

Multimodal Generation III

Large Foundational Models and Physical World Models for Multimodal AI

Lecture 7

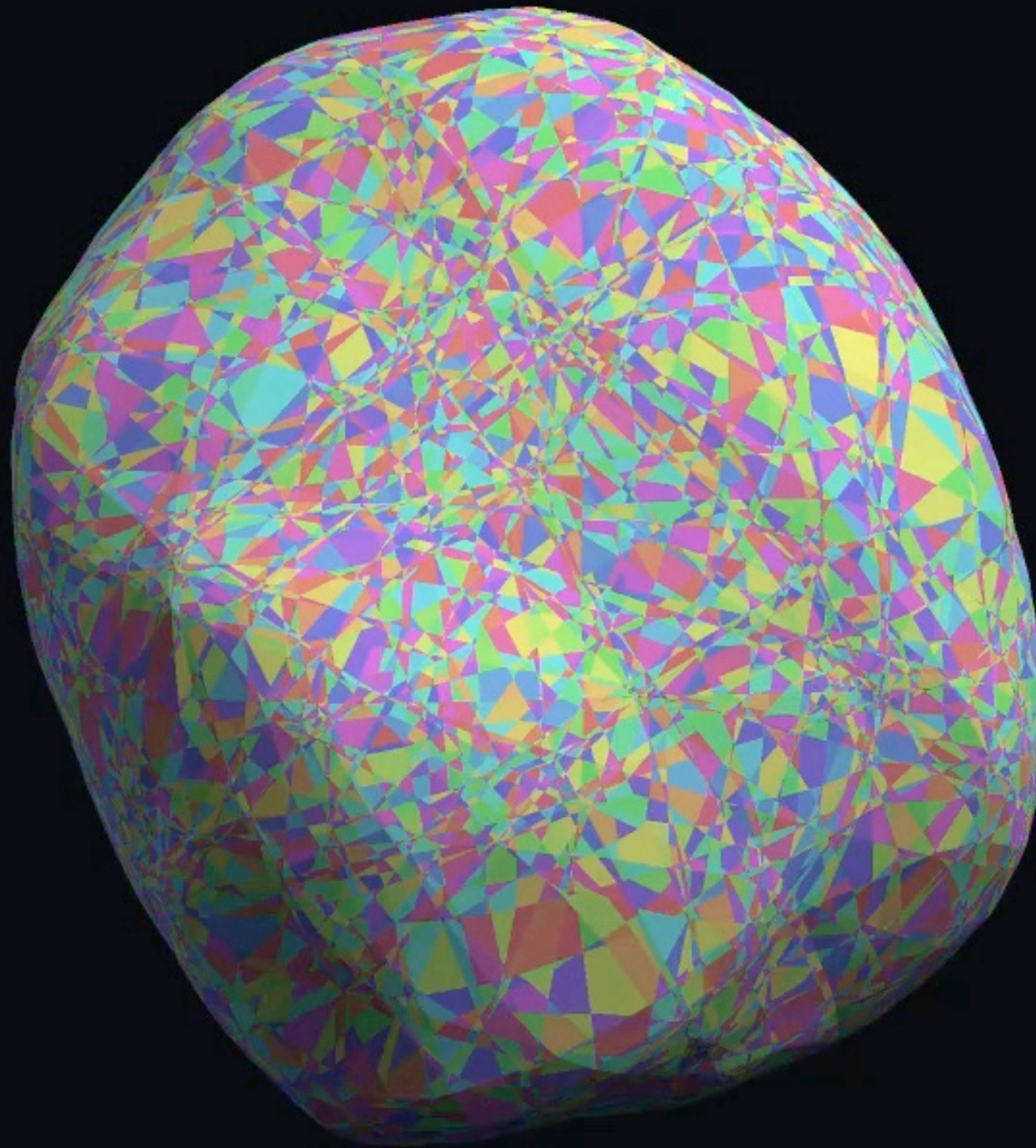
Jin-Hwa Kim and Sangdoo Yun

Multimodal Generation series

- *Multimodal Generation I on Oct 2*
 - NeRF and pinhole camera system
- *Multimodal Generation II on Oct 9 (Online)*
 - Topics in Neural Graphics: Camera pose refinement, Hash Grid, and surface modelings
- *Multimodal Generation III on Oct 16*
 - 3D generation using SDS, world models, and Seoul World Model

Iteration 1 / 5,000

35,268 polygons (exact zero set)



4x64 ReLU MLP · exact surface, no marching cubes

Objective

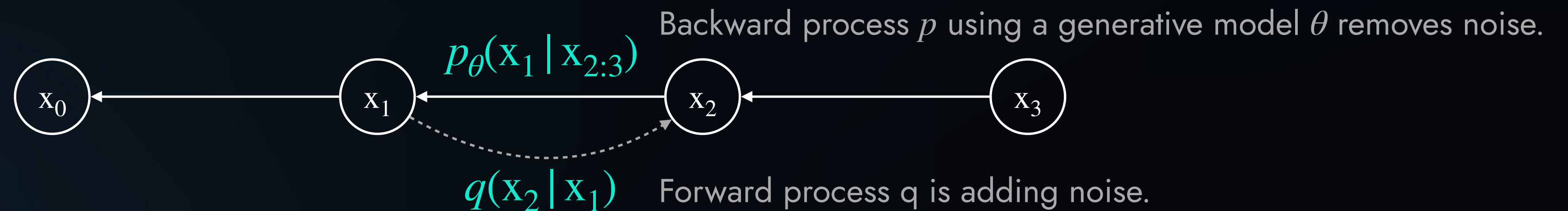
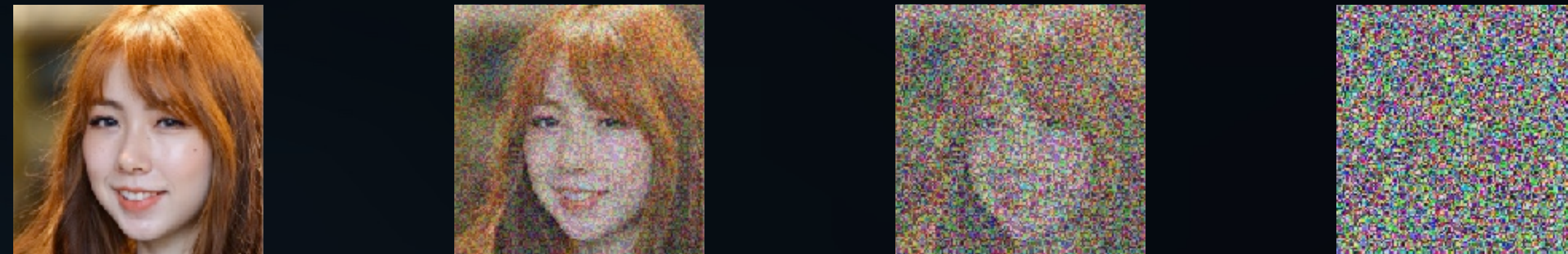
- Focus on the core concepts of the Variational Diffusion Model (Kingma et al., 2021), simplifying diffusion loss by the KL divergence between two isotropic Gaussians.
- Explore key theories and concepts of the score distillation method SDS, widely used in multimodal image and 3D generation.
- Introduce the fundamentals of world models, and present **Seoul World Model** (Seo et al., 2026), our effort to build a physical world model powering real-world applications.

Overview

DDPM & DDIM (though DDIM is not much covered)

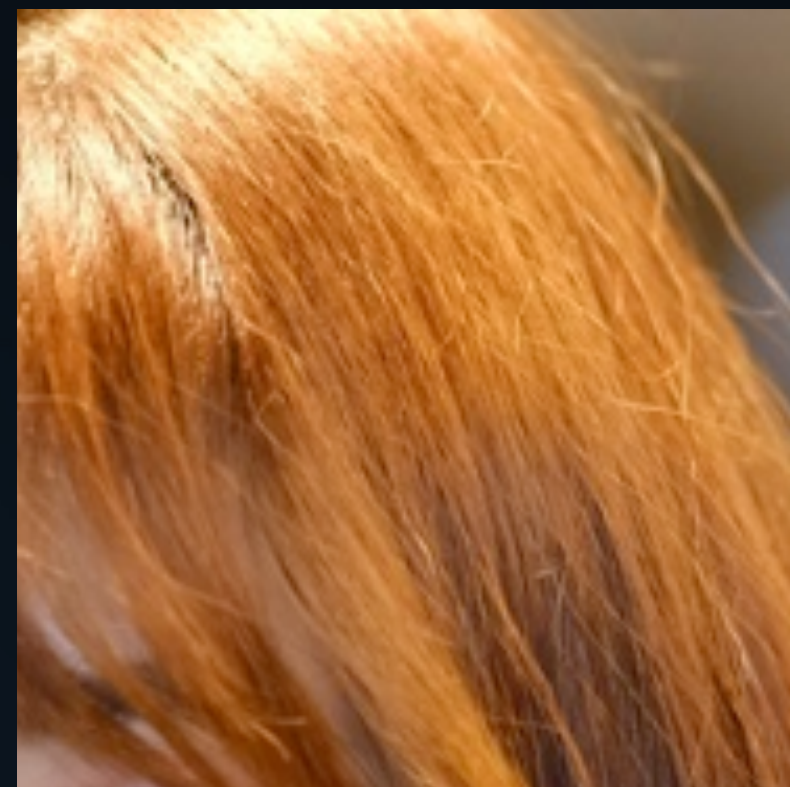
Diffusion models

- One of successful generative models for these days
- The generative process *progressively* denoises a noisy observation called:
Denoising diffusion probabilistic models (DDPM) (Ho et al., 2020).



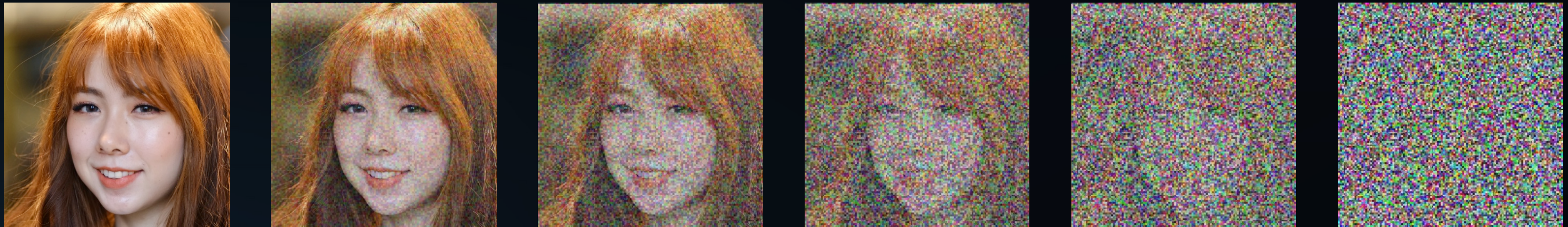
Stationarity and locality

- “They also make strong and mostly correct assumptions about the nature of images (namely, *stationarity* of statistics and *locality* of pixel dependencies).”
(Krizhevsky et al., 2012)
- Roughly speaking, the stationary and locality allows us *pattern* recognition and *hierarchical* representation learning.



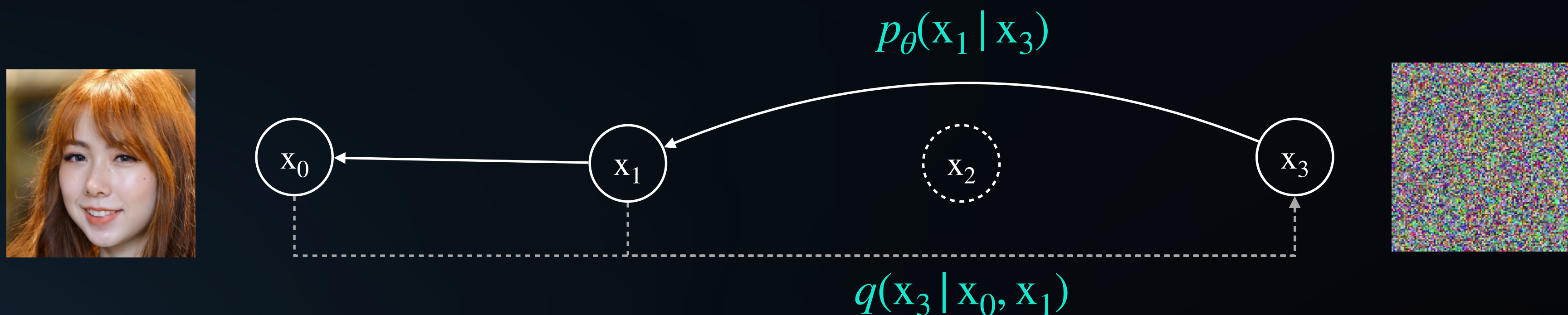
Drawback of DDPM

- Many steps are required in order to produce a high-quality sample.
- Since the *denoising difficulty* increases as the number of steps decreases.



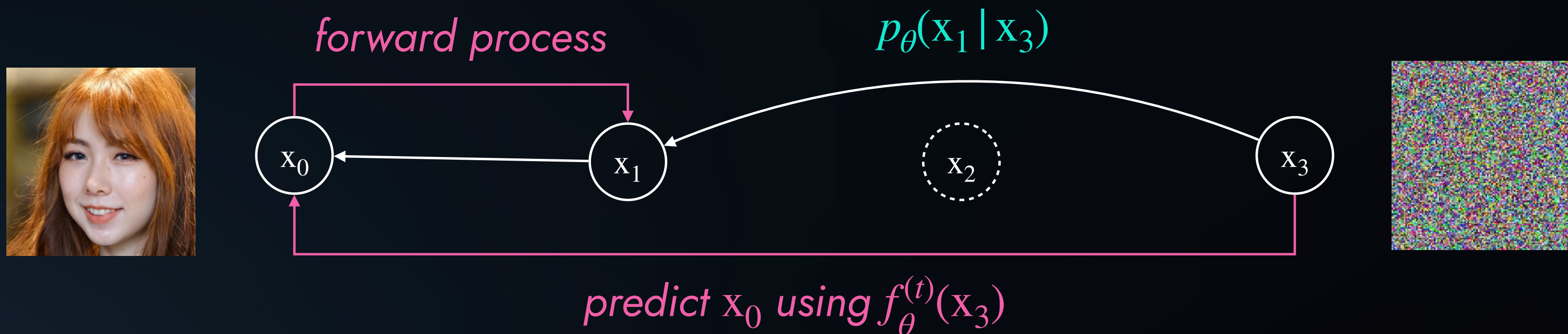
Implicit probabilistic model

- *Denoising diffusion implicit models (DDIM)* can produce the high-quality samples much faster (~50x), empirically (Song et al., 2021).
- A short explanation is we can take a “*shortcut*” for multiple steps.
- There is the trade-off between *sample quality* and *training speed*.



Long explanation w/o equations

- If we have access to the original image x_0 , we can easily get x_{t-1} since from x_0 to x_{t-1} is the forward process (adding Gaussian noises) which is analytically calculated (Song et al., 2021).
- So, the problem is changed from predicting previous state x_{t-1} to *predicting the denoised observation x_0 from the current state x_t .*



Flow matching

- **Probability path**

- Interpolate linearly between a data sample and a noise sample.

$$\mathbf{x}_t = (1 - t)\mathbf{x}_0 + t\epsilon, \quad \mathbf{x}_0 \sim p_{\text{data}}, \quad \epsilon \sim \mathcal{N}(0, I)$$

- **Training:** regress the velocity

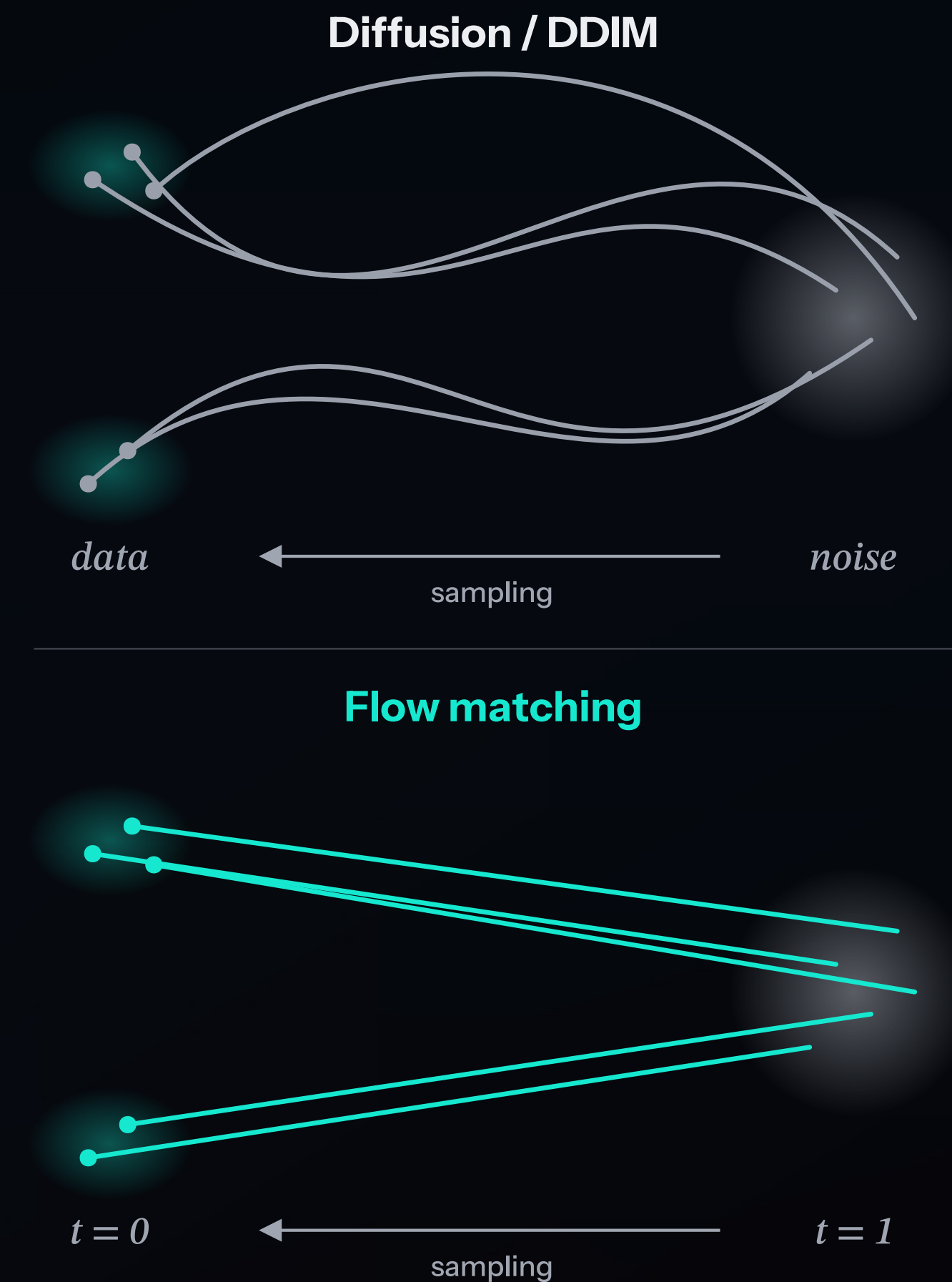
- Same recipe as DDPM, but the target is the velocity $\epsilon - \mathbf{x}_0$ instead of the noise ϵ .

