{i,k}, \sigma^r_{i,k}) \in \mathbb{R} \times \mathbb{R}_{>0}$ of the log-normal posterior

$$q_\phi(\log r_{i,k} \mid \boldsymbol{x}) \;=\; \mathcal{N}\big(\mu^r_{i,k}(\boldsymbol{x}),\; (\sigma^r_{i,k}(\boldsymbol{x}))^2\big).$$

The prior is a standard normal on the log-radius, $p(\log r_{i,k}) = \mathcal{N}(0, 1)$, centered at unit radius.

Closed-form KL. The KL divergence between the posterior and the prior for token i and sub-sphere k is

$$\mathrm{KL}\big(q_\phi \,\|\, p\big)_{i,k} \;=\; \tfrac{1}{2}\big[\,(\mu^r_{i,k})^2 + (\sigma^r_{i,k})^2 - 1 - \log(\sigma^r_{i,k})^2\,\big]\,,$$

summed over all tokens and sub-spheres. We write $\mathrm{KL}^{\boldsymbol{u}}$ for the direction-side KL of Bond et al. (2026), between the Power Spherical posterior and the uniform distribution on $\mathbb{S}^{d-1}$.

Combined objective. The KL part of the VAE loss is

$$\mathcal{L}_{\mathrm{KL}} \;=\; \lambda\, w(s)\,\Big(\mathrm{KL}^{\boldsymbol{u}} \;+\; \rho\,\mathrm{KL}^{\log r}\Big)\,,$$

with $\lambda = 10^{-4}$, a radius weight $\rho = 0.05$, and a weight $w(s)$ that ramps linearly from 0 to 1 over the first 5000 training steps. The log-radius posterior is initialized with standard deviation 0.5. The small radius weight lets magnitude information pass through the bottleneck while the directions are regularized toward their full prior. The remaining terms are the feature reconstruction loss (MSE plus a cosine term on the decoded features) and the auxiliary losses of Bond et al. (2026), with the weights listed in Table A4.

B FACTORIZED DIFFUSION TRANSFORMER: FULL ARCHITECTURE

Each DiT block applies four residual sub-layers in order: (1) self-attention within each element, (2) self-attention across elements, (3) cross-attention to the concatenated conditioning tokens $\boldsymbol{c}_{\mathrm{tok}}$ of the three streams (CLIP-L, T5-large, DINOv2 image reference), and (4) a SwiGLU feed-forward network. Each sub-layer is preceded by adaptive layer normalization, and its output is scaled by a learned residual gate.

Position embeddings. Self-attention within an element uses 3D rotary position embeddings (Su et al., 2021) over the grid of $L = 256$ patch tokens, and the summary token receives no rotary embedding. It is identified by a learned token embedding instead. Self-attention across elements adds a learned absolute embedding $\boldsymbol{p}_n \in \mathbb{R}^{D_{\mathrm{DiT}}}$ to every token of element n, which gives this path access to sequence order.

Modulation. The shift, scale, and gate of each sub-layer are predicted by a per-block MLP from the sum of the flow-time embedding and the pooled conditioning $\sum_c W^{\mathrm{pool}}_c \bar{E}_c(\boldsymbol{y}_c)$. The model stacks 12 blocks with hidden size 1152 and 16 heads of dimension 72.

