Beyond Distribution Matching: Self-Supervised Representation Forcing for Few-Step Video Generation

Junyi Chen[1,*] **Haiyu Zhang**[2] **Zhoujie Fu**[3] **Tengfei Wang**[2,†] **Chunchao Guo**[2]

[1]Shanghai Jiao Tong University [2]Hunyuan, Tencent [3]Nanyang Technological University

Abstract

Few-step distillation is essential for deploying modern video diffusion models. Among existing approaches, Distribution Matching Distillation (DMD) has emerged as a highly promising paradigm by achieving strong output-space fidelity without adversarial training, while remaining flexible across various student architectures. We observe, however, that DMD fundamentally relies on an output-space objective, lacking explicit regularization on the model's internal representations. As the distillation step count aggressively shrinks, this output-only supervision becomes insufficient, causing the distilled student to degrade in both fine-grained spatial structures and coherent temporal dynamics. To address this limitation, we introduce Self Flow Forcing (SF^2), which incorporates a predictive representation loss into the DMD framework without altering its output objective. By providing the student and an Exponential Moving Average (EMA) teacher with heterogeneous noise levels, we establish an information asymmetry. This formulation forces the student to predict the teacher's internal representations from a more corrupted input, effectively integrating self-supervised representation alignment into distribution matching. Extensive experiments demonstrate that SF^2 effectively mitigates temporal flickering and consistently improves both spatial fidelity and motion quality. Furthermore, the proposed mechanism is architecture-agnostic—seamlessly supporting both bidirectional and causal models—and requires no external feature extractors, introducing minimal computational overhead.

Project page: https://sotamak1r.github.io/SF2-web/

Code: https://github.com/SOTAMak1r/SF2/

1 Introduction

Video diffusion models [4, 5, 10, 15, 34] have recently delivered impressive advances in synthesizing high-fidelity, dynamic visual content. Yet, their iterative generation process typically requires tens to hundreds of denoising steps, posing a significant obstacle to real-world deployment. To bridge this gap, various few-step distillation paradigms have been explored, including Consistency Models [28, 36], Trajectory Straightening [26, 27], and Adversarial Distillation [33]. Among these, Distribution Matching Distillation (DMD) [48, 49] has emerged as a highly promising approach by achieving strong output-space fidelity, while seamlessly supporting various student architectures.

However, when DMD is applied to the video domain, a characteristic degradation emerges at very low step counts. Specifically, the few-step distilled student suffers along two critical axes of video quality: fine-grained spatial structures (e.g., text rendering, complex geometry, and small object details) become blurred or inconsistent, and motion exhibits pronounced temporal flickering and physically unnatural dynamics. We observe this dual degradation in both our bidirectional and causal

[*]Work done during an internship at Hunyuan, Tencent.

[†]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

baselines, suggesting that it stems from the underlying distillation formulation itself rather than specific architectural choices.

We attribute this degradation to DMD's exclusive reliance on output-space supervision. DMD trains the student by matching its marginal samples to the teacher's at the output level, but it lacks explicit regularization to enforce the alignment of internal representations. In standard multi-step diffusion, this output-level proxy is sufficient, as the student can gradually refine its predictions and absorb fine-grained structures through iterative denoising. As the step budget aggressively shrinks, however, this proxy becomes inadequate. The student lacks the iterative capacity to recover rich, structured internal features, leading to the observed deterioration in both spatial fidelity and temporal coherence.

A complementary perspective from self-supervised representation learning offers a natural solution to this structural degradation. Classic predictive frameworks, from masked autoencoding [12, 39] to V-JEPA [1], have established that representation-level prediction enforces robust spatiotemporal understanding. Recent methods [6, 19, 51] have successfully adapted this principle for generative models, proving that internal feature regularization directly translates to higher generation fidelity. DMD, by contrast, is entirely devoid of such explicit representation-level guidance, optimizing only the marginal output distribution.

Motivated by this observation, we introduce SF^2, a novel paradigm that injects a predictive representation objective directly into the DMD framework without altering its output loss. The core mechanism is to establish an information asymmetry between the student and an Exponential Moving Average (EMA) teacher. By feeding the two networks with heterogeneous noise levels, we present the student with a more corrupted view of the underlying sequence and force it to predict the teacher's robust internal representations. This formulation explicitly integrates self-supervised representation alignment into distribution matching. Importantly, our mechanism requires no external feature extractors—the teacher is a self-bootstrapped EMA of the student—and adds only one extra forward pass per iteration, remaining agnostic to the specific architectural form of the generator.