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\epsilon - \mathbf{x}_0)\|^2$$

- **Sampling:** integrate an ODE from $t = 1$ to 0

- Deterministic like DDIM; *straighter* paths need fewer Euler steps.

$$\mathbf{x}_{t-\Delta t} = \mathbf{x}_t - \Delta t \cdot \mathbf{v}_\theta(\mathbf{x}_t, t), \quad \mathbf{x}_1 = \epsilon$$



Variational Diffusion Model

A simplified explanation omitting noise schedule

Evidence lower bound (ELBO)

- The Evidence Lower Bound (ELBO) is a concept from Bayesian machine learning, particularly in Variational Inference. In simple terms:
- It's a way to approximate the log evidence (or log-likelihood) of observed data x when directly computing it (especially the marginal distribution) is intractable, *because the marginal distribution $p(x) = \int p(x | z) p(z) dz$ requires integrating over all latent variables z .*

$$\begin{aligned}\log p(x) &\geq \underbrace{\mathbb{E}_{q(z)}[\log p(x, z)] - \mathbb{E}_{q(z)}[\log q(z)]}_{ELBO} \\ &= \underbrace{\mathbb{E}_{q(z)}[\log p(x | z)]}_{Reconstruction} - \underbrace{\text{KL}[q(z) \parallel p(z)]}_{Prior}\end{aligned}$$

Proof (1/2)

- Start with the log marginal likelihood:

$$\begin{aligned}\log p(x) &= \log \int p(x, z) dz && \text{Marginal distribution w.r.t. a new variable } z \\ &= \log \int q(z) \frac{p(x, z)}{q(z)} dz. && \text{Introduce a tractable distribution } q(z)\end{aligned}$$

- Apply *Jensen's inequality* since the logarithm is concave:

$$\begin{aligned}\log p(x) &\geq \int q(z) \log \frac{p(x, z)}{q(z)} dz && \text{Jensen's inequality} \\ &= \mathbb{E}_{q(z)}[\log p(x, z)] - \mathbb{E}_{q(z)}[\log q(z)].\end{aligned}$$

Proof (2/2)

- Apply *Jensen's inequality* since the logarithm is concave (restated):

$$\begin{aligned}\log p(x) &\geq \int q(z) \log \frac{p(x, z)}{q(z)} dz && \text{Jensen's inequality} \\ &= \mathbb{E}_{q(z)}[\log p(x, z)] - \mathbb{E}_{q(z)}[\log q(z)] && \text{From the previous} \\ &= \mathbb{E}_{q(z)}[\log p(x | z)p(z)] - \mathbb{E}_{q(z)}[\log q(z)] \\ &= \mathbb{E}_{q(z)}[\log p(x | z)] - \mathbb{E}_{q(z)}[\log q(z)/p(z)] \\ &= \underbrace{\mathbb{E}_{q(z)}[\log p(x | z)]}_{\text{Reconstruction}} - \underbrace{\mathbb{E}_{q(z)}[\log q(z)/p(z)]}_{\text{Prior}}\end{aligned}$$

- *Reconstruction* term measures how well z explains the observed data x .
- *Prior* term encourages it to stay close to the prior $p(z)$.

Forward time diffusion process

- A Gaussian diffusion process begins with the data $\mathbf{x}$, and defines a sequence of the latent variables $\mathbf{z}_t$, increasingly noisy versions of $\mathbf{x}$, from $\mathbf{z}_0$ to $\mathbf{z}_1$.
- The distribution of latent variable $\mathbf{z}_t$ conditioned on $\mathbf{x}$, for any $t \in [0, 1]$ is given by:

$$q(\mathbf{z}_t | \mathbf{x}) = \mathcal{N}(\alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I})$$

where α_t and σ_t^2 are positive scalar functions of t , having *variance-preserving*.

- Note that $q(\mathbf{z}_t | \mathbf{z}_s)$ for any $0 \leq s < t \leq 1$ is Gaussian, and $q(\mathbf{z}_s | \mathbf{z}_t, \mathbf{x})$ is also Gaussian under Markov assumption and Bayes rule (ref. Appendix A of [Kingma et al., 2021](#)).
- *The noise schedule is out-of-scope in this talk* (ref. Sec. 3.2 of [Kingma et al., 2021](#)).

Reverse time generative model

- A generative model for the data $\mathbf{x}$, using $s(i) := (i - 1)/T$, $t(i) := i/T$ is given by:

$$p(\mathbf{x}) = \int_{\mathbf{z}} p(\mathbf{z}_1) p(\mathbf{x} \mid \mathbf{z}_0) \prod_{i=1}^T p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}).$$

- Due to our noise schedule, $p(\mathbf{z}_1) = \mathcal{N}(\mathbf{z}_1; 0, \mathbf{I})$ and $p(\mathbf{x} \mid \mathbf{z}_0) \approx 1$.
- Finally, we *choose* the conditional model distributions as:

$$p(\mathbf{z}_s \mid \mathbf{z}_t) = q(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{x} = \hat{\mathbf{x}}_{\theta}(\mathbf{z}_t; t))$$

where $\hat{\mathbf{x}}_{\theta}(\mathbf{z}_t; t) = (\mathbf{z}_t - \sigma_t \hat{\epsilon}_{\theta}(\mathbf{z}_t; t)) / \alpha_t$. In this way, we connect the reverse model p to the forward model q .

Variational lower bound for diffusion models

- Kingma et al. (2021) suggested the variational lower bound (VLB) of the marginal likelihood, which is as follows:

$$-\log p(x) \leq \underbrace{\text{KL}(q(z_1 | x) || p(z_1))}_{\text{Prior}} + \underbrace{\mathbb{E}_{q(z_0|x)}[-\log p(x | z_0)]}_{\text{Reconstruction}} + \underbrace{\mathcal{L}_T(x)}_{\text{Diffusion}}$$

where z_0 is an image, z_1 is noise, and it assumes a *Markov chain in diffusion process*.

- *Reconstruction* term measures how well z_0 explains the observed data x .
- *Prior* term encourages it to stay close to the prior $p(z_1)$ of Gaussian distribution.
- Notice that *Prior* and *Reconstruction* terms are negligible in practice.

Diffusion loss

- The diffusion loss is defined as follows:

$$\mathcal{L}_T(\mathbf{x}) = \sum_{i=1}^T \mathbb{E}_{q(\mathbf{z}_{t(i)}|\mathbf{x})} \text{KL} \left[q(\mathbf{z}_{s(i)} | \mathbf{z}_{t(i)}, \mathbf{x}) \| p(\mathbf{z}_{s(i)} | \mathbf{z}_{t(i)}) \right].$$

- Remind that $q(\mathbf{z}_{s(i)} | \mathbf{z}_{t(i)}, \mathbf{x} = \hat{\mathbf{x}}_{\theta}(\mathbf{z}_{t(i)}; t(i)))$, where the original data $\mathbf{x}$ is replaced by the output of a denoising model θ predicting $\mathbf{x}$ from its noisy version $\mathbf{z}_t$.
- $q(\mathbf{z}_{s(i)} | \mathbf{z}_{t(i)}, \mathbf{x})$ and $p(\mathbf{z}_{s(i)} | \mathbf{z}_{t(i)})$ are Gaussians.
- So, the KL term has a closed-form solution.

KL divergence of two Gaussians

- Suppose that two Gaussian distributions $\mathcal{N}_0, \mathcal{N}_1$ have means μ_0, μ_1 , and covariances Σ_0, Σ_1 . Then, the KL divergence between $\mathcal{N}_0$ and $\mathcal{N}_1$ is as follows:

$$D_{KL}(\mathcal{N}_0 \parallel \mathcal{N}_1) = \frac{1}{2} \left[\text{tr}(\Sigma_1^{-1} \Sigma_0) + (\mu_1 - \mu_0)^\top \Sigma_1^{-1} (\mu_1 - \mu_0) - k + \ln \left(\frac{\det \Sigma_1}{\det \Sigma_0} \right) \right].$$

- If the covariances are the same *isotropic* Gaussian distribution, *i.e.*, $c\mathbf{I}$, the above equation is further simplified as follows:

$$D_{KL}(\mathcal{N}_0 \parallel \mathcal{N}_1) = \frac{1}{2c} \|\mu_1 - \mu_0\|^2.$$

Monte Carlo estimator

- Using the previous solution, the diffusion loss is considerably simplified as follows:

$$\mathcal{L}_T(\mathbf{x}) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I}), i \sim U\{1, T\}} \left[w(t) \|\mathbf{x} - \hat{\mathbf{x}}_{\theta}(\mathbf{z}_t; t)\|_2^2 \right]$$

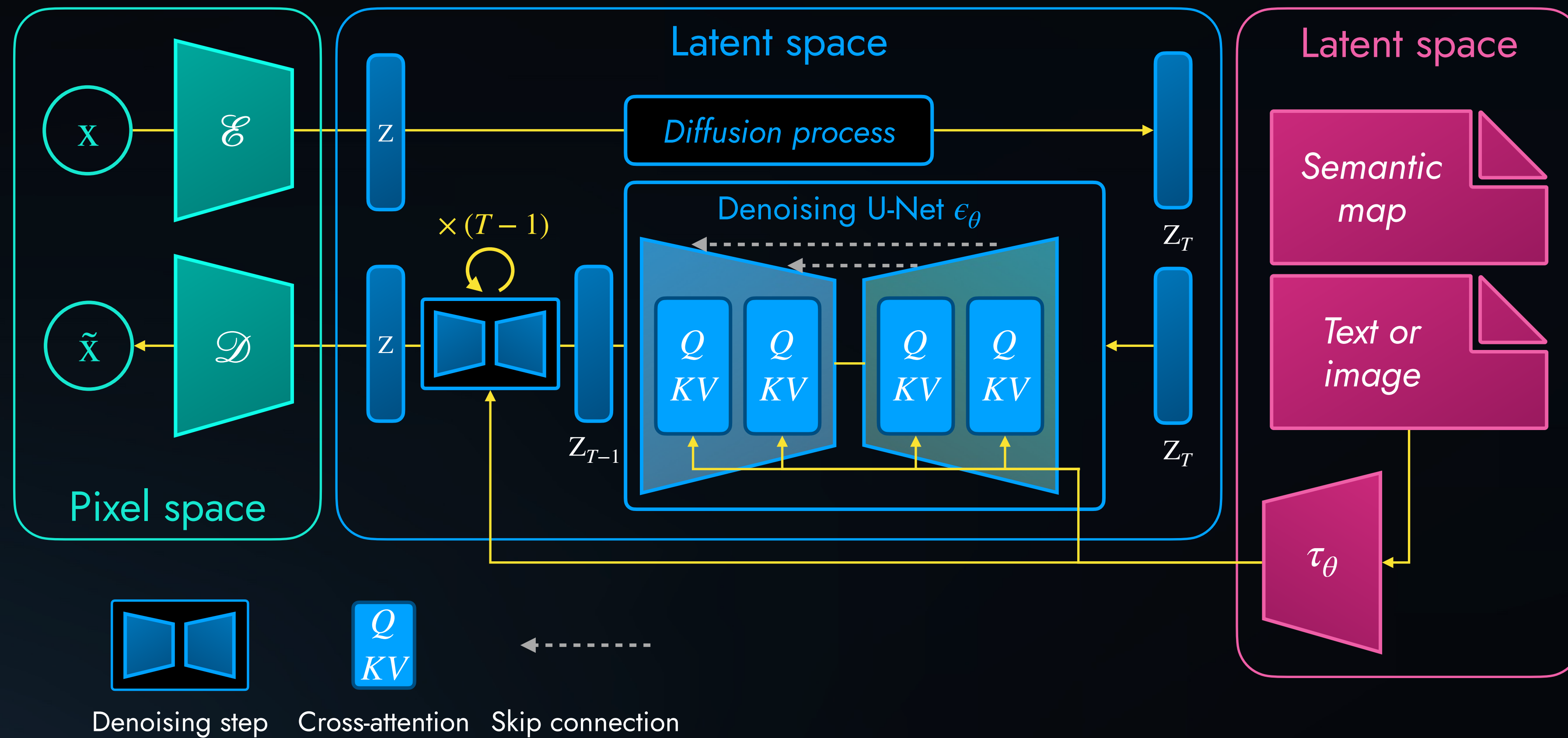
where we reparameterize $\mathbf{z}_t \sim q(\mathbf{z}_t | \mathbf{x})$ as $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, use a Monte Carlo sampling, and $w(t)$ is a scalar-valued function of t .

- In terms of the noise prediction model,

$$\mathcal{L}_T(\mathbf{x}) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I}), i \sim U\{1, T\}} \left[w'(t) \|\epsilon - \hat{\epsilon}_{\theta}(\mathbf{z}'_t; t)\|_2^2 \right]$$

where $w'(t)$ is another scalar-valued function of t and $\mathbf{z}'_t = \alpha'_t \mathbf{x} + \sigma'_t \epsilon$. For simplicity, we omit further details of the noise schedule w.r.t. α'_t and σ'_t .

Latent diffusion model



Score Distillation Sampling



Definitions

- The previous diffusion loss for parameters ϕ is defined as follows:

$$\mathcal{L}_{Diff}(\phi, \mathbf{x}) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I}), i \sim U\{0, T\}} \left[w(t) \|\epsilon - \hat{\epsilon}_{\phi}(\mathbf{z}'_t; t)\|_2^2 \right].$$

- The *score function* is given by $s_{\phi}(\mathbf{z}_t; t) = -\epsilon_{\phi}(\mathbf{z}_t; t)/\sigma_t$ where a scalar σ_t is chosen for variance preservation (Kinngma et al., 2021; Song et al., 2021).
- The *classifier-free guidance* (CFG, Ho & Salimans, 2022) introduces a guidance scale parameter ω as follows:

$$\hat{\epsilon}_{\phi}(\mathbf{z}_t; y, t) = (1 + \omega)\epsilon_{\phi}(\mathbf{z}_t; y, t) - \omega\epsilon_{\phi}(\mathbf{z}_t; t)$$

where y is a conditioning text embedding for a text-to-image generation task.

Deep image prior

- What if the data x is a generated datapoint by $x = g(\theta)$, *i.e.*, the rendered image from a NeRF in the DreamFusion (Poole et al., 2022)?

$$\theta^* = \operatorname{argmin}_{\theta} \mathcal{L}_{Diff}(\phi, x = g(\theta))$$

- This differentiable image parameterization (Mordvintsev et al., 2018) or deep image prior (Ulyanov et al., 2018) usually does not produce realistic samples, even when using an identity DIP where $x = \theta$.
- The gradient analysis may give you a hint.

U-net Jacobian is minor

- The gradient of $\mathcal{L}_{Diff}$ is as follows:

$$\nabla_{\theta} \mathcal{L}_{Diff}(\phi, \mathbf{x} = g(\theta)) = \mathbb{E}_{t, \epsilon} \left[\underbrace{w(t) (\hat{\epsilon}_{\phi}(\mathbf{z}_t; y, t) - \epsilon)}_{\text{Noise residual}} \underbrace{\frac{\partial \hat{\epsilon}_{\phi}(\mathbf{z}_t; y, t)}{\partial \mathbf{z}_t}}_{\text{U-net}} \underbrace{\frac{\partial \mathbf{z}_t}{\partial \mathbf{x}}}_{\alpha_t \mathbf{I}} \underbrace{\frac{\partial \mathbf{x}}{\partial \theta}}_{\text{Generator}} \right].$$

- The term $\partial \mathbf{z}_t / \partial \mathbf{x} = \alpha_t \mathbf{I}$ can be integrated into $w(t)$.
- The U-net Jacobian term is expensive to compute and “poorly conditioned for small noise levels as it is trained to approximate the scaled Hessian of the marginal density.”
- The noise ϵ follows an **isotropic** Gaussian distribution, which is approximated by $\hat{\epsilon}_{\phi}$. The optimization may not heavily rely on the fine-grained structure of the Jacobian.

Score distillation sampling

- Backpropagating through the U-Net is expensive, and its Jacobian is poorly conditioned at small noise levels. Omitting it works well in practice.
- So, we simply omit the U-net Jacobian term as follows:

$$\nabla_{\theta} \mathcal{L}_{SDS}(\phi, \mathbf{x} = g(\theta)) := \mathbb{E}_{t, \epsilon} \left[w'(t) (\hat{\epsilon}_{\phi}(z_t; y, t) - \epsilon) \frac{\partial \mathbf{x}}{\partial \theta} \right].$$

- While it may appear ad-hoc, one can show that it is an (approximated) gradient of a weighted probability density distillation loss* (Oord et al., 2018) as follows:

$$\nabla_{\theta} \mathcal{L}_{SDS}(\phi, \mathbf{x} = g(\theta)) = \nabla_{\theta} \mathbb{E}_t \left[\sigma_t / \alpha_t w'(t) \text{KL} [q(z_t | g(\theta); y, t) \| p_{\phi}(z_t; y, t)] \right].$$

- For a fixed ϕ , the loss is minimized when $g(\theta)$ lies in a high-density region of p_{ϕ} . This *mode-seeking behavior* partly explains SDS's over-saturated, low-diversity results.

*Unlike the ELBO's diffusion loss, which compares per-step transitions from $t(i)$ to $s(i)$, it compares the marginals $q(z_t | x)$ and $p_{\phi}(z_t)$ at t .

Gaussian noise prediction

- Remarkably, $\partial \hat{\epsilon}_\phi / \partial z_t$ does not appear: the network directly outputs the score $s_\phi(z_t) \approx \nabla_{z_t} \log p_\phi(z_t)$, so we never differentiate through it.
- Let's see the gradient of the second term* of the KL divergence as follows:

$$\begin{aligned}\nabla_{z_t} \log p_\phi(z_t) &= \nabla_{z_t} \log p(z_t) = \frac{\alpha_t \mathbb{E}[x | z_t] - z_t}{\sigma_t^2} && \text{by Tweedie's formula} \\ &\approx - (z_t - \alpha_t \hat{x}_\phi(z_t; t)) / \sigma_t^2 && \text{by a noise prediction model} \\ &= - (\cancel{\alpha_t \hat{x}_\phi(z_t; t)} + \sigma_t \hat{\epsilon}_\phi(z_t; t) - \cancel{\alpha_t \hat{x}_\phi(z_t; t)}) / \sigma_t^2 \\ &= - \hat{\epsilon}_\phi(z_t; t) / \sigma_t \\ &= s_\phi(z_t),\end{aligned}$$

where s_ϕ is obtained from the pretrained noise predictor ϕ via *Tweedie's formula*.

*The second term of $\text{KL}[q(z_t | g(\theta)) \| p_\phi(z_t)] = \mathbb{E}_{q(z_t | g(\theta))} [\log q(z_t | g(\theta)) - \log p_\phi(z_t)]$.