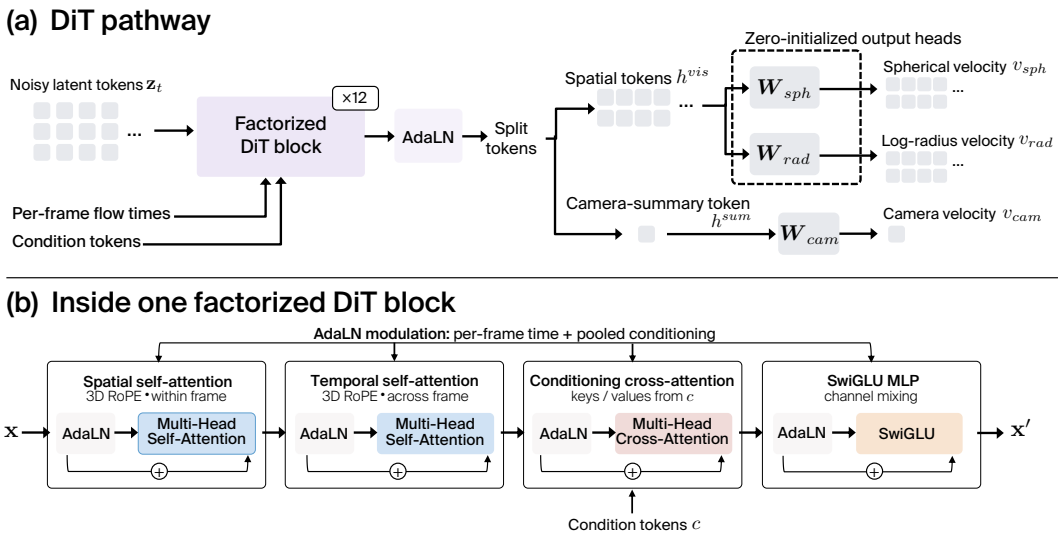


Figure A1: **MIRAGE DiT architecture. (a)** Noisy latent tokens z_t are processed by 12 factorized DiT blocks, conditioned on per-frame flow times $t_{1:N}$ and conditioning tokens c. A final AdaLN produces a camera-summary token h_n^{sum} and spatial tokens $h_{n,1:L}^{\text{vis}}$. Zero-initialized output heads map the spatial tokens to spherical and log-radius velocities, v_θ^{sph} and v_θ^{rad}, and the camera-summary token to camera velocity v_θ^{cam}. **(b)** Each factorized block applies, in sequence, spatial self-attention within each frame, temporal self-attention across frames, conditioning cross-attention using keys and values from c, and a SwiGLU MLP. Every stage is preceded by AdaLN and followed by a residual addition; AdaLN is modulated by per-frame flow time and pooled conditioning. Spatial and temporal attention use 3D RoPE.

Cross-attention scale. Each block has a learnable scalar $\gamma_l \in \mathbb{R}$ that multiplies the cross-attention residual, initialized to $\gamma_l = 2.0$, so the network can adjust the strength of conditioning with depth.

Input embedding and output heads. At the input, the direction and log-radius of each patch latent are embedded separately to dimension D_{DiT} and summed. The summary latent has its own input projection and a learned stream embedding. At the output, three zero-initialized linear heads predict the direction, log-radius, and summary velocities. With zero initialization the network starts by predicting zero velocity, and the velocity field is learned from there. Figure A1 shows the full block diagram.

C VAE CEILING AND HEAD PRECISION

The VAE ceiling passes each real clip through the frozen encoder, a deterministic VAE encode and decode, and the frozen heads, without any generation. It bounds what a generator in the MIRAGE latent space can reach. Table A1 reports it on the first 100 test clips next to the RAE-matched baseline on the same clips, with the heads run in fp32 and in bf16. Depth scale is the median ratio of ceiling to reference depth per clip.

With fp32 heads the ceiling is below the RAE samples on both histogram KL divergences for PAGE-4D and Depth Anything V2, so the hybrid VAE preserves the depth distribution closely despite compressing the features 24 to 64 times. The Depth Anything V2 VAE reduces depth by about 5 percent, which accounts for its larger histogram Wasserstein distance. Running the same heads in bf16 changes the depth histograms little in absolute terms but raises the gradient KL by a factor of 5 to 500, because bf16 rounding makes neighbouring depth values equal or places them on a coarse grid. All results in Tables 1 to 3 therefore run the heads in fp32. The VGGT ablations of Table 4 decode with bf16 heads for every row, so that table reports only depth-FVD and the depth-value histograms.

Table A1: **VAE ceiling on the first 100 test clips at the native clip length, against RAE-matched samples on the same clips.** Units match Table 1. Histograms use 16,384 sampled pixels per clip, which reproduces the full-pixel values to three significant figures. DUSt3R is evaluated on all eight frame pairs here, so no RAE row is shown.