We validate SF^2 across two representative video distillation settings: a bidirectional text-to-video student and a causal autoregressive student, building upon the standard ODE-initialization protocols [53]. In both scenarios, SF^2 effectively mitigates temporal flickering and consistently improves both spatial fidelity and motion quality with minimal computational overhead.

In summary, our main contributions are:

- We identify a critical limitation of applying standard DMD to video generation: its exclusive reliance on output-space supervision lacks explicit regularization for the student's internal representations. In the few-step regime, this results in the widely observed spatial artifacts and temporal flickering.
- We propose SF^2, which incorporates a predictive representation objective into the DMD framework. By utilizing heterogeneous noise levels and an EMA teacher to create an information asymmetry, we force the student to predict the teacher's internal representations from a more corrupted view, effectively integrating representation-level alignment with distribution matching.
- Extensive experiments on bidirectional and causal video distillation settings demonstrate that SF^2 consistently enhances spatial fidelity and temporal coherence, all while requiring no external feature extractors and introducing minimal computational overhead.

2 Related Work

Self Flow Forcing sits at the intersection of two rapidly evolving domains: the acceleration of video diffusion models through step distillation, and the enhancement of generative paradigms via self-supervised representation learning. We briefly review both areas and position our contribution.

2.1 Efficient Video Distillation

Step distillation aims to compress the iterative sampling process of diffusion models into a few steps. In the image domain, paradigms such as Progressive Distillation [32], Consistency Models [28, 36], Adversarial Diffusion Distillation [33], flow-based trajectory straightening [26, 27], and Distribution Matching Distillation (DMD) [48, 49] have established the feasibility of few-step generation. These acceleration frameworks have subsequently been extended to the video domain. This includes

bidirectional text-to-video adaptations such as VideoLCM [44], T2V-Turbo [24], and PCM [42]. In parallel, causal autoregressive generation frameworks have been accelerated through CausVid [50], Self-Forcing [16], and Causal-Forcing [53]. However, a persistent commonality across these methods is their exclusive reliance on marginal output distribution supervision, leaving the student's internal representations completely unconstrained. Self Flow Forcing addresses this limitation by introducing a representation-level objective that operates orthogonally to the existing output objective.

2.2 Self-Supervised Representation Learning in Generative Models

A growing body of work demonstrates that integrating representation-level priors into generative training yields cleaner internal features and superior generation quality. This integration typically follows two distinct routes. The first route injects semantic priors directly into the latent space: methods such as VA-VAE [47], SemanticGen [2] and RAE [52] align or replace the standard VAE latents with rich semantic features from frozen external perception encoders [3, 29, 40]. The second route operates within the generative backbone itself: REPA [23, 51] aligns the intermediate layers of a diffusion transformer with these frozen external encoders during training. To remove the dependency on external extractors, SRA [19] provides representation guidance via self-distillation, while Self-Flow [6] integrates a predictive objective by feeding heterogeneous noise levels across tokens, creating an information asymmetry the network must inherently resolve.

This predictive, asymmetric paradigm is particularly crucial for the video domain. As established by architectures like V-JEPA [1] and VideoMAE [39], learning to predict the representations of masked (perturbed) observations serves as a powerful inductive bias for capturing complex spatiotemporal dynamics. However, while the aforementioned representation strategies successfully enhance the training of base multi-step generative models —which operate with large step budgets and extensive data —their potential within the highly compressed regime of few-step model distillation remains entirely unexplored. Self Flow Forcing bridges this gap. By porting the predictive representation-forcing principle into video distillation via asymmetric noise, we force the student to capture robust spatiotemporal dynamics, effectively repairing the degradation of DMD without relying on external encoders.

3 Preliminaries

Modern video generators are dominated by diffusion and flow-matching models [25, 31], which produce high-quality samples at the cost of tens to hundreds of sequential evaluation steps. Distribution Matching Distillation (DMD) addresses this inference bottleneck by compressing such multi-step generators into few-step students via score matching in the output space. For causal autoregressive students, Self-Forcing [16] further resolves the train–test mismatch. We briefly review both below.