Tweedie's formula

- Marginal is the mixture of Gaussian bumps on data points:

$$p(z) = \int q(z | x) p(x) dx, \quad q(z | x) = \mathcal{N}(\alpha x, \sigma^2 I)$$

- **Step 1.** Since the gradient of the Gaussian is $\nabla_z q(z | x) = q(z | x) \cdot \frac{\alpha x - z}{\sigma^2}$, so we have:

$$\nabla_z p(z) = \int q(z | x) \frac{\alpha x - z}{\sigma^2} p(x) dx$$

- **Step 2.** Divide both sides by $p(z)$. By Bayes rule,

$$\nabla_z \log p(z) = \int \frac{\alpha x - z}{\sigma^2} p(x | z) dx$$

- **Step 3.** Take the expectation. Using linearity, we have *Tweedie's formula*:

$$\nabla_z \log p(z) = \frac{\alpha \mathbb{E}[x | z] - z}{\sigma^2}.$$

Gaussian noise prediction (cont'd)

- By the chain rule, relying on $s_\phi(z_t) \approx \nabla_{z_t} \log p_\phi(z_t)$, and $x = g(\theta)$, $z_t = \alpha_t x + \sigma_t \epsilon$:

$$\nabla_\theta \log p_\phi(z_t) = \nabla_{z_t} \log p_\phi(z_t) \frac{\partial z_t}{\partial \theta} \approx s_\phi(z_t) \frac{\partial z_t}{\partial \theta} = \alpha_t s_\phi(z_t) \frac{\partial x}{\partial \theta} = -\frac{\alpha_t}{\sigma_t} \hat{\epsilon}_\phi(z_t) \frac{\partial x}{\partial \theta}.$$

- Using the path derivative only (the zero-mean score term is dropped, cf. Roeder et al., 2017):

Left term $\nabla_\theta \log q(z_t | x) \rightarrow -\frac{\alpha_t}{\sigma_t} \epsilon \frac{\partial x}{\partial \theta}$. Since $\mathbb{E}_\epsilon[\epsilon] = 0$, the ϵ term acts as a control variate that reduces variance without changing the expected gradient. Therefore, we can rewrite as follows:

$$\begin{aligned} \nabla_\theta \mathcal{L}_{SDS}(\phi, x = g(\theta)) &= \nabla_\theta \mathbb{E}_{t,\epsilon} \left[\sigma_t / \alpha_t w'(t) \text{KL}[q(z_t | g(\theta); y, t) \parallel p_\phi(z_t; y, t)] \right] \\ &= \mathbb{E}_{t,\epsilon} \left[\sigma_t / \alpha_t w'(t) \left(\nabla_\theta \log q(z_t | g(\theta); y, t) - \nabla_\theta \log p_\phi(z_t; y, t) \right) \right] \\ &= \mathbb{E}_{t,\epsilon} \left[w'(t) \left(\hat{\epsilon}_\phi(z_t; y, t) - \epsilon \right) \frac{\partial x}{\partial \theta} \right]. \end{aligned}$$

Case study: 3DFuse

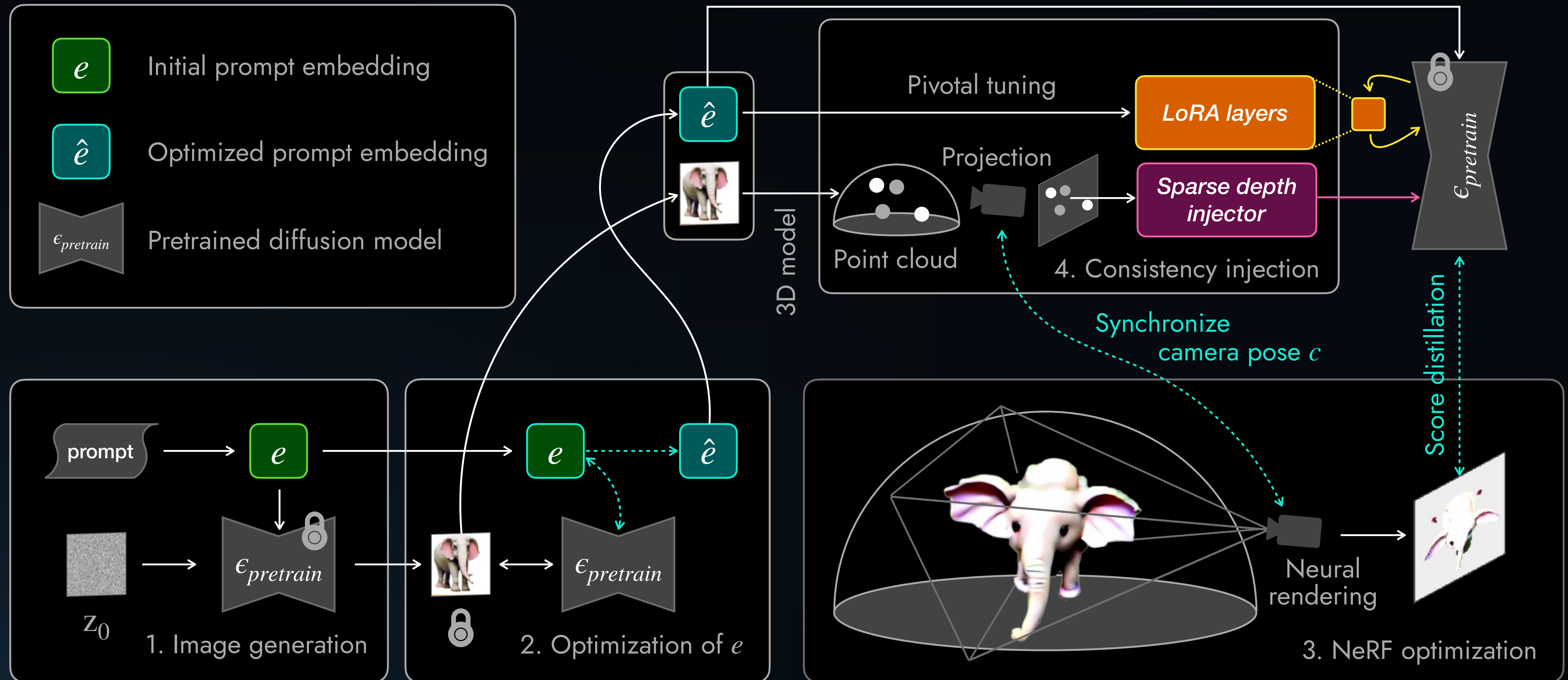
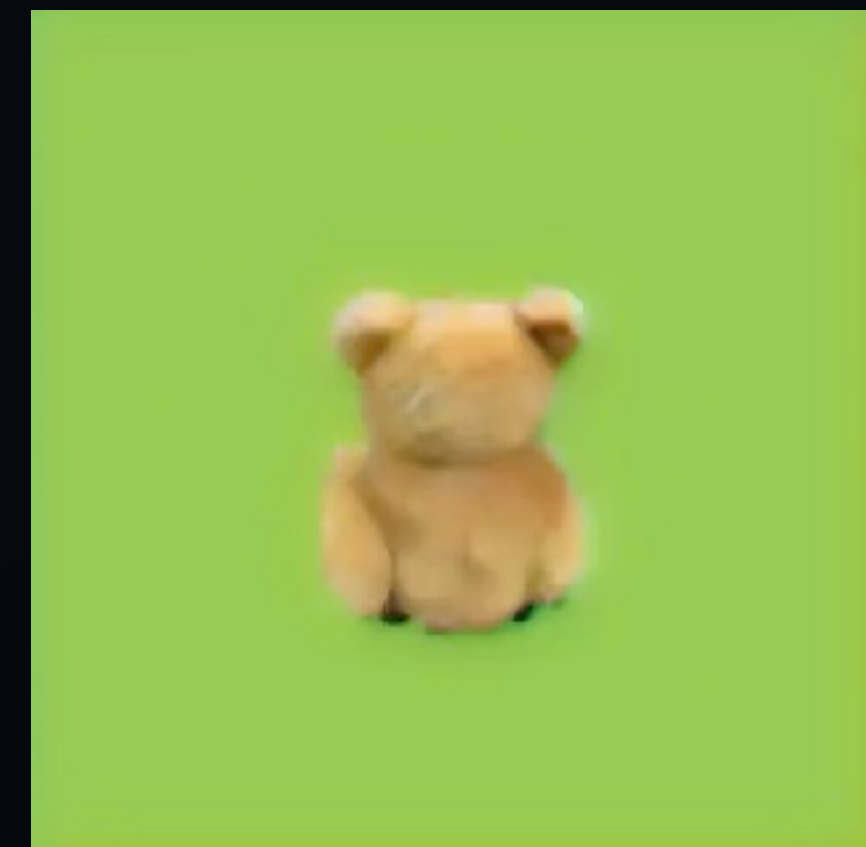


Image-conditional Generation

- Since text-to-3D generation is inherently under-constrained, it is common practice to reformulate the task by leveraging pretrained text-to-image models as a prior.
- Following this line of work, we instead directly provide an input image and point cloud geometry, which effectively reconfigures our framework into an image-conditional setting.



What are World Models?

Craik's insight

- Kenneth Craik, The Nature of Explanation (Craik, 1943)
 - *"If the organism carries a small-scale model of external reality and of its own possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise."*
- The brain *simulates* before it acts.
 - No need to physically explore every possibility
- Neural evidences (Nobel prize 2014)
 - *Grid cells* (entorhinal cortex): internal coordinate system for space
 - *Place cells* (hippocampus): fire at a specific location, "where am I"
 - During sleep, hippocampus *replays* experiences: updating the world model offline



Kenneth Craik
(1914–1945)

Why “World Models” become a buzzword?

Classic usage in RL

- Notably, World Models (Ha & Schmidhuber, 2018) predicts a next latent frame.
- Trained *entirely* inside the dream without high-cost environment interaction

Modern usage (2023~)

- Models that understands how the world works
- LLMs, video generator, foundation models

Three different bets:

- 1) *RL camp*: explicit latent-space simulator
- 2) *LeCun camp*: joint embedding & prediction
- 3) *Video camp*: generative pixel simulator



The agent learning to navigate a racing environment (Ha & Schmidhuber, 2018)

***RL Camp: Dreamer,
Planning by Imagination***

Dream, then act

- *Dreamer* by Google DeepMind (Hafner et al., 2020)
- Recurrent State Space Model (RSSM)
 - Encodes observations into latent state z_t
 - Learns the **transition** $p(z_{t+1} | z_t, a_t)$ in a latent space
 - Decodes back to the observation space for a training signal
- *Policy is trained purely in imagination*
 - Rollout H -step trajectories inside the model
- *Sample efficiency*
 - Orders of magnitude fewer real interactions than model-free RL

Three-Component Loop

Let z_t , x_t , and a_t be latent state, observation, and action, respectively.

Representation model (posterior)

$$q_\theta(z_t | z_{t-1}, a_{t-1}, x_t)$$

Transition model (prior, no x_t)

$$p_\theta(z_t | z_{t-1}, a_{t-1})$$

Policy & value (actor-critic)

$$\pi_\phi(a_t | z_t), \quad V_\psi(z_t)$$

where the *actor* a_t decides what to do, the *critic* V_ψ estimates the expected value.

Dreamer series: 5 years' progress

DreamerV1 (Hafner et al., 2020)

- DMControl & Atari
- Latent imagination within actor-critic rollout
- Competitive with model-free at 5x fewer steps

DreamerV2 (Hafner et al., 2021)

- Discrete latents via straight-through gradients
- Matches Rainbow on Atari with 200x less data

DreamerV3 (Hafner et al., 2025a)

- 150+ *diverse tasks* with one fixed hyperparameter set
- Fixed KL balancing, symlog predictions
- First to collect Minecraft diamonds from scratch
- Published in *Nature 2025*

DreamerV4 (Hafner et al., 2025b)

- Minecraft diamonds from offline data only
- No real environment interaction during training
- World model as a *data-efficient* simulator

Why sample efficiency matters?

RL bottleneck

- Model-free RL learns only from real interactions
 - Robot falls over: a huge cost in the real world
- Millions of steps needed for a simple task
 - e.g., Atari needs ~200M frames, 38 days of playing

Actor-Critic (AC) in brief

- Critic provides a *bootstrap signal*:
 - No need to wait until episode ends
- *Lower variance* than pure policy gradient (model-free RL)

Sample Efficiency

The same performance. Real interaction cost cut by 5x by training Actor-Critic *inside the dream* not in the real environment.

In robotics fewer falls, less wear, and faster iteration.

But doesn't imagination compound errors?

Model bias problem

- *World Model* is an approximation.
- Over H imagined steps, errors accumulate.
- Actor can exploit model flaws:
 - the policy that scores high in the dream may fail in reality

How Dreamer limits the damage

- Short horizon $H \approx 15$
- Critic bootstrapping: $V(z_H)$ terminates the rollout
 - Long-range error propagation cut off

Bias-variance trade-off

Method	Variance	Bias
Model-free (MC)	High	None
Model-free (AC)	Medium	None
Dreamer	Low	Present

Dreamer's Bet

In structured environments, *bias from model errors* is smaller than *variance from real rollouts*.

DreamerV3/V4 continue to address this: V4's shift to offline data *reduces model exploitation*.

Variance of policy gradient

High Variance

Model-free (MC)

- REINFORCE (Williams, 1992)
- A score-function estimator
$$\nabla_{\phi} \mathbb{E}[R(\tau)] = \mathbb{E}[\mathbf{R}(\tau) \cdot \nabla_{\phi} \log \pi_{\phi}(a | s)]$$
- The scale and variance of the reward R multiplicatively impacts on the policy gradient.

Medium Variance

Model-free (AC)

- Baseline + TD bootstrap
$$\nabla_{\phi} \mathbb{E}[R(\tau)] = \mathbb{E}[(r_t + \gamma V(s_{t+1}) - V(s_t)) \cdot \nabla_{\phi} \log \pi_{\phi}(a | s)]$$
- A **single-step bootstrap-reduced variance** instead of full trajectory's.
- However, the noise source $\nabla_{\theta} \log \pi_{\theta}(a | s)$ remains.

Low Variance

Dreamer (Hafner et al., 2020)

- Latent imagination
- Path-wise gradient:
$$\nabla_{\phi} \mathbb{E}_{s_{\tau+1} \sim q_{\theta}, a_{\tau} \sim q_{\phi}} \left[\sum_{\tau=t}^{t+H} V^{\lambda}(s_{\tau}) \right]$$
where V^{λ} is λ -return (bootstrapped with critic v_{ψ}).
- Low variance with **analytic error back-propagation** for ϕ

*LeCun Camp: JEPA,
Beyond Next-Token Prediction*

LeCun's critique of LLMs as World Models

- “A path Towards Autonomous Machine Intelligence” (LeCun, 2022)
- Autoregressive LLMs are fundamentally limited:
 - Next-token prediction in pixel space is too low-level
 - Forces the model to represent every detail
 - Cannot represent *uncertainty* over multi-modal futures
- Physical commonsense requires *abstract* prediction
 - “The cup will fall.” — not a pixel-by-pixel forecast
 - Humans predict at the level of concepts, not sensory details



<https://ai.meta.com/research/vjepa/>

👉 Generative prediction $\neq$ understanding

Joint Embedding Predictive Architecture (JEPA)

- Encoder maps $x \rightarrow s_x$, context encoder maps $y \rightarrow s_y$
- Predictor maps $s_x \rightarrow \hat{s}_y$ using latent target
 - Predictions in the representation space
 - No decoder, no reconstruction loss
- Training objective: minimize $\|\hat{s}_y - \text{sg}(s_y)\|^2$
 - Stop-gradient on target to prevent collapse
 - EMA target encoder (à la BYOL, DINO)
- V-JEPA 2 (Assran et al., 2025)
 - Train 1M hours of internet video for robotics
 - Action-conditioned prediction using: $\hat{s}_{t+1} = f(s_t, a_t)$

Double-edged Sword

Abstract prediction over generation for commonsense and planning. However, V-JEPA 2.1 (Mur-Labadia et al., 2026) adds DINOv2-style dense supervision; pure abstraction may not suffice.

Autoregressive vs. JEPA

Autoregressive

- Predict next token and frame explicitly
- Must generate all sensory details
- Uncertainty = sample from distribution
- **Pro**: flexible, open-ended generation
- **Con**: expensive, hallucinates details

LLM Camp Claim

"GPT-4 or Gemini already has world knowledge — Scale is enough."

JEPA

- Abstract future-state prediction
- Ignores irrelevant detail by design
- Uncertainty encoded in latent geometry
- **Pro**: efficient, abstract, grounded
- **Con**: harder to evaluate, no generation

LeCun Camp Claim

"Tokens will never give you physical intuition. You need a different architecture."

Video Camp:
Generative Simulator

Are video generations World Models?

- **Sora** (OpenAI, 2024)

- “video generation as world simulators”
 - Trained on massive internet video
 - Generates long, physically plausible videos
- A powerful video predictor implicitly learns physics and commonsense
 - Evidence for:
 - lighting, consistency, continuation after occlusion
 - Evidence against:
 - Cup tilted sideways, water stays
 - Count errors, non-physical motion

- **Google Genie** (Bruce et al., 2024)

- Interactive action-conditioned
- *Unsupervised action discovery*

- **NVIDIA Cosmos** (Agarwal et al., 2025)

- World Foundation Models for robotics and autonomous vehicles
- Large-scale video DiT

Three camps: A summary

RL Camp (Dreamer)

- Latent dynamics model
- Trained end-to-end for RL
- Strong on control tasks
- Limited to environments similar to training

💡 *“Compact Latent state is enough for planing.”*

LeCun Camp (JEPA)

- Non-generative, joint embedding
- Self-supervised from videos
- Abstract commonsense
- Harder to evaluate the latent directly

💡 *“Abstraction is necessary for general intelligence.”*

Video Camp (Sora, Genie, Cosmos)

- Pixel-level generation
- Internet-scale training
- Rich, controllable outputs
- Physics: poorly understood

💡 *“Scale + generation = implicit world knowledge!”*

Open Questions

What makes a World Model “Good”?

- Evaluation is hard: no universal benchmark
 - ICML 2025 workshop: “Assessing World Models”
 - CVPR 2025 workshop: “WorldModelBench”
 - CVPR 2026 workshop: “Video World Models”
 - ICLR 2026 workshop: “World Models”
- Desiderata from workshop discussions:
 1. **Predictive accuracy**: does $\hat{s}_{t+1}$ match s_{t+1} ?
 2. **Causal fidelity**: a_t produces the right effect?
 3. **Compositionality**: can it generalize to novel?
 4. **Long-horizontal stability**: does error compound?
 5. **Downstream utility**: does it improve agents?

Evaluation landscape

- **RL bench**: DMControl, Atari 100k, Minecraft Diamond
- **Physics**: IntPhys, PHYRE, PhyGenBench
- **Causal**: CLEVRER
- **Video pred**: FVD, PSNR, SSIM
- **Robotics**: RLBench, Open X-Embodiment

Deeper Question:

Does the model understand physics, or does it just look like it does?