Encoder	Setting	Depth scale	Hist KL ($\times 10^2$)	Hist W ($\times 10^2$)	Grad KL ($\times 10^3$)	Grad W ($\times 10^3$)
PAGE-4D	VAE ceiling, fp32 heads	1.000	0.05	0.14	0.56	0.32
	VAE ceiling, bf16 heads	1.001	0.24	0.15	2.98	0.24
	RAE-matched	0.989	4.08	6.51	5.53	0.75
DAv2	VAE ceiling, fp32 heads	0.952	0.32	5.18	0.23	0.50
	VAE ceiling, bf16 heads	0.952	0.57	5.22	47.63	2.72
	RAE-matched	1.002	0.79	2.76	1.34	1.82
DUSt3R	VAE ceiling, fp32 heads	0.998	0.13	0.96	0.12	0.07
	VAE ceiling, bf16 heads	0.996	0.15	1.00	62.10	0.40

D Point-cloud geometry

DUSt3R allows for outputting 3D points, so it allows a direct comparison of generated scene geometry. Table A2 reports the voxel-density and bounding-box volume divergences of the point clouds on the first frame pair. MIRAGE matches the distribution of point density more closely than RAE-matched, with a density KL of 0.0007 against 0.0021, and has a slightly lower volume KL, but a larger volume Wasserstein distance (0.75 against 0.17).

Table A2: **Point-cloud geometry for DUSt3R on the first frame pair of the first 128 test clips.** Density compares distributions of per-voxel point counts (voxel size 0.05). Volume compares distributions of per-clip bounding-box volumes. Lower is better throughout.

Model	Density KL	Density W	Volume KL	Volume W
RAE-matched	0.0021	5.85	0.323	**0.168**
MIRAGE	**0.0007**	**3.31**	**0.305**	0.753

E Bootstrap intervals for the main comparison

Each metric of Table 1 is recomputed from per-clip statistics: I3D features for depth-FVD, and per-clip histogram counts on the bin edges of the full evaluation for the histogram divergences. These reproduce the reported values exactly. We then resample the 128 test clips with replacement 2000 times. Every model of an encoder uses the same resampled clips, and so does its reference set, so the comparison is paired. The histogram bin edges stay fixed across resamples. Resampling with replacement duplicates clips, which biases FVD and plug-in KL estimates upward, so Table A3 reports the ratio of MIRAGE to each baseline, in which this bias largely cancels, with its 2.5th and 97.5th percentiles.

Table A3: **Paired bootstrap over the 128 test clips of Table 1 (2000 resamples).** Each entry is the ratio of the MIRAGE metric to the baseline metric, with the 95 percent interval in brackets. Values below 1 favor MIRAGE. Bold marks intervals that exclude 1.

Encoder	Baseline	Depth-FVD	Hist KL	Hist W	Grad KL	Grad W
PAGE-4D	RAE-XL	1.043 [0.961, 1.163]	0.991 [0.592, 1.706]	0.755 [0.515, 1.196]	**0.255 [0.087, 0.812]**	**0.329 [0.199, 0.610]**
	RAE-matched	1.028 [0.947, 1.141]	1.104 [0.633, 1.881]	0.840 [0.545, 1.332]	**0.241 [0.082, 0.823]**	**0.339 [0.198, 0.643]**
DAv2	RAE-XL	1.024 [0.967, 1.091]	1.437 [0.613, 2.389]	2.183 [0.728, 3.184]	**0.385 [0.205, 0.934]**	**0.405 [0.247, 0.664]**
	RAE-matched	**1.062 [1.005, 1.126]**	1.482 [0.724, 2.394]	**2.150 [1.025, 3.053]**	0.599 [0.319, 1.297]	**0.569 [0.346, 0.961]**
DUSt3R	RAE-matched	1.013 [0.926, 1.100]	0.628 [0.465, 1.419]	0.473 [0.312, 1.408]	0.190 [0.108, 2.145]	0.291 [0.166, 2.600]

F Implementation Details and Hyperparameters

Tables A4 and A5 summarize the training and sampling configurations used in our experiments. Table A4 reports the hyperparameters of the hybrid VAE, temporal compressor, fixed-window DiT, and long-sequence DiT, together with the sampling settings. Unless otherwise noted, these configurations correspond to Depth Anything V2; the other encoders differ only in encoder-specific feature dimensions and the exceptions stated in the table. Table A5 gives the corresponding configurations of the RAE stage-2 baselines. The baselines use the same conditioning streams, training data, sampling procedure, and effective batch size as MIRAGE, so that the comparison primarily reflects the choice of generative representation and model architecture.