Diffusion models. Given clean data $x_0 \sim p_{\text{data}}$, a generative forward process interpolates between the data distribution and a tractable noise distribution [14, 35]:

$$x_t \;=\; \alpha_t\, x_0 \;+\; \sigma_t\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \qquad t \in [0, T], \tag{1}$$

where the schedule (α_t, σ_t) transports x_0 at $t{=}0$ to nearly pure noise at $t{=}T$. Flow matching [25], the de facto choice for modern video generators, fits the same template under a linear schedule. A neural network is trained to invert this process by predicting an interconvertible target—such as the noise ϵ, the clean data x_0, or the velocity v—all of which fundamentally parameterize the score $\nabla_{x_t} \log p_t(x_t)$ [21, 35]. At inference time, however, generation requires integrating this vector field over tens to hundreds of timesteps, creating a severe sampling bottleneck.

Distribution Matching Distillation. DMD [48, 49] addresses this bottleneck by training a few-step student generator G_θ. It maps a noisy latent x_t at timestep t directly to a clean prediction $\hat{x}_0 = G_\theta(x_t, t)$, such that the student's output distribution p_θ matches the target distribution p_{real} implied by the frozen multi-step teacher. To bypass intractable high-dimensional distribution matching, DMD minimizes the reverse KL divergence:

$$\mathcal{L}_{\text{DMD}} \;=\; \mathbb{E}_t[D_{\text{KL}}(p_\theta \,\|\, p_{\text{real}})] \tag{2}$$

through score matching at intermediate noise levels. Two score networks are maintained alongside the student: a frozen real-score critic $s_{\text{real}}(x_t, t)$ inherited from the teacher, and a learnable fake-score

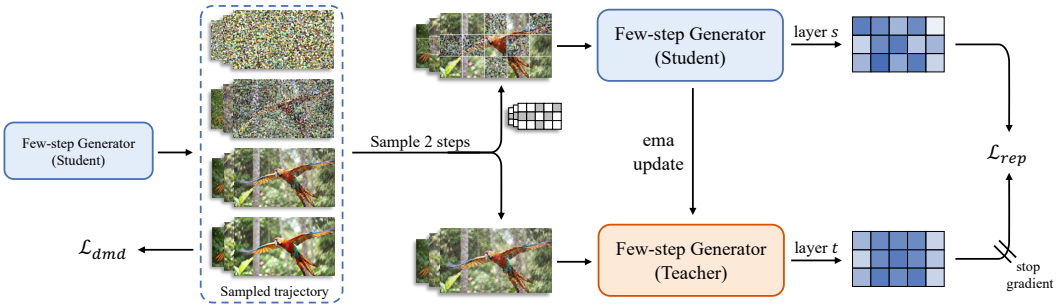


Figure 1: **Overview of SF^2.** From the student's denoising trajectory, the cleanest prediction drives the standard DMD loss $\mathcal{L}_{\text{DMD}}$, while two intermediate states (x_t, x_s) with $t > s$ feed the representation-forcing branch: a spatial mask mixes them into an asymmetric, highly corrupted input for the student, while the EMA teacher sees only the cleaner x_s. Hidden states from asymmetric layers are aligned via learnable projectors with a stop-gradient cosine loss $\mathcal{L}_{\text{rep}}$. The teacher is updated by EMA and requires no external pre-trained features.

critic $s_\psi(x_t, t)$ tracking the student's current output distribution and is updated via the standard denoising loss. The resulting DMD gradient applied to the student is:

$$\nabla_\theta \mathcal{L}_{\text{DMD}} = \mathbb{E}_{x_t,t}\left[\omega(t)\left(s_\psi(x_t,t) - s_{\text{real}}(x_t,t)\right)\nabla_\theta G_\theta(x_t,t)\right], \tag{3}$$

where $\omega(t)$ is a timestep weighting. Intuitively, this gradient pushes the student's outputs away from regions where its own samples are more likely than the teacher's. Critically for our motivation, this learning signal is derived entirely from the marginal output space $\hat{x}_0$, leaving the student's intermediate representation layers unregularized.

Self-Forcing for causal video distillation. While standard DMD is sufficient for bidirectional video distillation, causal autoregressive video generators face an additional context-conditioning challenge. Standard teacher forcing conditions every generated chunk on clean ground-truth history, causing severe error accumulation during autoregressive inference. Diffusion forcing [7] relaxes this by assigning independent noise levels per chunk, yet the conditioning history still relies on real data rather than the model's own predictions. Self-Forcing [16] closes this train–test gap by rolling the student out chunk-by-chunk on its own previously generated context. By propagating information via a key-value (KV) cache, training and inference share the exact same conditioning distribution. We adopt this protocol as the foundation for our causal autoregressive experiments.