2026 World Model workshops

CVPR 2026 (Denver in June)

- Video World Models: interaction, memory & efficiency
- 4D World Models: bridging generation & reconstruction
- World Models Meet Active Sensing & Closed-Loop Planing
- GigaBrain Challenge: World Models for VLA
- Tutorial: From Perception to Simulation

ICML 2026 (Seoul in July)

- From Frames to Stories: Toward Reliable Long-Horizon Video Generation

ECCV 2026 (Malmö in September)

- Safe World Models for Trustworthy Embodied AI
- Human Motion-Informed World Models
- World Models in the Loop
- 3D in the Era of World Models
- How to Build Effective World Models for Embodied AI
- On-device Embodied World Models

Physics grounding & causal structure

- **Correlation vs. causation**: the core unsolved problem
 - Internet video captures *correlations* of physics, not physics itself
 - A model that interpolates well can still fail on interventions
- **Elias Bareinboim** (NeurIPS'25w keynote speaker)
 - “World models need explicit causal structure.”
 - Causal graphs enable do-calculus interventions, not just observational predictions
 - Pearl’s ladder (Pearl et al., 2018): **Association** → **Intervention** → **Counterfactual**
- Current models are largely at level 1 (association)



Demis Hassabis at Lex Fridman (2025)

“A world model, a model of how the world works, the mechanics of the world, the **physics of the world** ...”

World Models in Physical Space

Beyond simulation

- World models so far, simulated environment
- **New frontier**: anchor the model to a real, GPS-tagged location (Seo et al., 2026)
 - Street Views as geometric scaffolding
 - Kilometer-scale trajectory video generation

The Street View moment in 2026

- **Seoul World Model** (Mar): seoul-scale world model, trajectory-faithful driving simulation
- CityRAG (Apr): multi-city RAG over Google Street View clips of 10 cities (Chou et al., 2026)
- Genie, Google I/O (May): Street View pin → interactive world

Why physical grounding matters?

- Autonomous Vehicle: photorealistic in real-world
- Digital twins: city-scale AR/VR, planning
- Rare events: simulate accidents, extreme weather
- New eval axis: geographic & trajectory fidelity

The Key Shift

World models are leaving the sandbox: Physical space, real coordinates and topology become the training and evaluation distribution.

Key open problems

Research frontiers

- **Scalable evaluation** beyond exhaustive labels
- **Unifying RL + Generation.** The same model for planing and synthesis
- **Multi-agent worlds.** Other agents as part of the modeled environment.
- **Embodiment gap.** *Sim-to-real* transfer for robot world models.

The Bit bets (2025~2027)

- **LeCun said**, “LLMs will become obsolete within five years. What we need is a world model.”
 - Would JEPA replace LLMs?
- DeepMind’s Genie 3 as a world simulator
- NVIDIA Cosmos ([Agarwal et al., 2025](#)) as physical world foundation model
- OpenAI’s Sora 2 with interactive control

Take-aways

1. World models learned simulators of state transitions
 - Not a single architecture, but they predict what happens next.
2. Three camps, three different bets
 - **RL**: latent-space planing works on hard tasks
 - **JEPA**: generation is wrong loss, abstract prediction is better.
 - **Video**: scale the simulator, understanding will emerge.
3. Evaluation is the bottleneck!
 - Downstream agent performance is the gold standard.
 - But physics/causal understanding remains hard to measure directly.



Genie 2 Lighting

Seoul World Model

Grounding World Simulation Models in a Real-World Metropolis

Junyoung Seol¹, Hyunwook Cho^{1*}, Minkyung Kwon¹, Jinhyeok Cho¹, Siyoon Jil¹, Gayoung Lee¹, Junho Kim²,
Joungbin Lee¹, Geonmo Gu¹, Dongyoon Han², Sangdoo Yoo², Seungryong Kim^{†1}, Jin-Hwa Kim^{†2,3}

KAIST AI

NAVER AI Lab

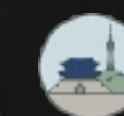
SNU AIIS

*Equal contribution

Corresponding authors



*The first author conducted this work as part of the NAVER Cloud Residency Program.



seoul-world-model.github.io

SWM gains global attention

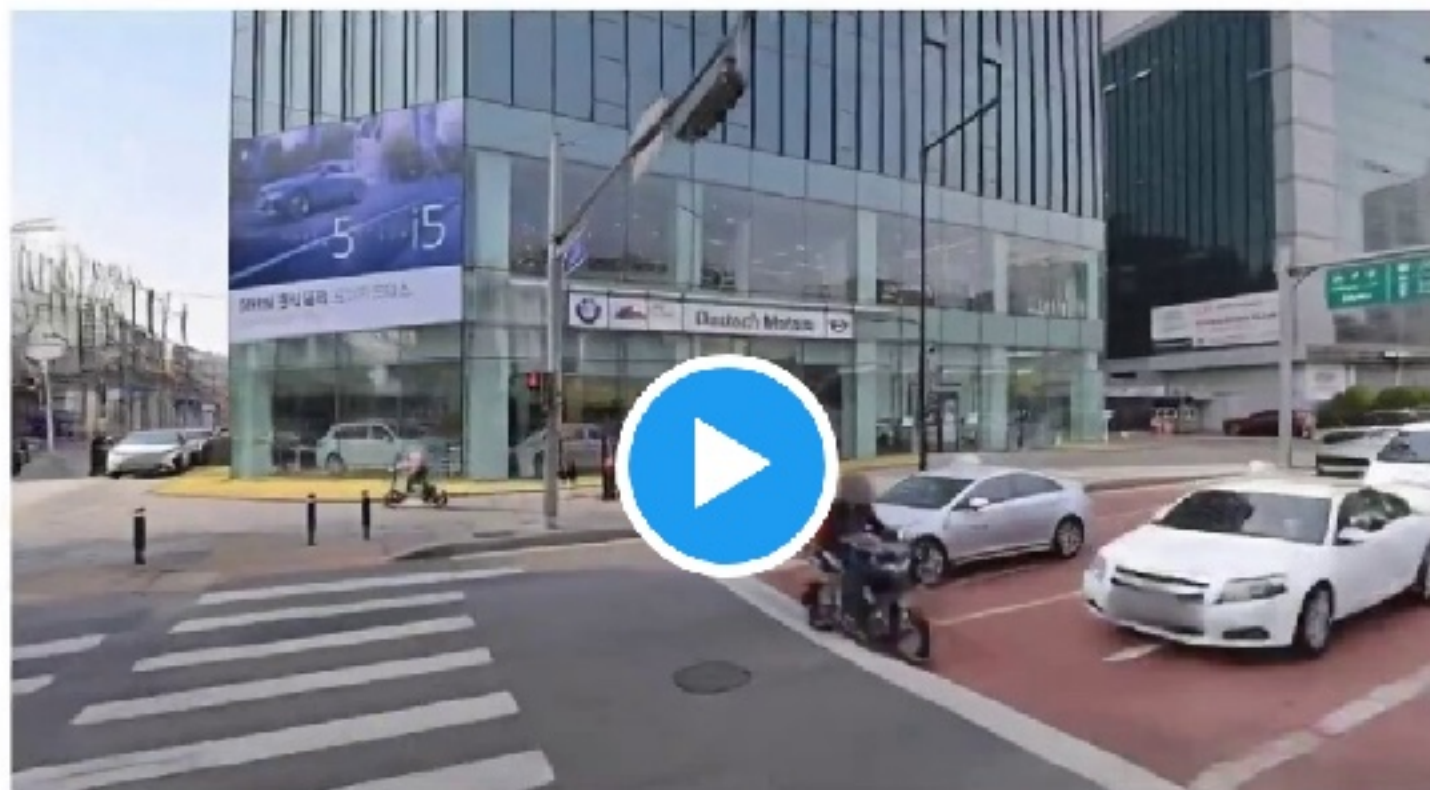


Deedy
@deedydas



Blows my mind that we currently possess the technology for Google Maps to turn all the street view images of the entire world into a video game you can play!

In the future, we'll be able to say "yeah let's check out New York City 100 years ago!"

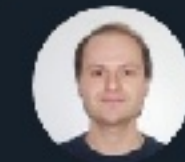


Given a starting location and camera actions, users can freely navigate within the coverage area.

2x speed

1:19 AM · Apr 2, 2026

2.3K Reply



Thomas Kipf
@tkipf



This is one of the most impressive world model projects I have seen. Very elegant and highly effective combination of an image retrieval mechanism (using 3D locations/views) and otherwise just pure generative modeling. This is the way.

11:32 PM · Mar 17, 2026

264 Reply



Top AI Papers 2026-W12

1. Demystifying Video Reasoning
2. AI Can Learn Scientific Taste
3. InCoder-32B
4. SocialOmni
5. MiroThinker-1.7 & H1
6. HSimul3R
7. OpenSeeker
8. EnterpriseOps-Gym
9. Qianfan-OCR
10. Grounding World Simulation Models in a Real-World Metropolis

World Simulation Models

World Simulation Models (WSMs)

- World Simulation Models $\mathcal{W}_\theta : (s_t, a_t) \rightarrow s_{t+1}$
 - *a.k.a.* World Model, World Generation Model
 - Given text, images, and actions, it generates a temporally consistent *virtual* world.
- Technical properties
 - Video Diffusion Transformer
 - Spatio-temporal consistency
 - Future predictions
 - Support multimodal inputs (e.g., textual prompts, initial frames, action sequences)

Our Motivation

Sovereign AI X Physical AI

Geospatial in Sovereign AI

- *Geospatial AI* understands and predicts the *physical world* by integrating spatial data across cities, infrastructure, and environments.
- *Sovereign AI* requires secure, self-operated 3D spatial models because strategic spatial assets cannot rely on external AI ecosystems.
- Without Geospatial AI, omni-modal systems may face limits in supporting large-scale national and enterprise decision-making.

3D City Models of Seoul
(NAVER LABS, 2019)



Grounding in a physical world

- Most recent world models generate plausible but imaginary environments.
 - SWM instead models the real world:
 - Renders videos of an actual metropolis (Seoul, Republic of Korea)
 - Leverages *large-scale* street-view data to capture real urban geometry
 - Retrieval-Augmented Generation (RAG) enables:
 - Efficient learning without memorizing the entire world
 - Direct use of existing, *continuously* updated map data, *not full retraining*
- 👉 From imagination to a physically grounded, updatable reality

Limitation of previous WSMs

- Previous WSMs generate and simulate a *virtual* world given by an initial image or text prompt.
- Thus, it lacks world-specificity, reflecting the real-world configuration.



Real-world geospatial conditioning

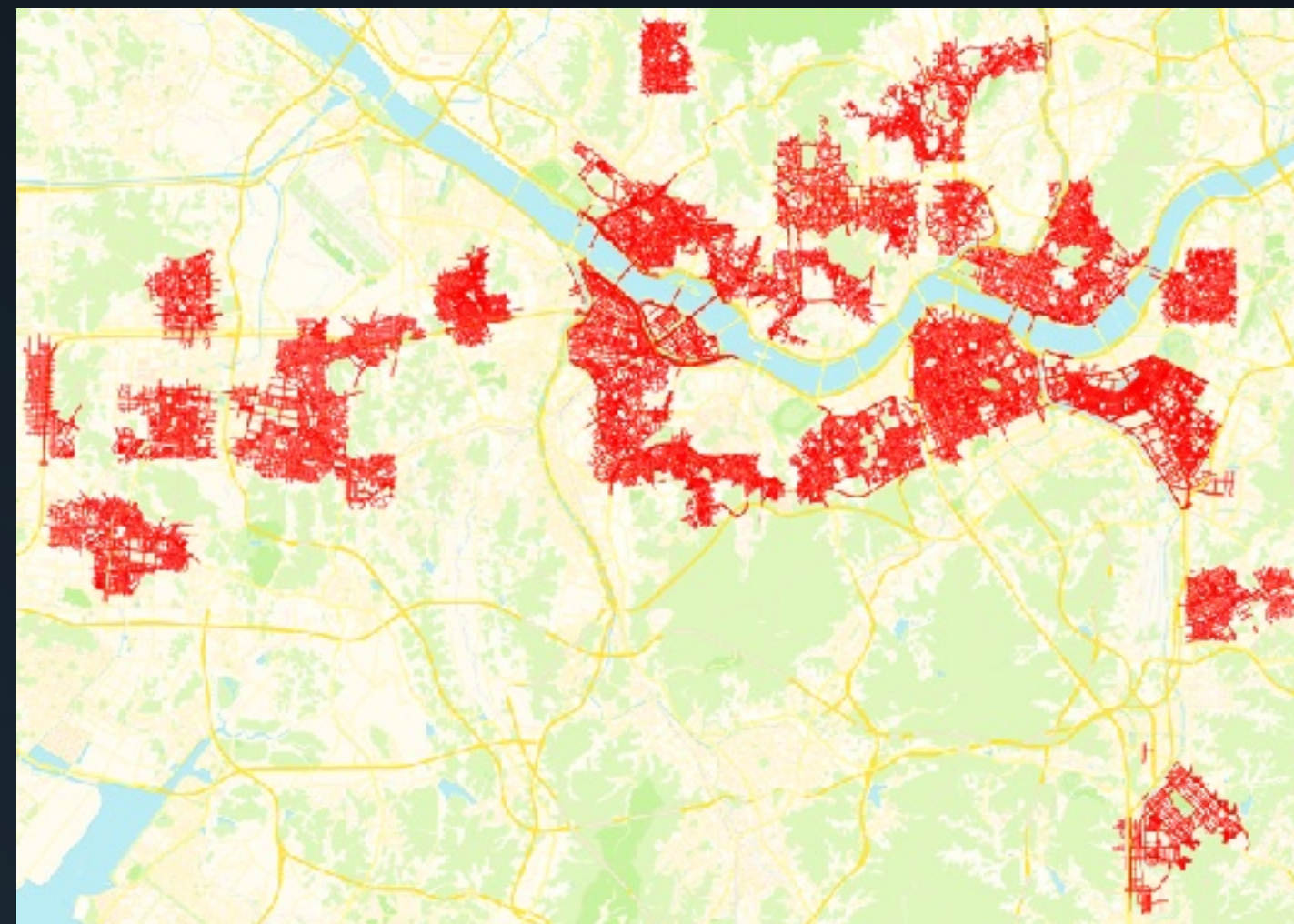
- Grounded generation of realistic environments
- Efficient training with retrieved street-view data and augmentation



Data Construction

Training data

- We build aligned pairs of K street-view references (input) and target video sequences.
- All reference and target images in a chunk are fed into **Depth Anything 3** (Lin et al., 2025) to obtain the corresponding depth maps and camera poses in a shared space.
- We recover *global coordinates* by aligning per-chunk camera poses to GPS coordinates.



Coverage area. Spatial distribution of collected street-view data within the Seoul metropolitan area, roughly 44.8km x 31.0 km.

By default, $K = 5$. The further details can be found in Appendix A.2 and C.3.

Two primary sources

- Aligned pairs of street-view references and target video sequences from two sources:
 - **NAVER Map Street-View**: 1.2M real panoramic images captured across Seoul and the surrounding Gyeonggi Province
 - **CARLA Unreal simulator**: 12.7K synthetic videos from an Unreal Engine-based CARLA ([Dosovitskiy et al., 2017](#)) urban simulator spanning 431,500 m² of city area.



NAVER Street View



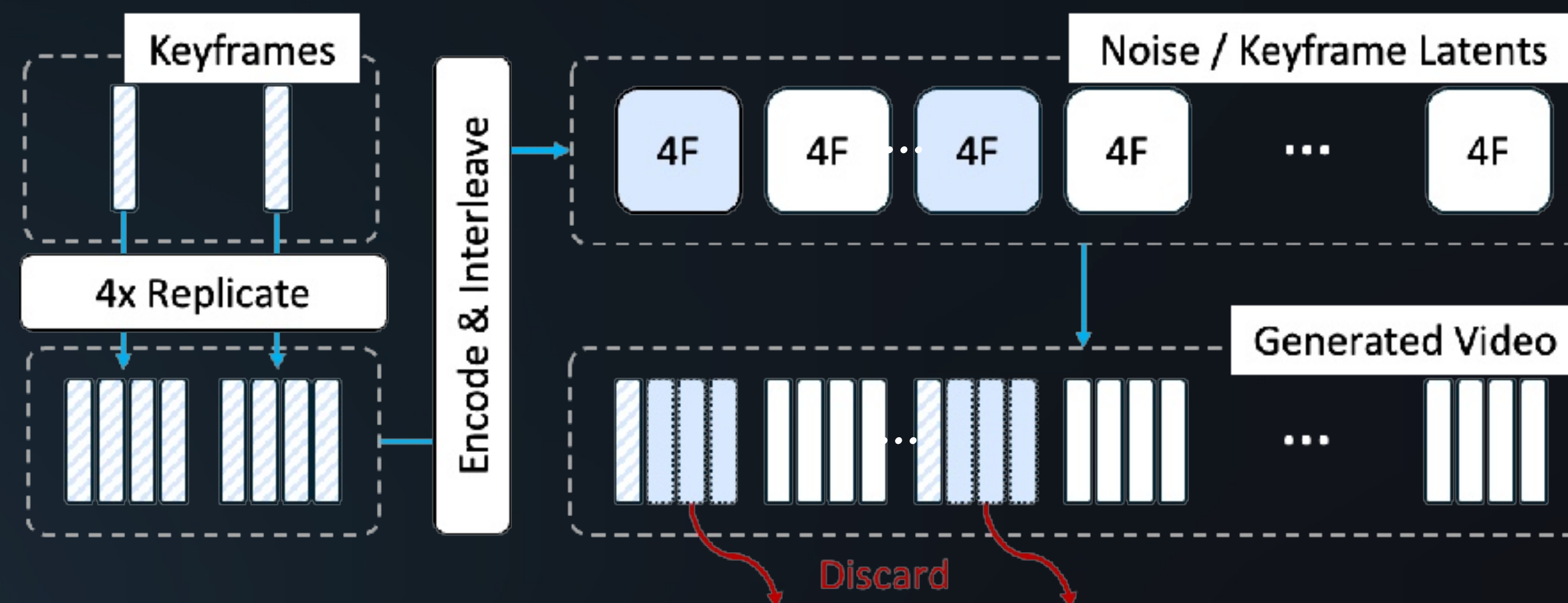
Examples of synthetic training data using CARLA

Street View is not a video!

- Using Street View poses two challenges for dynamic video generation: 1) *the absence of a corresponding target **Street View video*** and 2) *temporal inconsistency*.
 - Recordings when capturing street view are unavailable.
 - Nearby Street View images are not guaranteed to be captured consecutively.
- We propose two design choices: *view interpolation* and *cross-temporal pairing*.

From sparse Street Views to video

- To handle the large viewpoint gaps in Street View, we fine-tune the interpolator using Waymo driving videos with *intermittent freeze-frame* segments:
 - Each keyframe is repeated 4 times in a row, so the 3D VAE encodes them into a single latent vector.
 - The resulting videos alternate between brief freeze-frame and smooth-motion segments.
- After generation, the three duplicated frames per keyframe are removed.