Table A4: **Hyperparameters of the four training stages and the sampler, from the Depth Anything V2 configurations.** The other encoders differ only in the width of the extracted features and, for VGGT, in 8 VAE attention heads. The long-sequence DiT trains on 32 raw frames (8 compressed frames with $R = 4$). Batch sizes are per GPU, on 4 GPUs.

Hyperparameter	VAE	Compressor	DiT (fixed)	DiT (long-seq)
Model				
Encoder feature layers		2, 5, 8, 11 of the ViT-B encoder, concatenated per token		
Hidden dim	1024	256	1152	1152
Layers	12	2	12	12
Heads	16	4	16	16
Latent dim per token	128	128	128	128
Sub-spheres ($K \times d$)	16×8	16×8	16×8	16×8
Tokens per frame (L)	256	256	256	256
Frames per sample	4	16	8	32 raw / 8 compressed
Compression rate (R)	—	4	—	4
Self-REPA projection	—	—	3-layer MLP, bottleneck 2048	3-layer MLP, bottleneck 2048
Optimization				
Optimizer	AdamW	AdamW	AdamW	AdamW
Learning rate	2×10^{-4}	5×10^{-4}	1×10^{-4}	1×10^{-4}
Min LR (cosine)	—	1×10^{-6}	1×10^{-6}	1×10^{-6}
Warmup steps	5000	1000	3000	3000
Weight decay	0.05	0.01	0.01	0.01
Max grad norm	1.0	1.0	1.0	1.0
Per-GPU batch $\times$ accumulation	12×1	4×1	8×2	8×2
Training length	12 epochs	20k steps	90k steps	40k steps from the fixed DiT
Mixed precision	bf16	bf16	bf16	bf16
EMA decay	—	—	0.999	0.999
VAE objective				
Feature reconstruction (MSE $+ 0.5 \times$ cosine)	1.0	—	—	—
Depth-head consistency	0.3	—	—	—
JEPA / topology / variance	5.0 / 0.02 / 0.01	—	—	—
KL weight λ / radius weight ρ / KL ramp steps	10^{-4} / 0.05 / 5000	—	—	—
κ init / radius init std	30 / 0.5	—	—	—
Compressor objective				
Reconstruction MSE / cosine	—	1.0 / 0.5	—	—
Direction / radius terms	—	1.0 / 1.0	—	—
Motion / within-window variance	—	10.0 / 5.0	—	—
DiT objective				
Direction / log-radius / summary flow weights	—	—	1 / 1 / 1	1 / 1 / 1
Training flow-time distribution	—	—	uniform on $[0, 1]$	uniform on $[0, 1]$
Self-REPA weight / DiT block / layer of E	—	—	0.5 / 5 / last	0.5 / 5 / last
CFG drop probability	—	—	0.35	0.35
Flow-time curriculum: $\alpha = 0$ / ramp to 1 (steps)	—	—	15000 / 15000	15000 / 15000
Context probability p_{ctx}: hold / decay (steps)	—	—	0.5 for 30k / to 0 over 40k	0.5 for 30k / to 0 over 40k
Context dropout probability	—	—	0.1	0.1
Sampling				
Euler steps	—	—	50	50
CFG scale	—	—	4.0 to 1.0, linear in flow time	4.0 to 1.0, linear in flow time
Rolling context level t_c	—	—	—	1.0