4 Self Flow Forcing (SF^2)

In this section, we present SF^2, a framework for few-step video distillation. As identified in Sec. 3, a fundamental limitation of DMD is its exclusive reliance on output-level supervision, leaving the internal representations entirely unconstrained. Under aggressive few-step compression, this unconstrained freedom becomes harmful: the student tends to converge to shortcut representations that merely minimize the marginal distribution loss, fundamentally lacking temporal and structural coherence. SF^2 closes this gap by aligning the student's intermediate features with those of an exponential-moving-average (EMA) teacher using deliberately asymmetric inputs, while preserving the original DMD output objective. Concretely, we maintain three modules during training: the student G_θ, an EMA teacher $G_{\theta'}$ with identical architecture but no gradients, and the standard DMD critics s_ψ and s_{real}. The full training objective is formulated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{DMD}} + \lambda_{\text{rep}}\,\mathcal{L}_{\text{rep}}, \tag{4}$$

where $\mathcal{L}_{\text{rep}}$ reuses the rollout that DMD already performs. Per generator update, $\mathcal{L}_{\text{rep}}$ therefore requires one additional pass of the student (with gradients) and one of the EMA teacher (without gradients) over the asymmetric inputs; this cost does not grow with the number of sampling steps.

Following the pipeline illustrated in Fig. 1, we structure the remainder of this section around three core components. We first detail the construction of asymmetric inputs from a shared trajectory (Sec. 4.1). We then specify the representation loss $\mathcal{L}_{\text{rep}}$ (Sec. 4.2). Finally, we detail the EMA teacher update mechanism and projector warmup that stabilize this self-bootstrapped alignment (Sec. 4.3).

4.1 Asymmetric Inputs from a Shared Trajectory

In standard generative training, uniform noise is typically applied across all spatial-temporal tokens. The resulting denoising objective can often be minimized by exploiting low-level, local correlations, providing little incentive for the model to capture deep semantic structures. To break this local dependency, we draw inspiration from Self-Flow [6] and introduce information asymmetry. The core idea is to apply heterogeneous noise levels across different tokens, forcing the student to proactively infer heavily corrupted regions using the global context provided by cleaner tokens.

However, introducing this asymmetry naively—such as completely masking out a subset of tokens or assigning independent, randomized noise levels to every token—severely disrupts the underlying generative dynamics and creates a significant train–inference mismatch. To safely establish this asymmetry without compromising the generative pipeline, we adopt a dual-timestep formulation. Starting from a shared denoising trajectory, the student receives a token-level mixture of a noisier state and a cleaner state, whereas the teacher observes exclusively the cleaner state. The student must thus predict the teacher's representations from a strictly more corrupted view, avoiding trivial self-reconstruction.

Two trajectory states. The DMD generator loss already unrolls the student over K denoising steps $t_K > \cdots > t_1$,

$$\hat{x}_0^{(k)} = G_\theta(x_{t_k}, t_k), \qquad x_{t_{k-1}} = \alpha_{t_{k-1}}\,\hat{x}_0^{(k)} + \sigma_{t_{k-1}}\,\epsilon^{(k)}, \tag{5}$$

producing a denoising trajectory $\{(x_{t_k}, t_k)\}_{k=1}^{K}$ of progressively cleaner latents. From this trajectory, we sample two distinct steps (t, s) with $t > s$, giving a noisier latent x_t and a cleaner latent x_s. Since DMD is data-free, reusing this trajectory yields valid intermediate states without an extra generative rollout.

Per-token spatial masking. Let the latent video feature have F frames, each tokenized into an $H_p \times W_p$ patch grid. We sample a binary mask $M \in \{0,1\}^{B\times F\times H_p\times W_p}$ where each entry drawn as $\mathrm{Bern}(\rho)$, $\rho \sim \mathcal{U}(0, \rho_{\max})$. The student input is formulated as a per-token mixture of the two trajectory states, whereas the teacher observes exclusively the cleaner state:

$$\tilde{x}^{\text{stu}} = M \odot x_s + (1 - M) \odot x_t, \qquad \tilde{x}^{\text{tea}} = x_s. \tag{6}$$

To properly reflect this mixture, the timestep conditioning for the student is applied heterogeneously per token—using s where $M = 1$ and t where $M = 0$—whereas the teacher is uniformly conditioned on s. Consequently, the student receives a random mosaic of cleaner and noisier tokens. To match the teacher, the student is forced to aggregate context from cleaner spatial-temporal neighbors to infer the missing semantics of the corrupted regions.