Quantitative comparison

- We compare the two strategies quantitatively on the Waymo ([Kretzschmar et al., 2020](#)) test set, measuring PSNR, SSIM, and LPIPS between the interpolated and ground-truth videos.
 - IFF consistently and significantly outperforms across all metrics.
- ⚠ Notice that *view interpolation alone cannot generate an arbitrary trajectory* exploring Seoul.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
Channel Concatenation	22.52	0.628	0.245
Intermittent Freeze-Frame (Ours)	25.03	0.703	0.162



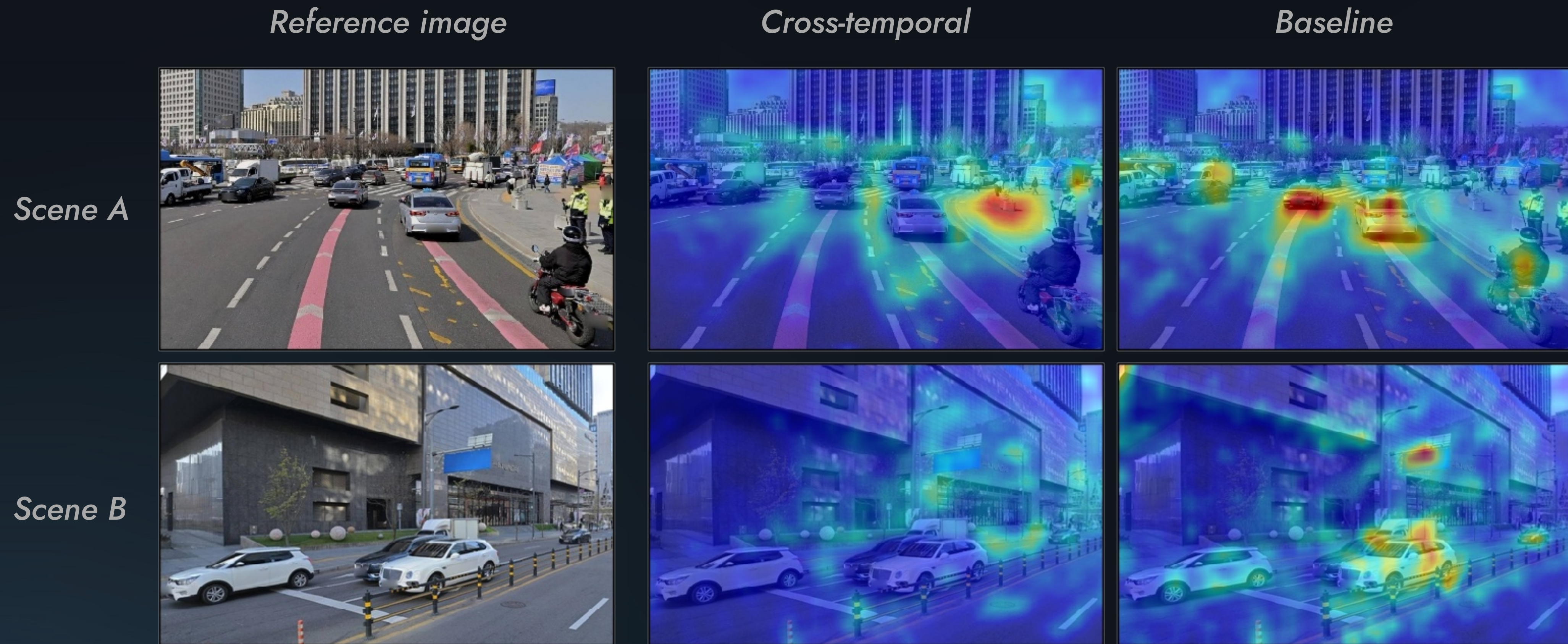
Cross-temporal pairing

- During training, the references must be captured at a different timestamp than the target sequence to prevent *spurious correlation*.
- Since transient objects often differ between reference and target images, the model must learn to rely on a temporally persistent spatial structure.



●: start point, ●: end point, and ●: reference points

Comparative results

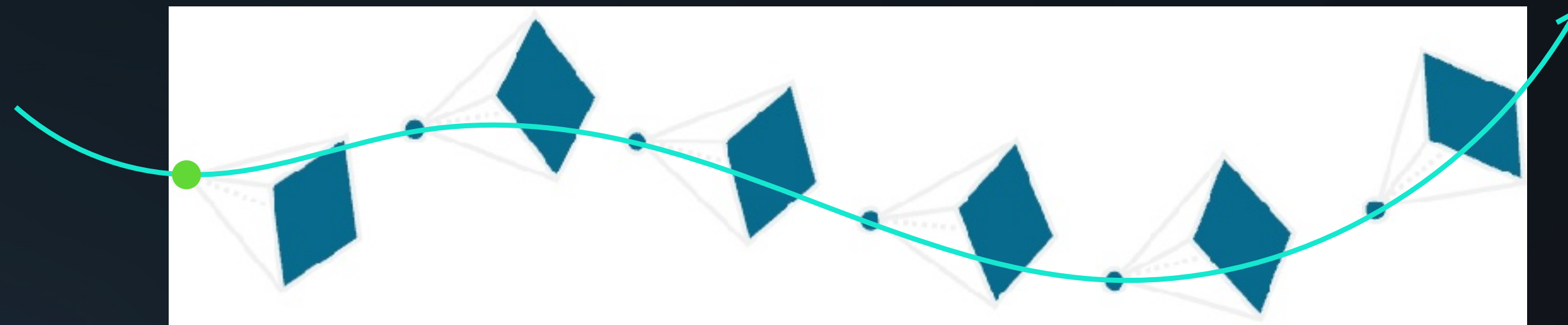


Cross-temporal pairing makes the model less attentive to dynamic objects (e.g., cars) as we intended.

Model

From Inputs to worlds

- User inputs
 - A starting location in world coordinates (can be absorbed into a camera trajectory)
 - A camera trajectory (a sequence of camera poses), which can be derived from user navigation
 - A text prompt (or a boring default prompt)



"A photorealistic urban street scene with dynamic traffic and city-scale surroundings under natural lighting."

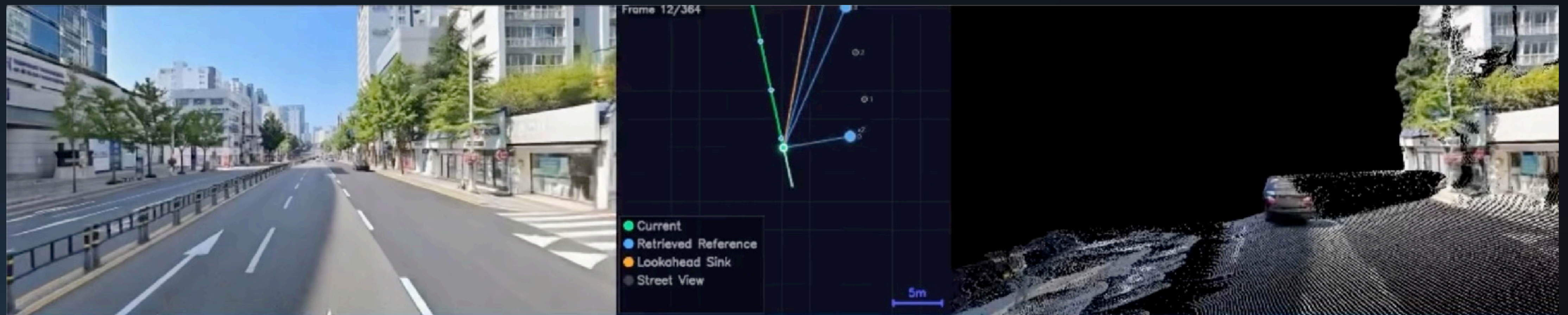
Model architecture

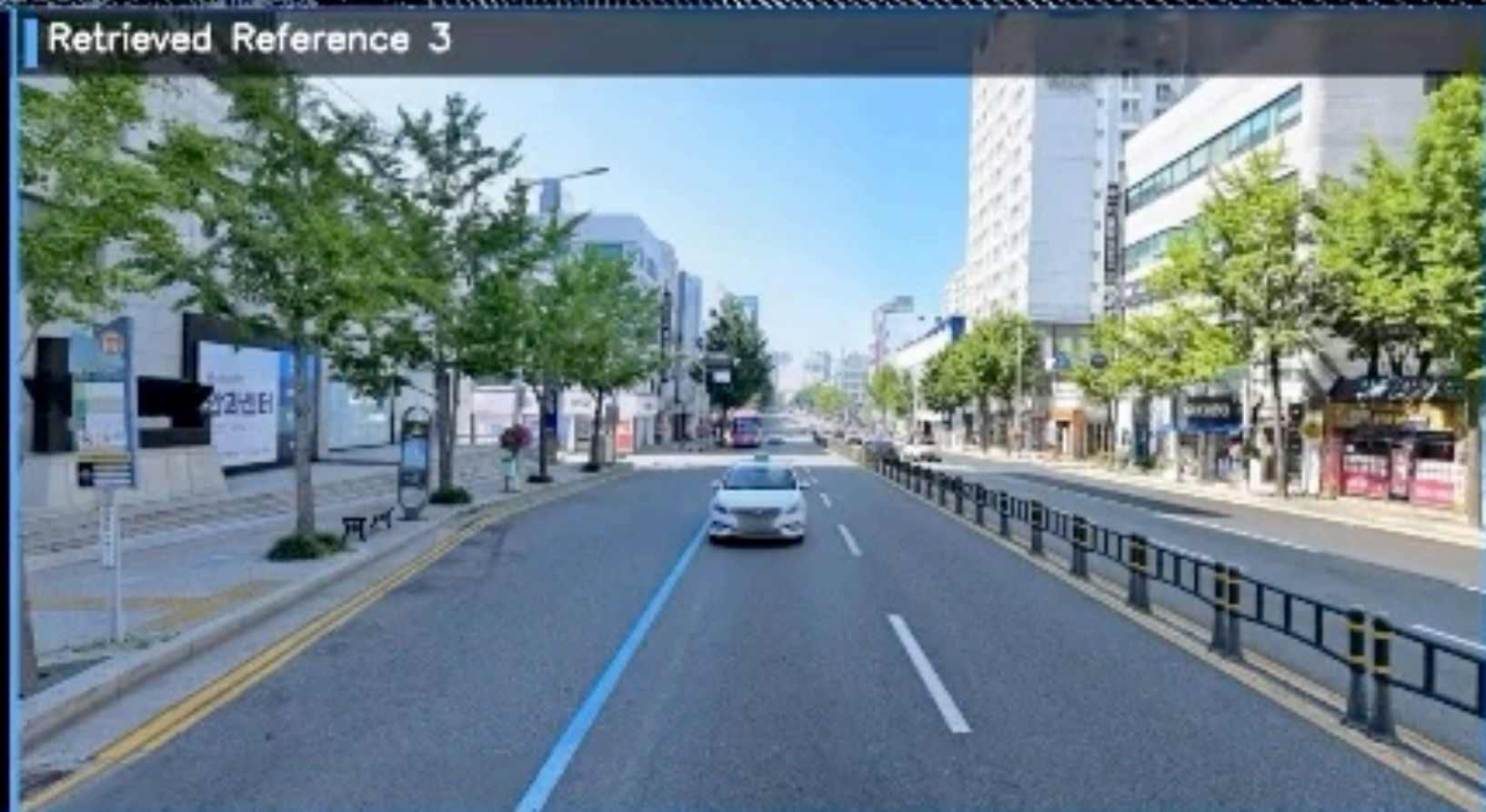
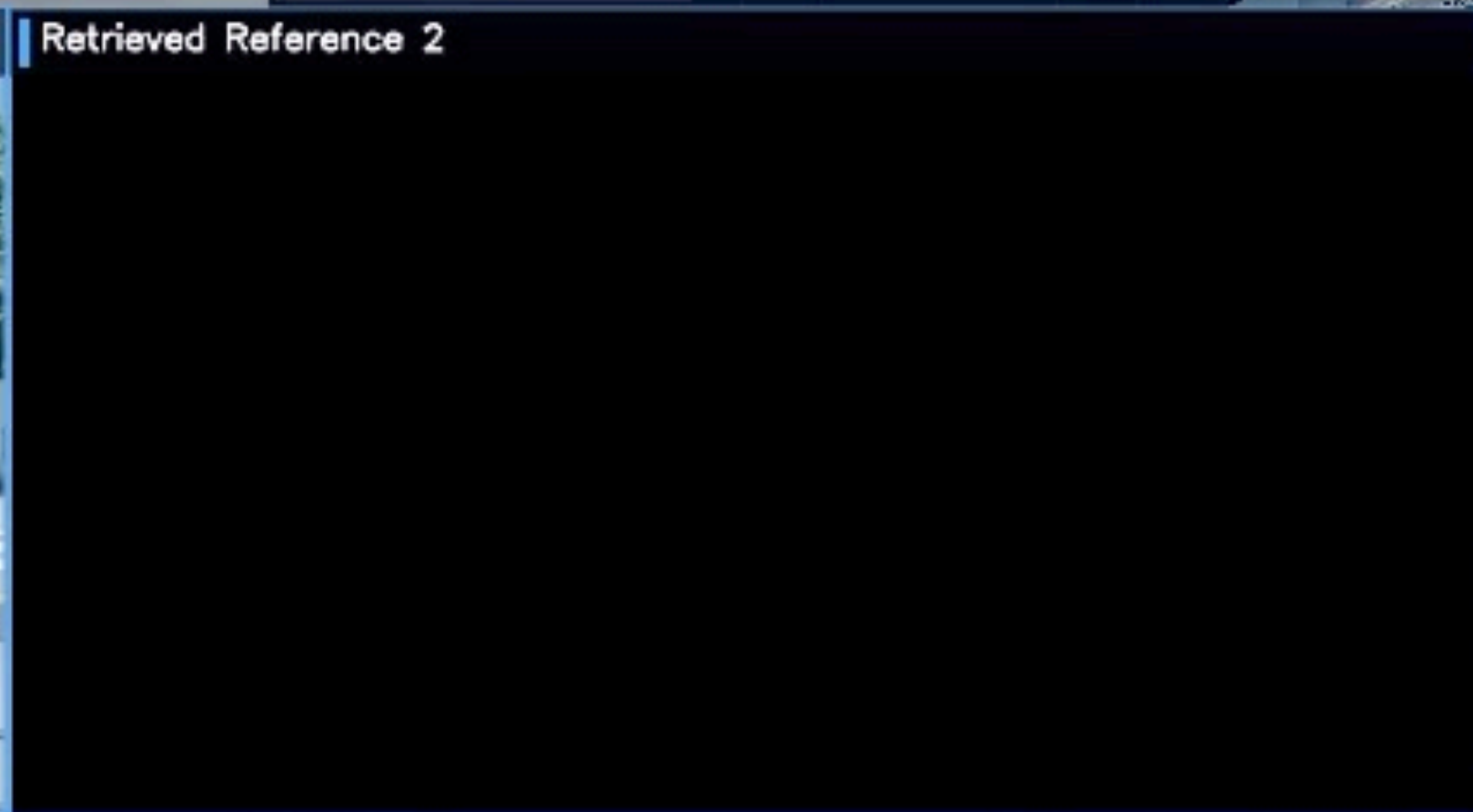
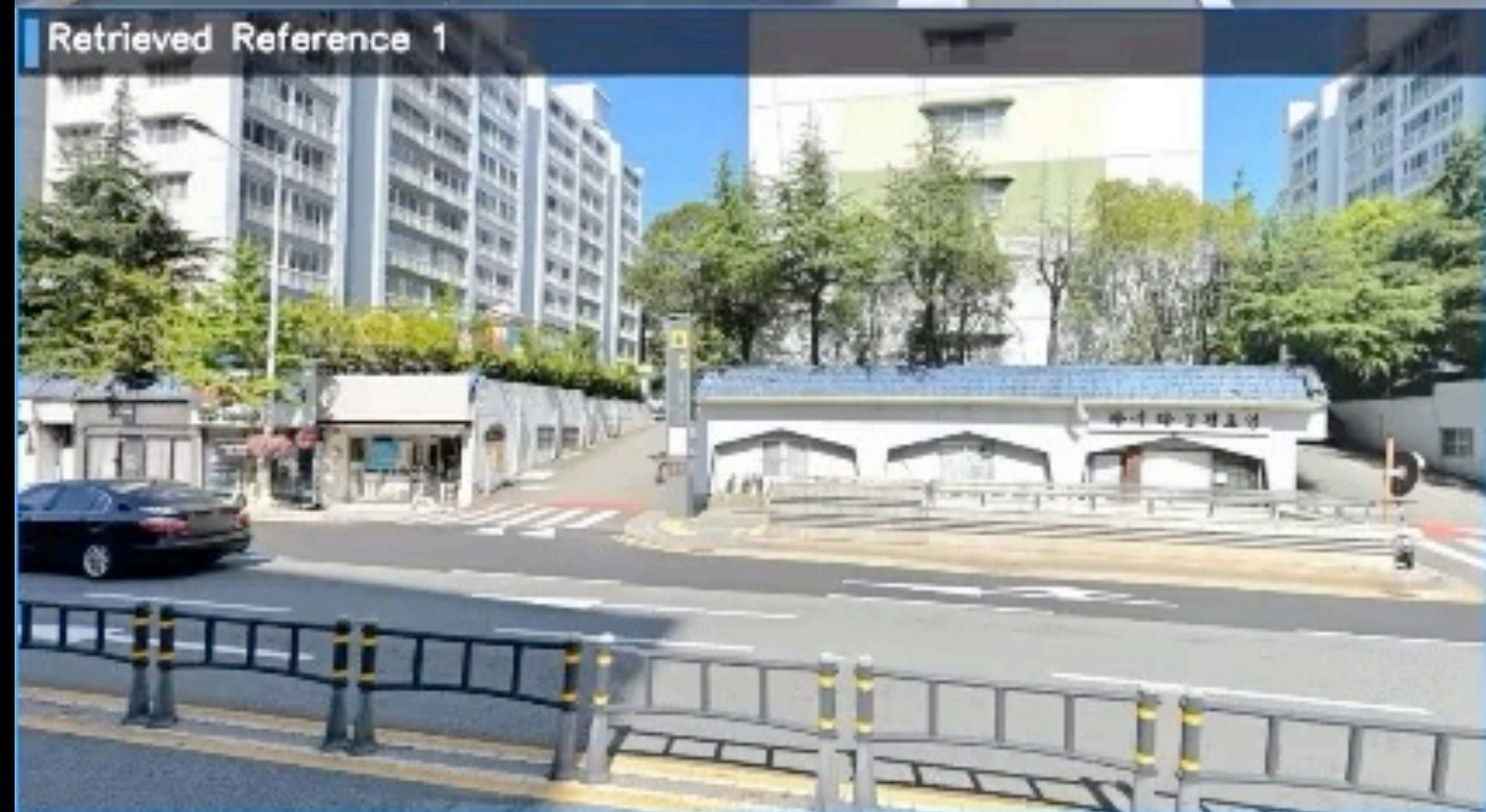
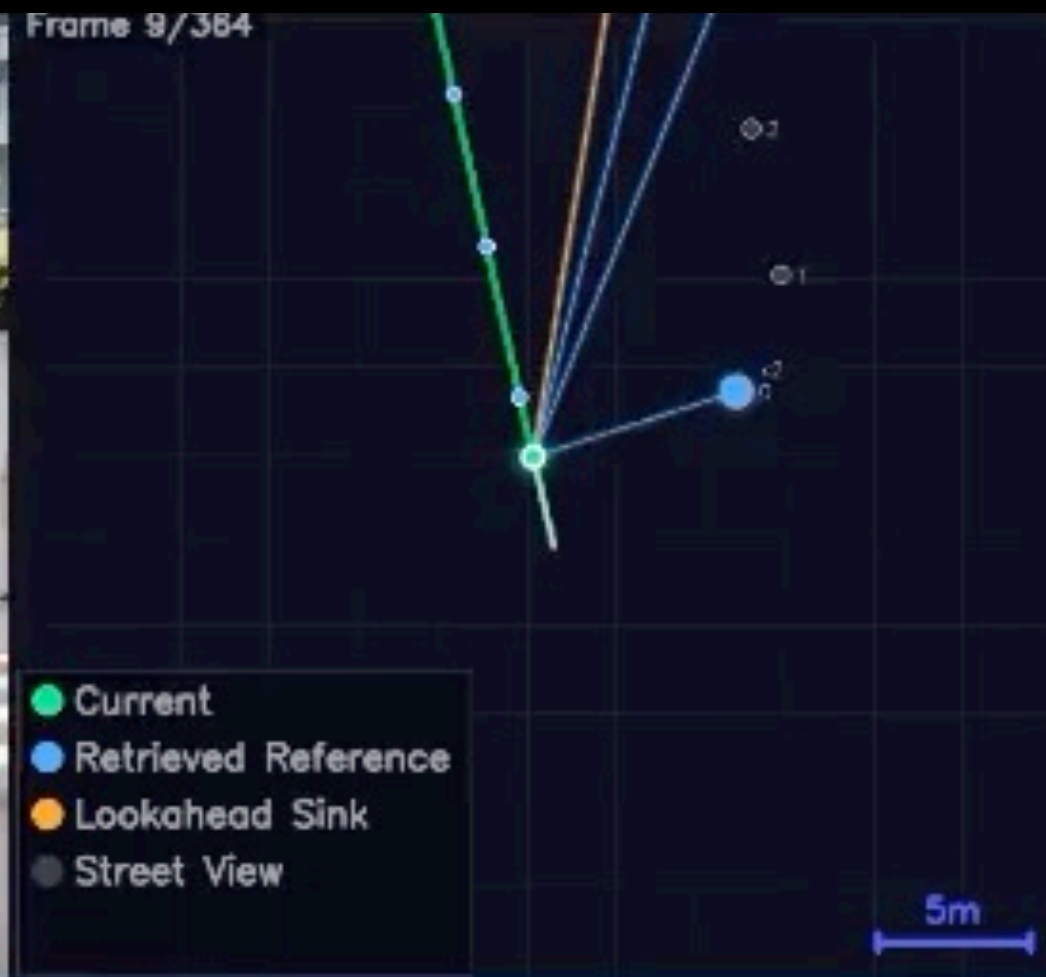
- A pretrained Diffusion Transformer* (Peebles et al., 2023; Agarwal et al., 2025)
 - For each chunk, noisy latents $Z^{(i)} = \{z_l^{(i)}\}_{l=0}^{L-1}$ to produce target latents $\hat{Z}^{(i)} = \{\hat{z}_l^{(i)}\}_{l=0}^{L-1}$.
 - Each subsequent chunk additionally conditions on history latents $Z_h^{(i)} = \{\hat{z}_l^{(i-1)}\}_{l=L-H}^{L-1}$ from *the tail of the preceding chunk's output*, providing temporal continuity ($H < L$).
 - At this moment, the model needs to memorize the entire world to generate videos from only input trajectories, exposing it to *error accumulation* in city-scale, long-horizon generation.
- 👉 Three key points in modeling:
Street View RAG, Virtual Lookahead Sink, and Geometric and Semantic Referencing.