G Qualitative Results

Figures A2–A6 show qualitative generations across the four pretrained geometric encoders used in our experiments. The examples illustrate two properties of MIRAGE. First, generated features remain interpretable by the frozen heads of different encoders: DUSt3R, PAGE-4D, and VGGT decode them into structured depth and 3D geometry, while Depth Anything V2 produces spatially coherent depth maps. Second, the generated representations retain temporal structure. The fixed-window Depth Anything V2 examples in Fig. A3 show the evolution of depth within the training

Table A5: **Hyperparameters of the RAE stage-2 baselines.** Both use the wide-head DiT of Zheng et al. (2026) on the raw encoder features, the same three conditioning streams, data, sampler settings, and effective batch of 64 clips as MIRAGE. Batch sizes are per GPU, on 4 GPUs.

Hyperparameter	RAE-matched	RAE-XL
DiT blocks (width 1152, 16 heads)	14	28
Wide head blocks (width 2048, 16 heads)	2	2
Parameters	623M to 661M	1.04B to 1.07B
Input channels per token	2048 (DUSt3R), 3072 (DAv2), 8192 (PAGE-4D)	
MLP / normalization	SwiGLU, ratio 4 / RMSNorm	
Position encoding	RoPE and learned position embedding	
Optimizer / learning rate	AdamW / 1×10^{-4}, cosine to 1×10^{-6}	
Warmup steps / weight decay / max grad norm	3000 / 0.01 / 1.0	
Per-GPU batch $\times$ accumulation	2×8	
Training steps	90k	
EMA decay	0.9995	
CFG drop probability	0.35	

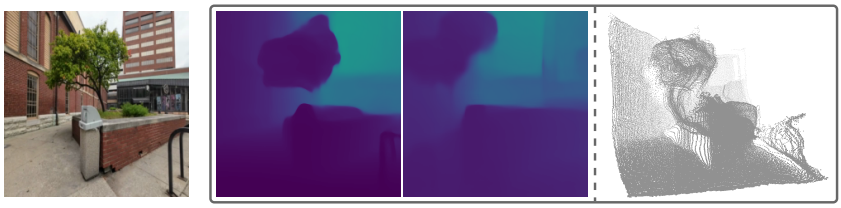
horizon, while Fig. A4 shows selected frames from 64-frame rollouts, demonstrating that the rolling formulation can extend this behavior beyond the DiT's fixed context window.

The multi-head examples further illustrate the main motivation for generating in task-aware feature space. For DUSt3R (Fig. A2), PAGE-4D (Fig. A5), and VGGT (Fig. A6), the depth and 3D outputs are obtained from the same generated encoder representation rather than from separately trained generators. Across the shown indoor and outdoor scenes, the frozen heads recover broadly consistent scene layout and geometry from these generated features.

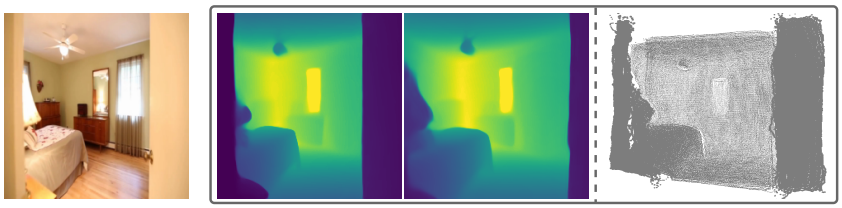
Figure A7 illustrates two recurring limitations. When the frozen encoder or its task heads produce unreliable geometry for the conditioning input, MIRAGE inherits this limitation and the generated geometry also degrades. In other cases, particularly for geometrically difficult scenes, the long-form model can underrepresent temporal change and produce nearly static depth sequences. These examples reflect the two main sources of error in the current framework: limitations of the pretrained representation being modeled and imperfect temporal dynamics during rolling generation.