*Cosmos-Predict2.5-2B-I2W, 2B DiT with 28 blocks, 16 attention heads, and a hidden size of 2,048.

Street View RAG

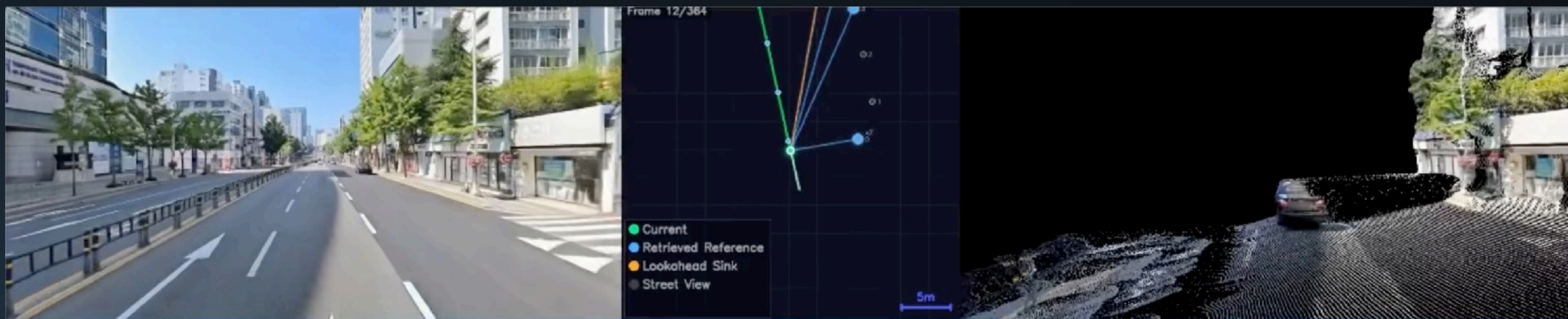
- SWM performs retrieval-augmented generation (RAG):
 - Given geographic coordinates, camera actions, and text prompts,
 - it utilizes geometric and semantic references (will discuss later), *adaptively retrieving* nearby street-view images from 1.2M panoramic images after pinhole rendering.
 - This anchors each generated chunk to the real layout and appearance of the location.





Virtual Lookahead Sink

- Autoregressive generation accumulates errors even with forcing-based distillations, diffusion, rolling, and self-forcings (Chen et al., 2024; Liu et al., 2025; Huang et al., 2025).
- The attention sink (Xiao et al., 2024; Shin et al., 2025) for the initial frame also has problems in long horizontal generations, as it becomes irrelevant in later frame generations.
- We use a nearby *future* Street View image as a *virtual lookahead sink*.

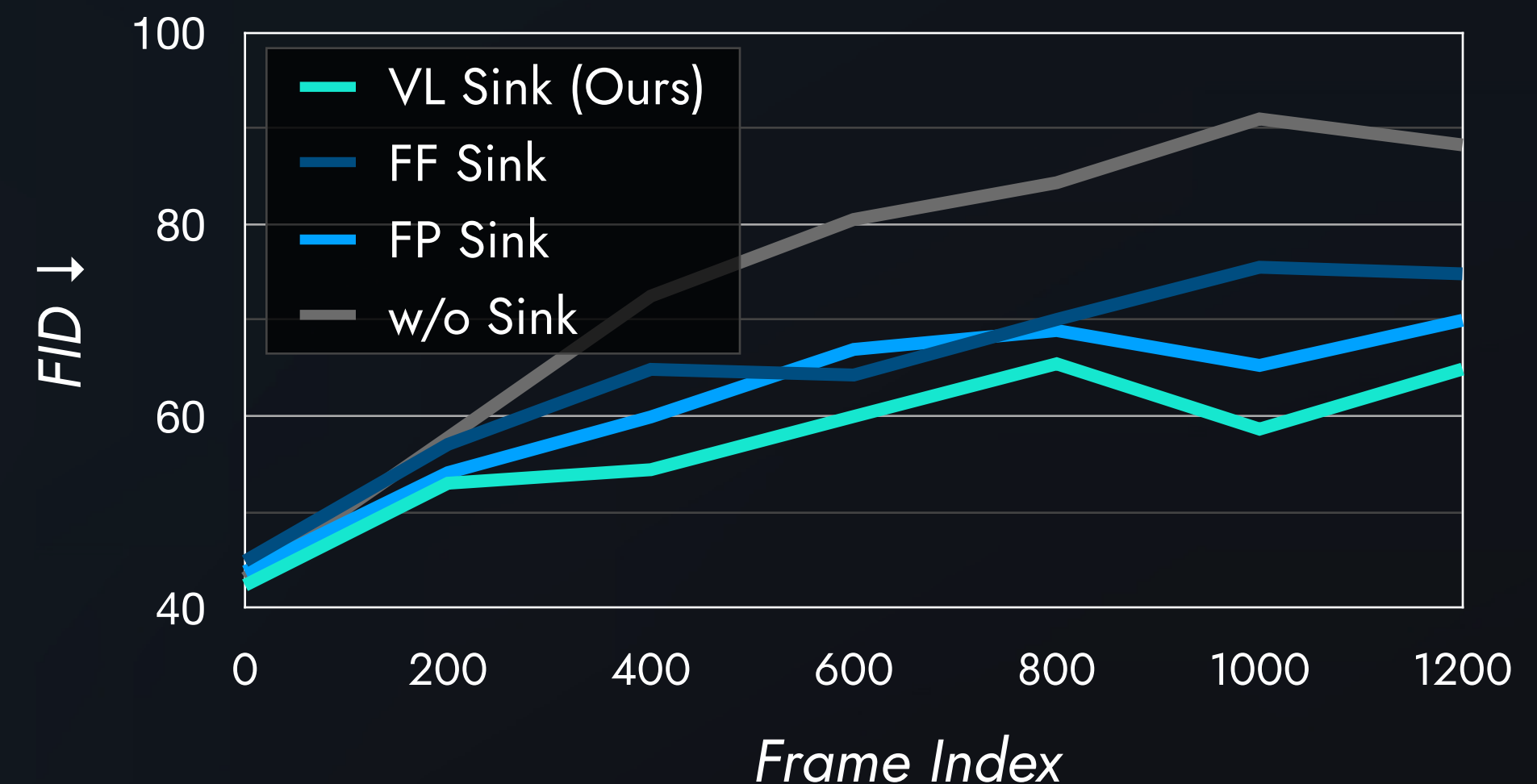
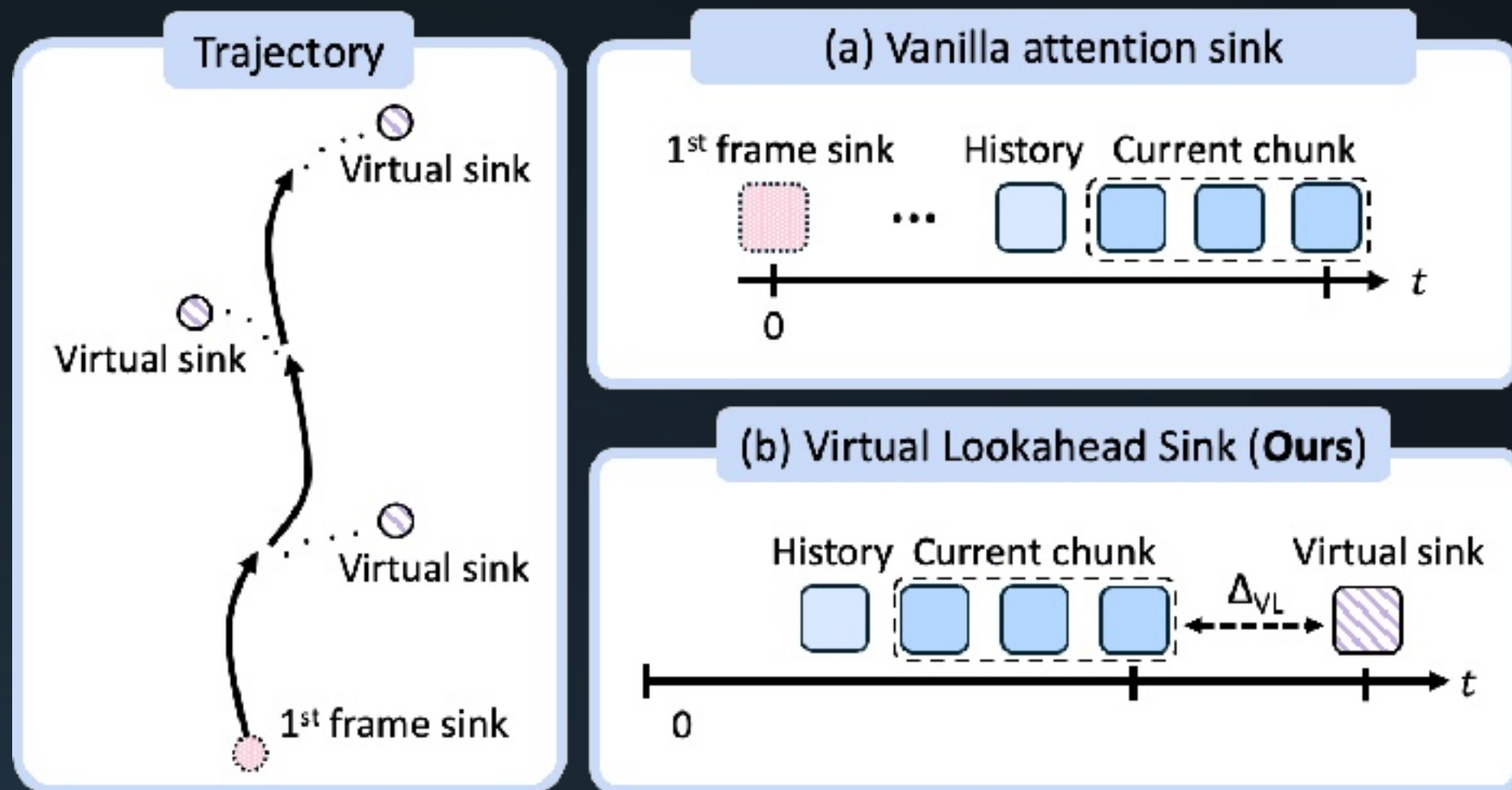


●: current location, ●: retrieved references, and ●: virtual lookahead sink

Virtual Lookahead Sink (cont'd)

- To sum up, the latent sequence fed to the model for the i -th chunk is $Z_{seq}^{(i)}$ and its RoPE temporal positions are $p_{seq}^{(i)}$, with Δ_{VL} of a temporal offset hyperparameter, as follows:

$$Z_{seq}^{(i)} = \left[\underbrace{Z_{1:H}^{(i)}}_{\text{history}}, \underbrace{Z_{H+1:H+L}^{(i)}}_{\text{target}}, \underbrace{z_{VL}^{(i)}}_{\text{sink}} \right] \quad \text{and} \quad p_{seq}^{(i)} = \left[\underbrace{1, \dots, H}_{\text{history}}, \underbrace{H+1, \dots, H+L}_{\text{target}}, \underbrace{H+L+\Delta_{VL}}_{\text{sink}} \right].$$



FF: First-Frame (baseline); FP: First-Position, where the retrieved reference image is placed at, and the attention sink is on it.

Geometric referencing

- For the j -th reference $x_{ref,j}$, the corresponding depth $d_{ref,j}$ and the relative camera transformation $c_{ref,j \rightarrow t}$, we reproject into the target viewpoint via depth-based forward splatting to get $x_{warp,t}$:

$$x_{warp,t} = \mathcal{T}(\pi^{-1}(x_{ref,j}, d_{ref,j}), c_{ref,j \rightarrow t}).$$

- Here, π^{-1} lifts the reference pixels to a 3D point cloud using the depth information, while $\mathcal{T}$ transforms the 3D point cloud into the target view.
- The warped video $X_{warp} = \{x_{warp,t}\}_{t=0}^{T-1}$ is encoded by the 3D VAE and *channel-wise concatenated* with the noisy target latent in the DiT input, Z_{seq} .

* $T = 77$ is the sequence length of Z_{seq} , along with the corresponding warpings (defined informally for brevity).

Semantic referencing

- The original reference latents z_{ref} are similarly concatenated with the target latents, *but along the temporal axis* at the RoPE position:

$$p_{ref,k} = H + L + G + k\Delta_{ref}.$$

- Here, G is a large temporal gap separating references from the generation window and Δ_{ref} is the inter-reference spacing.
- Unlike geometric referencings, and since it's temporal-axis concatenation, we use only the nearest reference per frame.
- Although the attention mechanism allows us to attend to all appearance cues of K references.

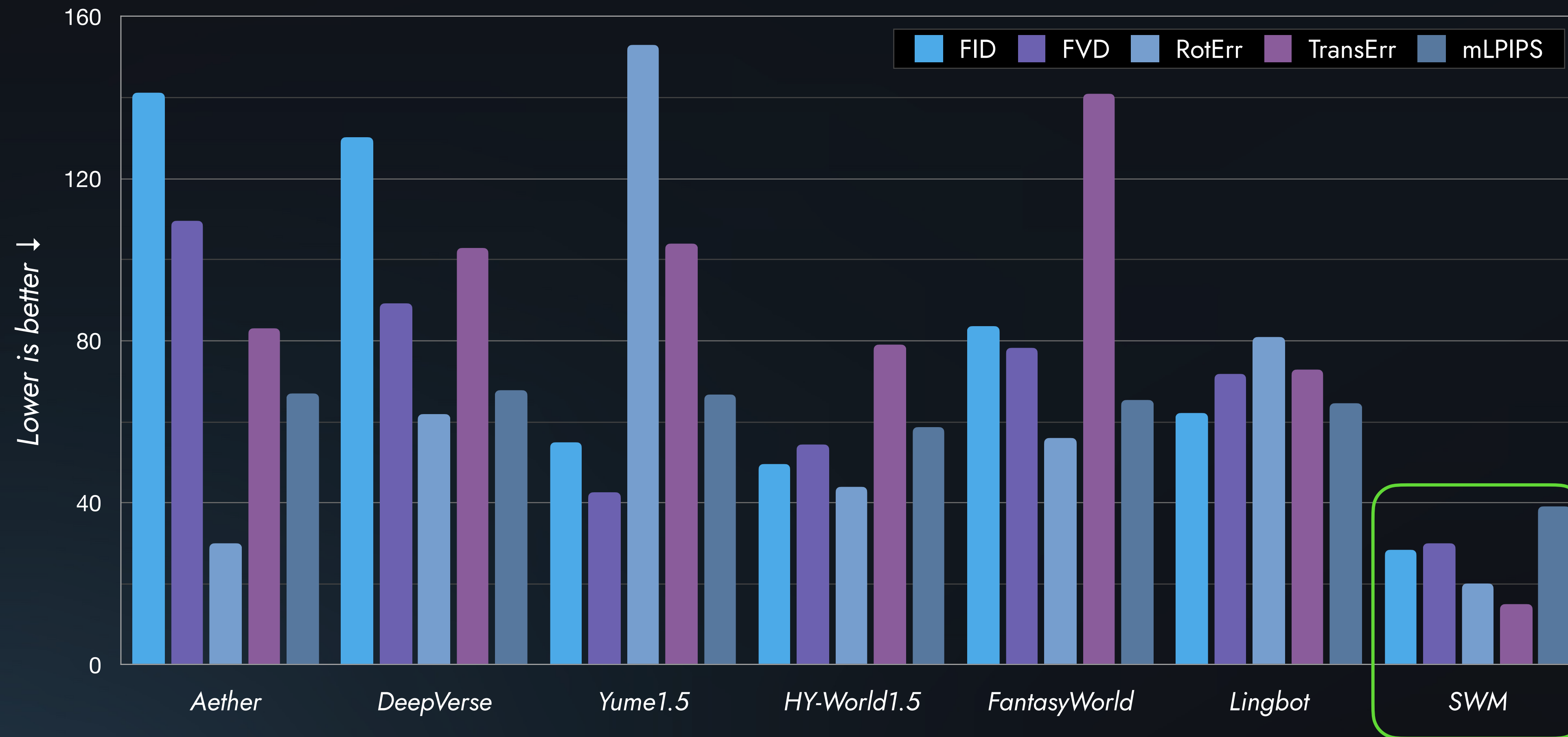
Experiments

Experiment setup

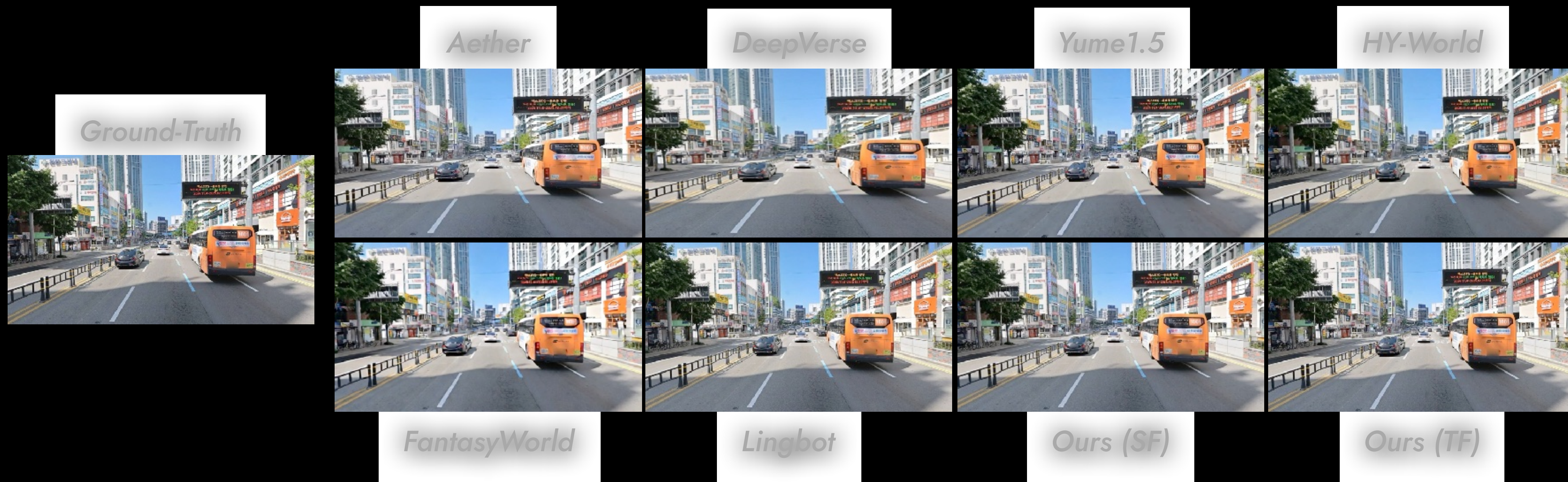
- Training specifications
 - Data preprocessing: $20K \times 0.1h$ (2K hours/gpu)
 - Video Interpolation model: $H100 \times 8$ (1 node), 1 day (192 hours/gpu)
 - Main model: $H100 \times 24$ (3 nodes), the batch size of 48, 4 day (2K hours/gpu)
- Evaluation benchmarks
 - Busan city from NAVER Street View, Ann Arbor City (in Michigan) from MARS*
 - Each benchmark contains 30 test sequences, consisting of 365 frames (~100 meters)
- Evaluation metrics
 - Visual and temporal quality: FID, FVD, and VBench
 - Camera-following accuracy: Rotation and translation errors
 - 3D adherence: Masked PSNR and LPIPS using SAM3 ([Carion et al., 2025](#))

*<https://ai4ce.github.io/MARS>

Quantitative comparisons



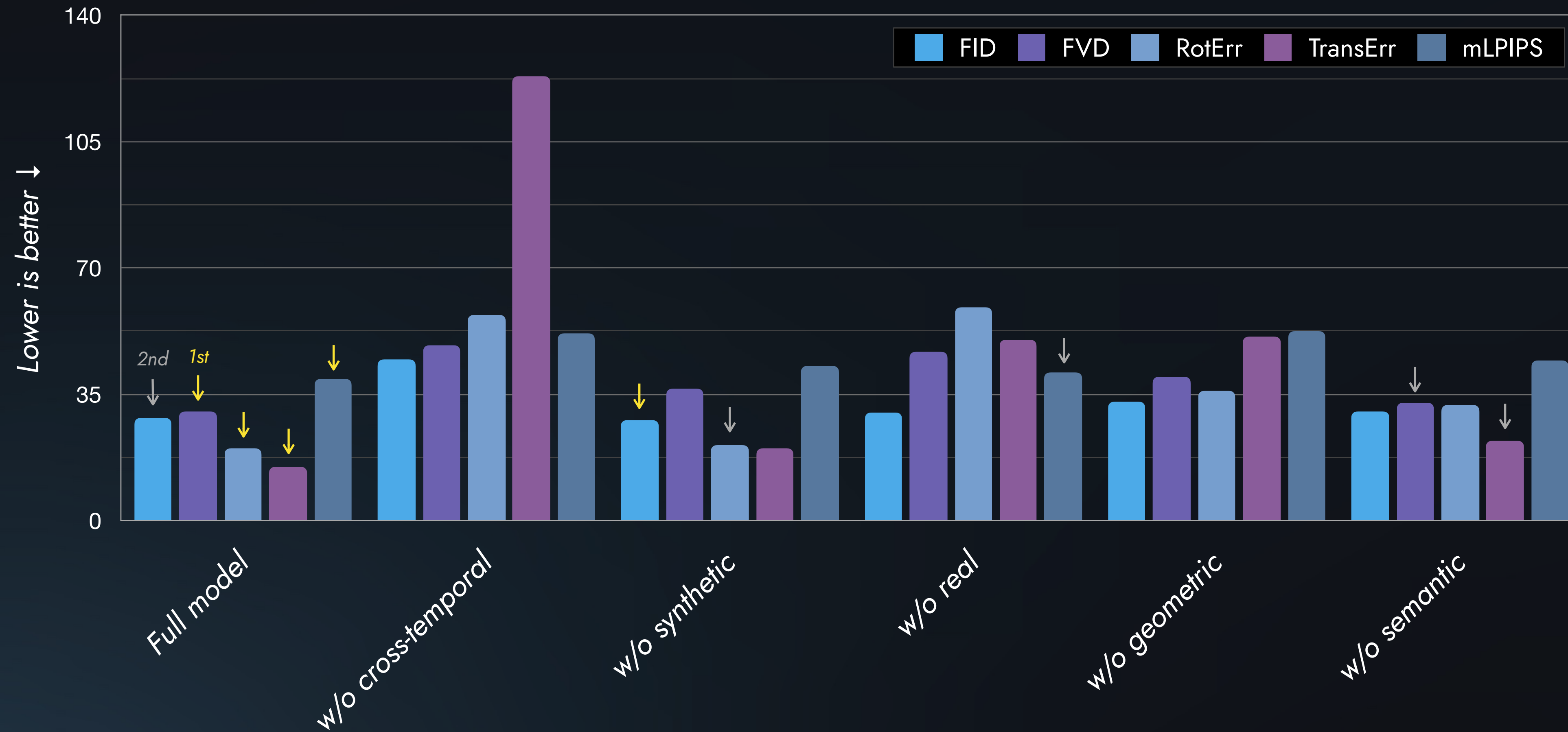
Metrics are rescaled for visualization: FVD $\times 0.1$, RotErr/TransErr $\times 1000$, and mLPIPS $\times 100$ on Busan-City-Bench.



Grounded in the Real World

Side-by-side with baselines: by retrieving real Street View, SWM grounds the actual world, while generation-only models hallucinate plausible but incorrect geometry.

Ablation study



Metrics are rescaled for visualization: FVD $\times 0.1$, RotErr/TransErr $\times 1000$, and mLPIPS $\times 100$ on Busan-City-Bench.



Ours trained without Cross-Temporal Pairing



Ours trained with Cross-Temporal Pairing

x2 speed

Ablation study

A clear, visual picture of what each component contributes.

Qualitative ablation results



Ground truth

Full model

Without a virtual lookahead sink,
errors accumulate over long trajectories.

Without semantic referencing,
appearance details are lost.



Ours without Virtual Lookahead Sink



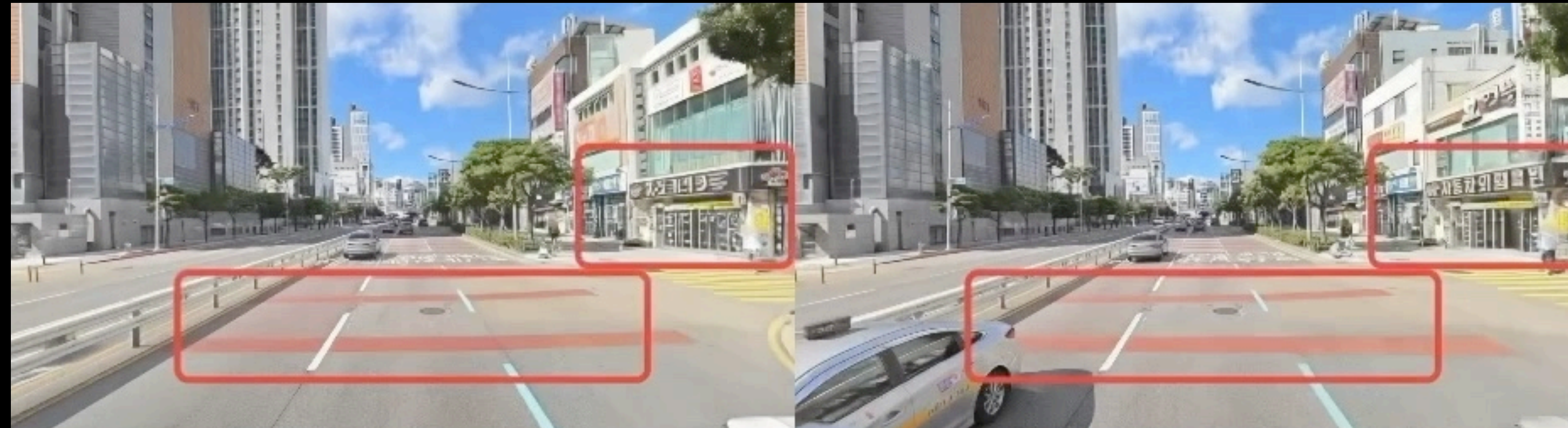
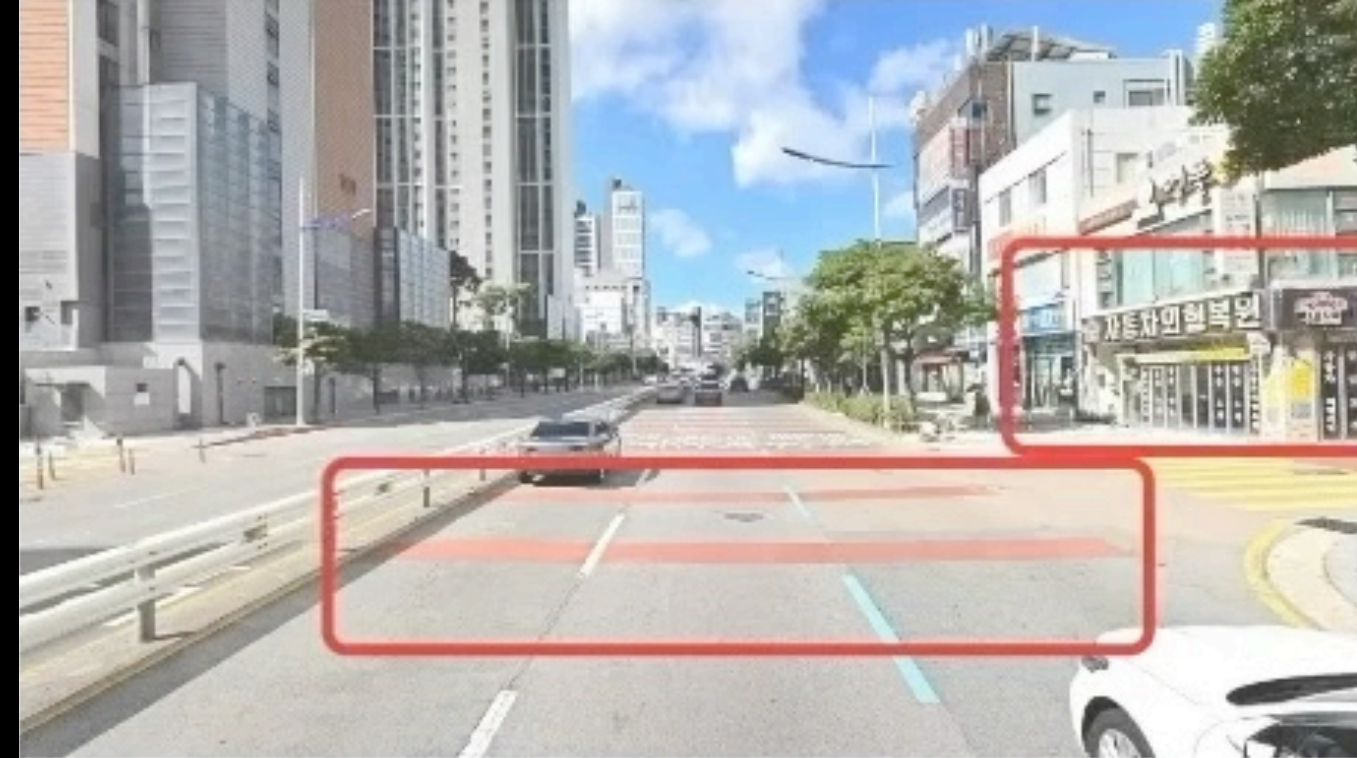
Ours with Virtual Lookahead Sink

x2 speed

High Quality, Without Error Accumulation

The Virtual Lookahead Sink removes the error accumulation that limits autoregressive video generation while semantic referencing restores fine detail — high fidelity sustained over long horizons.

Ground-Truth



Ours without Semantic Referencing

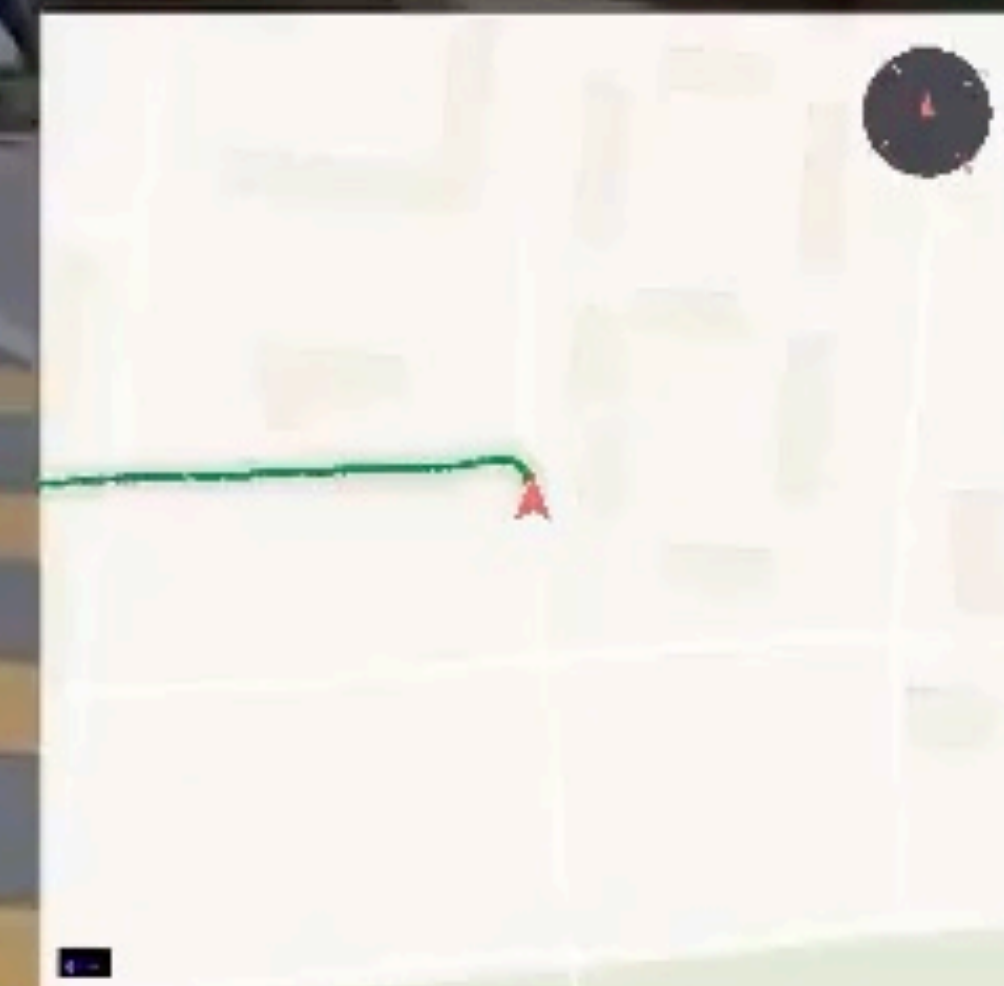
Ours with Semantic Referencing

High Quality, Without Error Accumulation

The Virtual Lookahead Sink removes the error accumulation that limits autoregressive video generation while semantic referencing restores fine detail — high fidelity sustained over long horizons.

Roaming the City for Kilometers

Grounded in a real city, SWM generates videos over multi-kilometer trajectories without accumulating errors, maintaining robust performance across long-horizon generation.



A street scene in a city, likely Seoul, South Korea. The street is lined with modern buildings, some with glass facades and others with brick. There are cars parked along the sides and a few cars driving. A pedestrian is walking on the sidewalk on the right. A speed limit sign of 50 is visible on a pole. The sky is overcast.

Navigate Freely, Like in a Video Game

From strolling down a sidewalk to glancing around a corner or cruising along a highway, SWM supports free-form navigation beyond simple forward driving, letting users explore the city through arbitrary camera trajectories.