H Audio Experiments

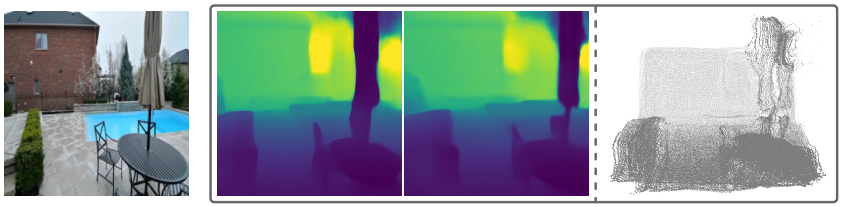
Using the same standard setup, we use the Dasheng tokenizer (Dinkel et al., 2026) as the underlying encoder and task head. The DiT is trained with the same setup, architecture, and losses as all the vision versions, for the same length of time. However, for conditioning, we use only the two text encoders, no first frame conditioning. Figure A8 provides a comparison of a GT and generate spectrogram produced by our model, showing that our model can in fact learn to reproduce the audio's temporal and spectral structure.



"A serene urban plaza with brick buildings, trees, and bike racks, surrounded by modern skyscrapers under a cloudy sky."

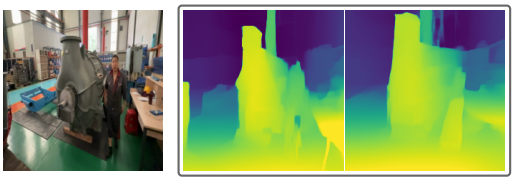


"A cozy bedroom with wooden floors, a bed with floral bedding, a dresser, and a ceiling fan."

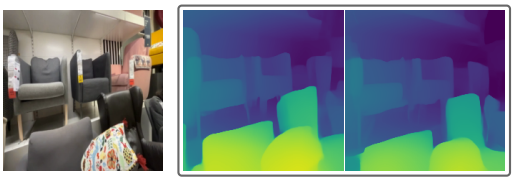


"A serene backyard with a swimming pool, patio furniture, and blooming trees, surrounded by brick houses."

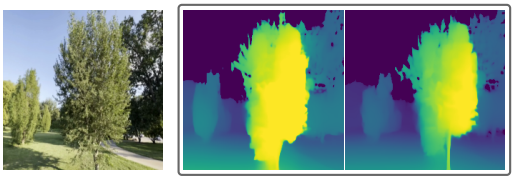
Figure A2: **Qualitative samples with DUSt3R.** Each row shows the conditioning input frame, depth predictions decoded from the MIRAGE-generated DUSt3R feature pair, and the corresponding 3D point cloud reconstructed in the shared coordinate frame. The text prompt used for conditioning is shown below each example. These results illustrate that MIRAGE generates DUSt3R representations that preserve coherent scene geometry across diverse indoor and outdoor environments.



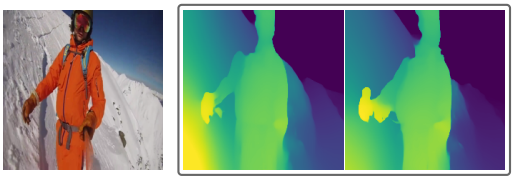
"A large industrial machine is showcased in a factory setting, with a worker inspecting it."



"A furniture store display featuring various armchairs and sofas with price tags, showcasing modern designs in neutral tones."

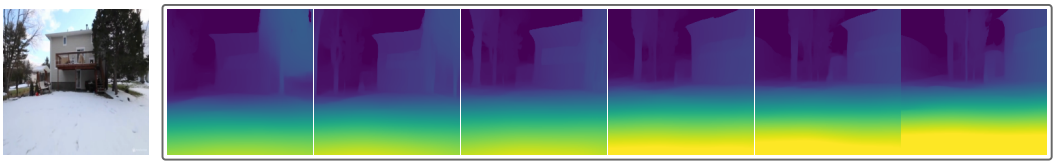


"A serene park scene with lush green trees and a clear blue sky, capturing the beauty of nature on a sunny day."

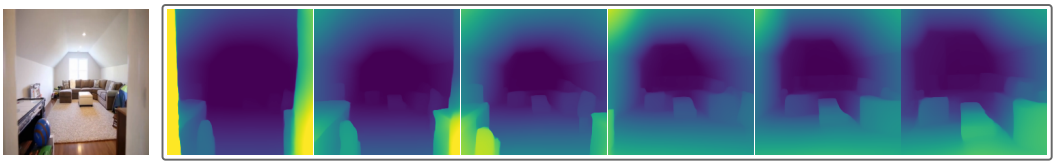


"A climber ascends a steep, snow-covered mountain, navigating with precision and determination under clear skies."