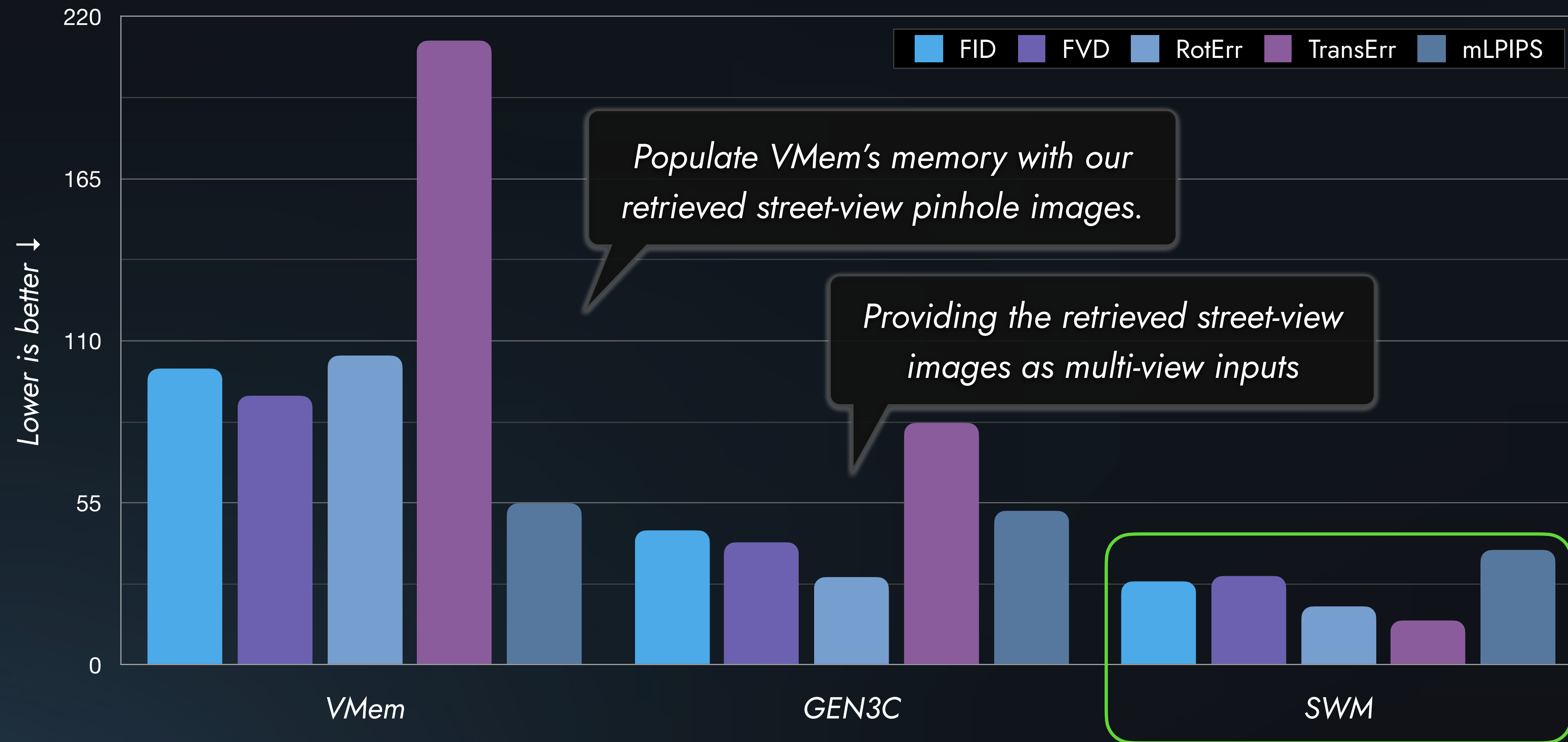
Text-Prompted Scenario Control

SWM also allows users to reshape familiar city scenes through text prompts: summon a massive wave onto the streets, drop Godzilla between skyscrapers, or imagine any scenario you like.

2x speed

What if it were **twilight** on this street?

Quantitative comparisons for static scenes

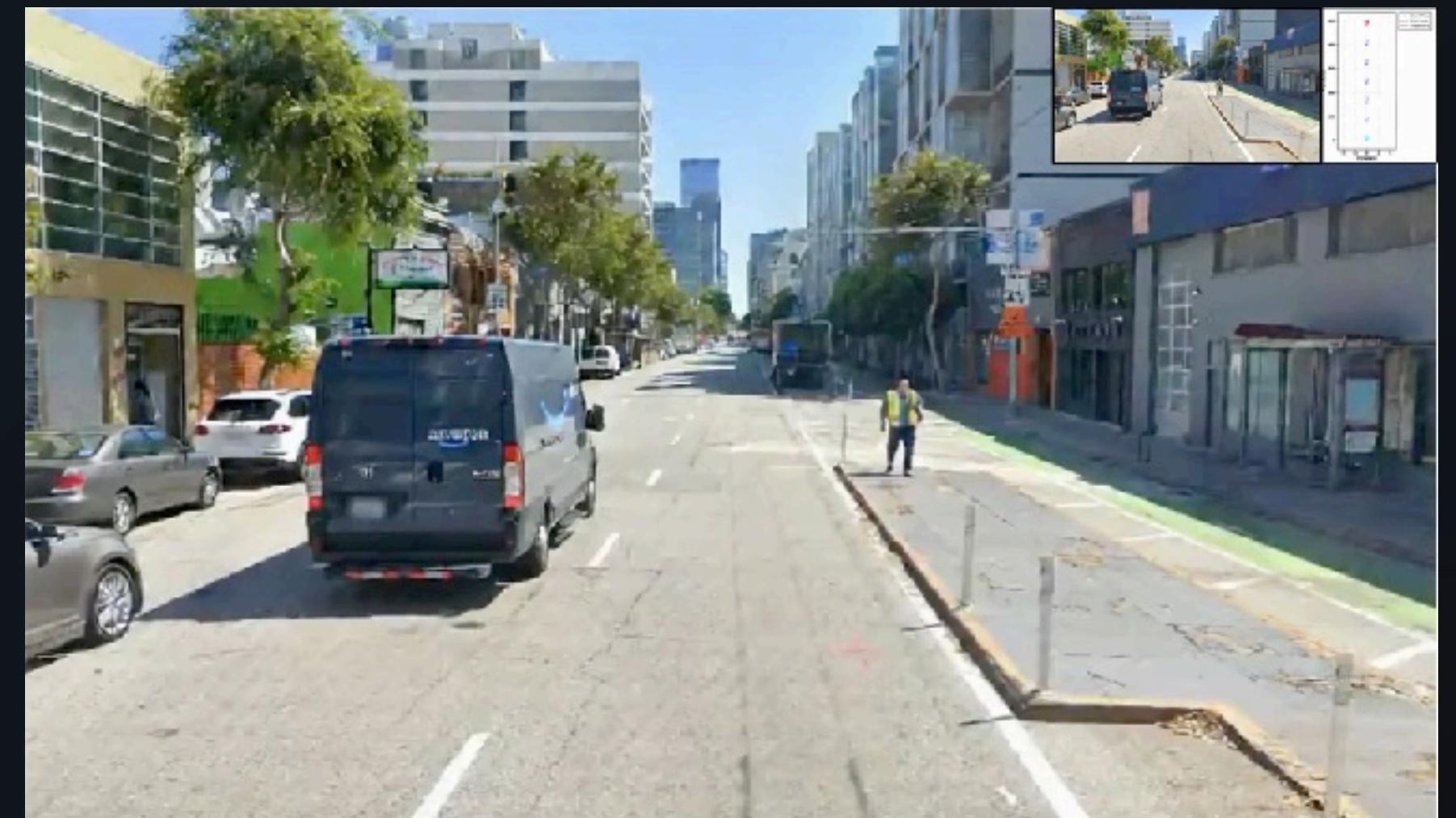


Metrics are rescaled for visualization: FVD $\times 0.1$, RotErr/TransErr $\times 1000$, and mLPIPS $\times 100$ on Busan-City-Bench.

Related Work

CityRAG

- A concurrent work (Chou et al., 2026) that also grounds video generation in a real city through retrieval over Google Street View.
- Generate a *navigable* video with the *static layout* of a real location (buildings, roads) while inheriting the *transient content* from a first image (weather, cars).
- *5.5M* Street View across 10 cities, including Paris, Honolulu, and San Francisco.
- Fine-tuned Wan 2.1 *14B* image-to-video



Design comparisons

CityRAG (Chou et al., 2026)

SWM (Ours)

10 cities

Data coverage

Seoul + zero-shot transfer to Busan, Ann Arbor

Driving + rotation augmentation

Trajectory diversity

Driving + rotation + CARLA synthetic

Image sequence (61–81 frames)

Retrieval unit

5 individual images per chunk

Single: clip latents as cross-attention K/V

Retrieval pathway

Dual: geometric warp + semantic reference tokens

Last-frame chaining only

Long-horizon anchor

Virtual Lookahead Sink (retrieved future frame)

None, first-image only (snowy)

Text control

Text-driven scenarios (monster, tsunami, etc.)

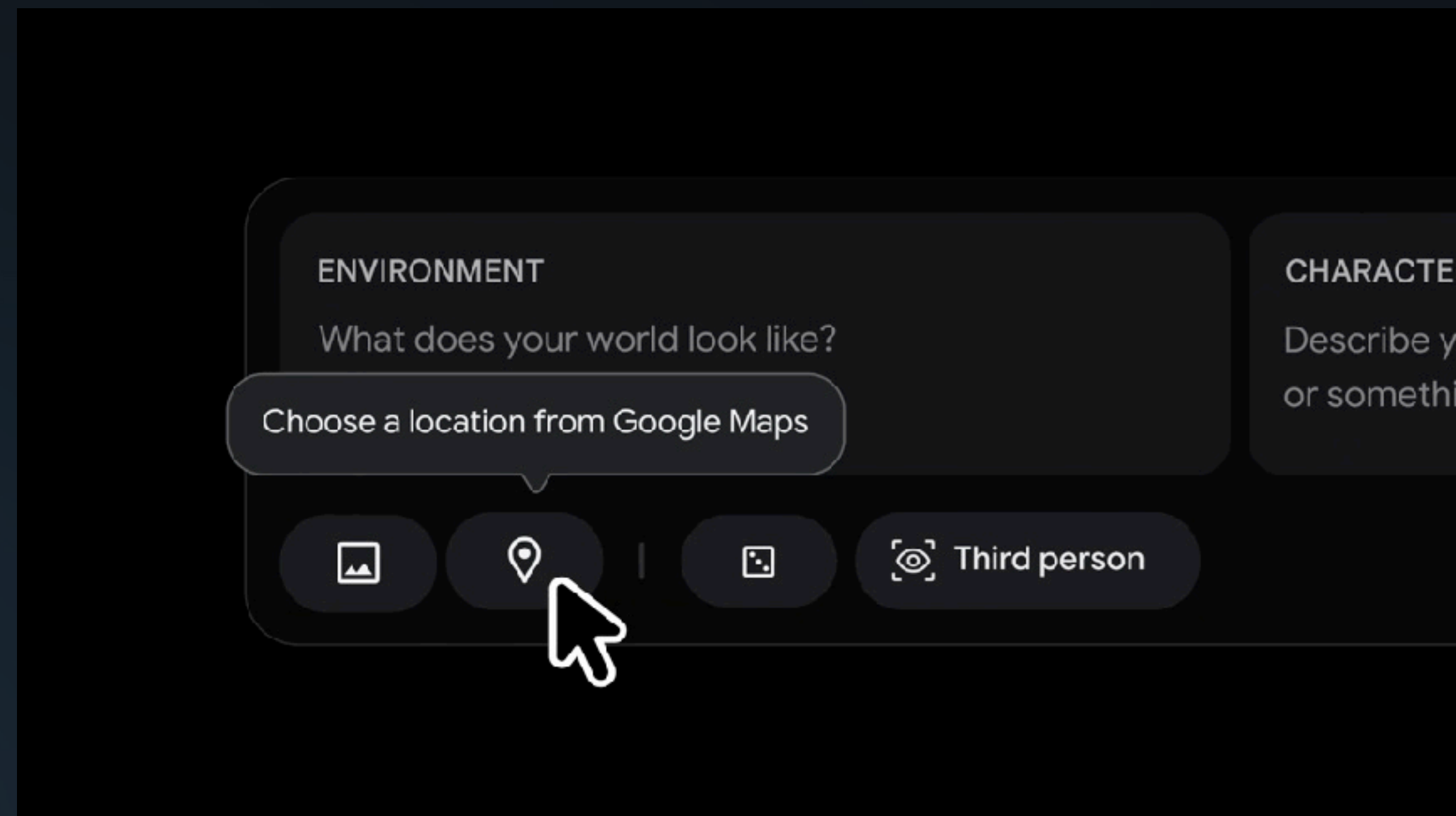
Not reported

Real-time variant

Self-Forcing: 15.2 fps for single H100

Google Genie - Street View grounding

- Google DeepMind launched a *Street View grounding* for **Project Genie** at I/O 2026.
- Pick a *Street View pin* on a map, choose a style (e.g., Ocean World), and describe a character — Genie generates an interactive world anchored at that real location.



The Street View moment

- Three groups now share the same intuition — anchor a world model in *geo-registered Street View imagery*.
 - **SWM** ([Ours, Mar 2026](#)) — *kilometer-scale, trajectory-faithful* video simulation in a real city.
 - **CityRAG** ([Chou et al., Apr 2026](#)) — multi-city RAG over Street View *video clips*.
 - **Genie** ([Google, May 2026](#)) — *single-panorama anchoring* for *imaginative* world generation.
- Different framings: *navigate the real city* (SWM, CityRAG) vs. *reimagine a real place* (Genie). Same scaffold, different goal.

Conclusions

Applications and Impact of SWM

- Foundation model for real-world urban simulation
 - 👉 Training data generation for autonomous driving
- Data densification via street-view interpolation
 - 👉 Improving map coverage and localization datasets
- 3D reconstruction (3DGS/mesh) and virtual city applications
 - 👉 Digital twins for smart city and AR/VR applications
- VFX, what-if scenario generation, and large-scale simulation
 - 👉 Simulating rare events (e.g., accidents, extreme weather) for urban-scale planning and risk analysis



Cosmos-Predict2.5 14B Demo

Future directions

Upper-bound from pseudo ground-truth

- View interpolation enables training but limits generation quality.
- Incorporating real video data could improve realism.

Temporal inconsistency in street-view data

- Arbitrary gaps between captures introduce artifacts.
- Cross-temporal pairing mitigates this, but it was not sufficient.

Limited modeling of dynamic objects

- Moving entities (e.g., vehicles) are not explicitly modeled.
- Limits applicability to interactive scenarios (e.g., autonomous driving)



Q&A

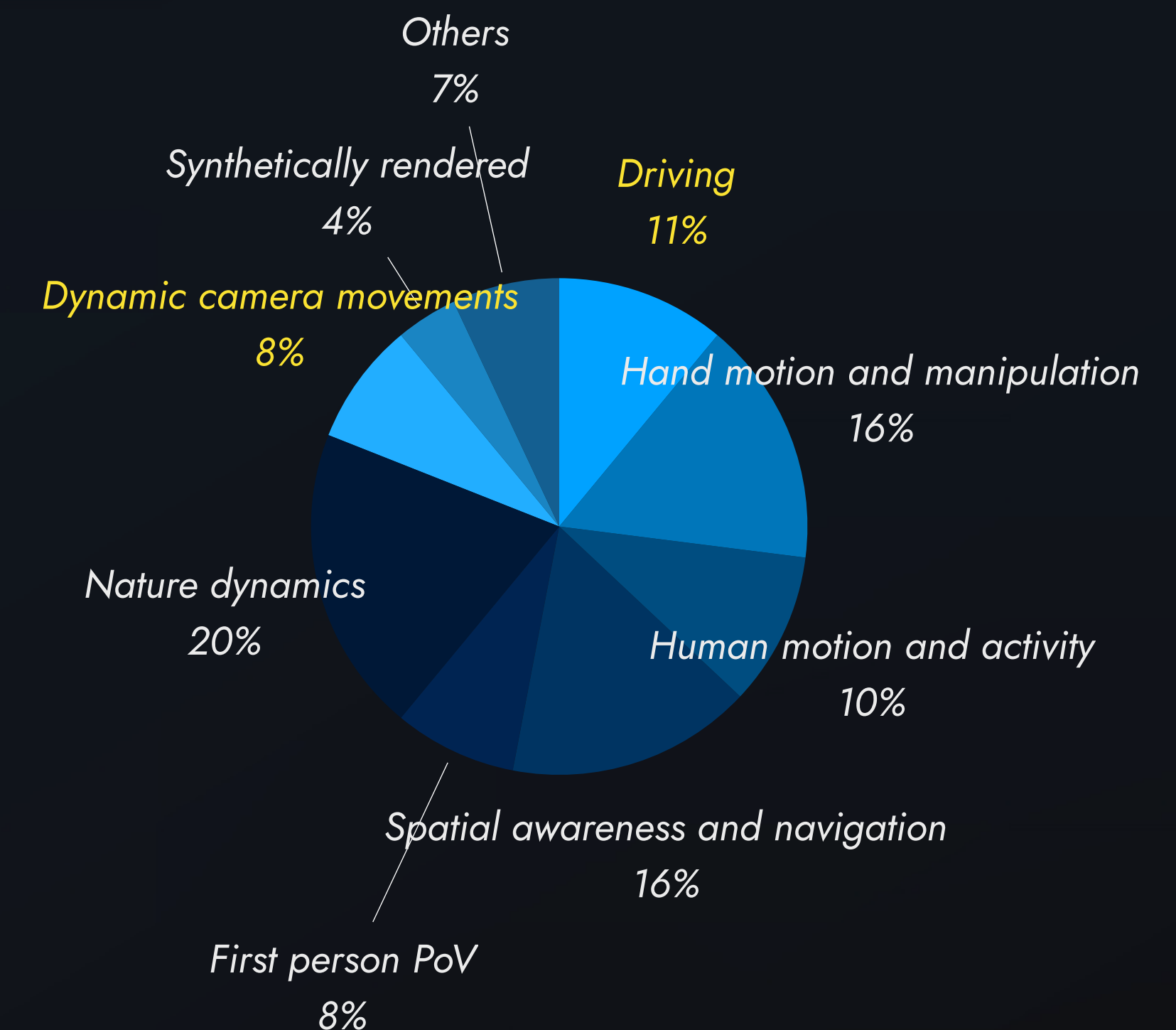
Thank you for your attention!



Appendix

World Foundation Models

- *Cosmos-Predict2.5-2B-12W* is pretrained on general-purpose real-world video data.
- A highly performant pre-trained WFM purpose-built for generating physics-aware world states for physical AI development.
- *20M hours* of raw videos with resolution from 720p to 4k.
- *10,000* NVIDIA H100 GPUs in a time span of *3 months*
- Free, commercial-use-friendly license



RAG details

- Retrieval conditions:
 - Nearest-neighbor search identifies candidate street-view locations along the target trajectory
 - Depth-based reprojection* filtering retains only those whose projected pixels exceed a coverage threshold in the nearest target view.
- Retrieval results are K pinhole references, the corresponding camera poses, and depth estimates in the world coordinate system.
 - 👉 *Remind that all reference and target images in a chunk are fed into DA3 to obtain the corresponding depth maps and camera poses in a shared space.*
- We'll cover how to use it shortly.

*It will be discussed later in the geometric referencing.

Comparisons with other methods



Other world models tend to struggle with scene coherence, trajectory alignment, or long-horizon stability.