Figure A3: **Qualitative fixed-length samples with Depth Anything V2.** Each example shows an input image and its corresponding text prompt, followed by the 1st and 8th generated depth maps from an 8-frame generation. The examples illustrate the spatial structure and temporal consistency of depth generated within the DiT's fixed training window.

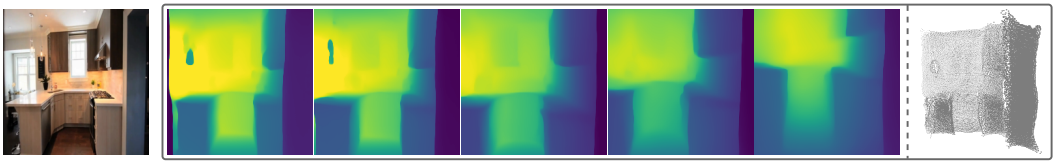


"A snowy backyard with a two-story house, a deck, and trees under a partly cloudy sky."

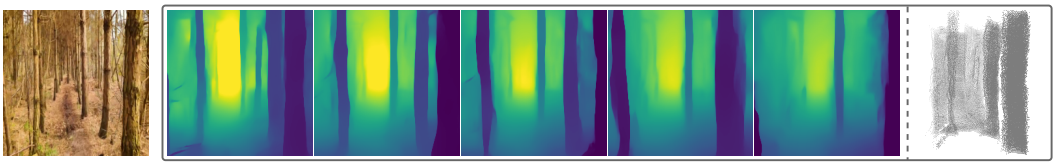


"A cozy attic room features a sectional sofa, a play kitchen, and a small tent, creating a fun and inviting space for children."

Figure A4: **Qualitative long-sequence samples with Depth Anything V2.** Each row shows an input image and its corresponding text prompt, followed by selected generated depth maps from a 64-frame generation. The examples illustrate the temporal consistency and spatial structure of depth generated beyond the DiT's fixed training window using the long-sequence formulation.



"A modern kitchen with dark cabinets, light countertops, and a gas stove, featuring a clean and organized design."



"A serene forest path lined with bare trees, bathed in sunlight, inviting exploration and tranquility."

Figure A5: **Qualitative samples with PAGE-4D.** Each row shows the conditioning input frame, representative depth maps decoded from the MIRAGE-generated PAGE-4D feature sequence, and the corresponding 3D geometry visualized as a point cloud. The text prompt used for conditioning is shown below each example. The results illustrate that MIRAGE can generate PAGE-4D representations whose frozen task heads recover coherent scene structure across both indoor and outdoor scenes.

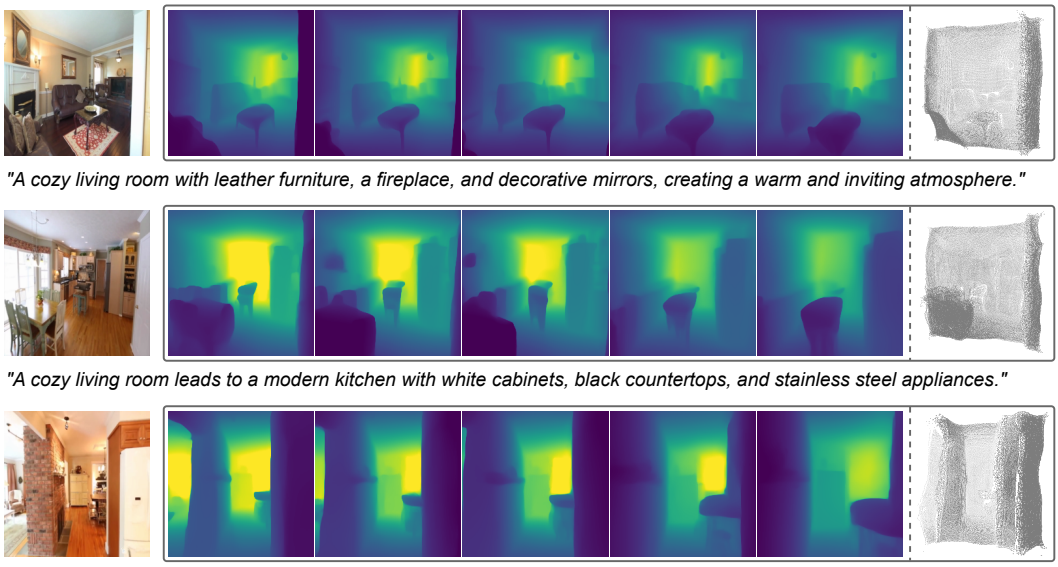


Figure A6: **Qualitative samples with VGGT**. Each row shows the conditioning input frame, representative depth maps decoded from the MIRAGE-generated VGGT feature sequence, and the corresponding 3D point cloud produced by the frozen VGGT task head. The text prompt used for conditioning is shown below each example. These samples illustrate how a single generated representation can be decoded into temporally consistent depth and 3D scene geometry across diverse scenes.

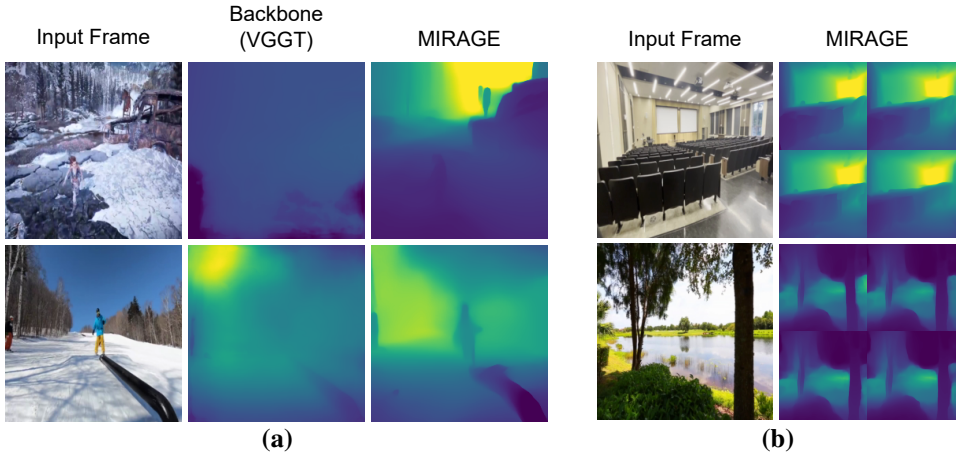


Figure A7: **Failure cases of our model.** **(a)** For challenging conditioning inputs where the frozen ViT backbone produces degraded geometric estimates, the generated outputs are also of reduced quality, reflecting limitations of the backbone representation and downstream task head. **(b)** For geometrically challenging scenes, the DiT can fail to capture temporal evolution, yielding nearly static generation. The 1st, 16th, 32nd, and 48th generated depth maps are shown in both instances.

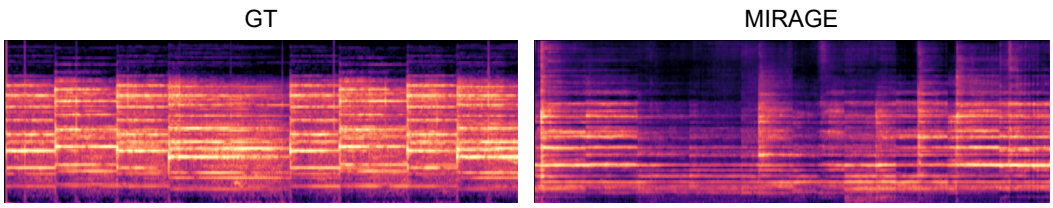


Figure A8: **Qualitative audio generation result.** Spectrograms of Ground-truth (GT) and MIRAGE-generated audio shown side by side, illustrating the model's ability to reproduce the reference audio's temporal and spectral structure.