4.2 Representation Forcing Loss

Given the asymmetric inputs formulated in Sec. 4.1, we extract hidden states from two index sets of transformer blocks [11, 30] $\mathcal{S} = \{s_1, \ldots, s_L\}$ (student) and $\mathcal{T} = \{t_1, \ldots, t_L\}$ (teacher), each yielding tensors of shape $[B, N, D]$. We deliberately select teacher layers that are deeper than their corresponding student layers. As observed in recent self-supervised generative frameworks [6, 19], robust semantic representations naturally emerge in deeper network layers. Drawing targets from deeper teacher layers thus provides a stronger semantic distillation signal, while applying the loss to shallower student layers prevents the representation objective from interfering with the model's final generative outputs.

To bridge the representation spaces, a per-layer projector $\pi_\ell(\cdot)$ maps the student features into the teacher's domain:

$$\tilde{h}_{s_\ell}^{\text{stu}} = \pi_\ell\big(h_{s_\ell}^{\text{stu}}\big), \tag{7}$$

which is implemented as a multi-layer perceptron (MLP) with input/output widths equal to the backbone hidden size D and an internal hidden width of $4D$. We minimize the negative cosine similarity between projected student features and detached teacher features, averaged over tokens and layer pairs:

$$\mathcal{L}_{\text{rep}} = \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{E}_n\Big[1 - \cos\big(\tilde{h}_{s_\ell,n}^{\text{stu}},\ \mathrm{sg}[h_{t_\ell,n}^{\text{tea}}]\big)\Big], \tag{8}$$

Table 1: **Per-cycle training-time breakdown (% of one cycle).** One cycle = 1 generator update + 5 critic updates. The SF^{2}branch is only $\sim$6% of the cycle.

	Generator update ($\times$1)				Critic update ($\times$5)			Total
Component	Rollout	DMD	**Ours**	Backward	Rollout	MSE	Backward	
% of cycle	4.4	20.1	**6.2**	6.4	25.0	8.7	29.2	100.0

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator. We opt for cosine similarity over ℓ_2 distance because the EMA teacher and the student inherently operate at slightly different feature magnitude scales, making directional alignment more robust and appropriate.

4.3 EMA Teacher and Projector Warmup

EMA teacher. The teacher model $G_{\theta'}$ is initialized as a copy of the student and updated by the standard EMA rule

$$\theta' \leftarrow \mu\,\theta' + (1-\mu)\,\theta, \tag{9}$$

with the momentum μ set close to 1 (e.g., 0.99). Consequently, the teacher tracks a slowly moving, temporally smoothed version of the student. Furthermore, deriving the teacher directly from the student's own parameter history ensures that the target representations remain strictly compatible with the model's native capacity, avoiding the distribution mismatch when distilling from disparate external models while eliminating the need for pre-trained vision encoders.

Projector warmup. Since distillation initializes both the student and the EMA teacher from a pre-trained backbone, whereas the projectors $\{\pi_\ell\}$ are initialized from scratch, the projectors inevitably output near-random vectors during the earliest training steps. Enabling the full objective at step 0 would undesirably backpropagate projector noise into the pretrained backbone and corrupt the delicate features we intend to align. To circumvent this, we implement a T_{proj}-step projector warmup phase. During this phase, $\mathcal{L}_{\text{DMD}}$ is temporarily disabled, and the student's hidden states $h^{\text{stu}}_{s_\ell}$ are detached before entering the projectors π_ℓ. Thus, only the projectors receive gradients. The teacher is kept strictly identical to the student, ensuring both backbones remain stationary while the projectors reliably calibrate against fixed targets. After the warmup concludes at step T_{proj}, the full objective (Eq. 4) is activated, and the EMA update (Eq. 9) begins. At this boundary, we also synchronize the projector learning rate with that of the backbone, ensuring all representation-related parameters optimize at a unified pace once the student is unfrozen.

5 Experiments

To evaluate the effectiveness of SF^2, we structure our experiments to answer three core questions: (1) How do the asymmetric masking strategies influence representation learning across different generation paradigms? (Sec. 5.2); (2) Does the representation alignment translate into perceptually coherent motions and detailed spatial structures? (Sec. 5.3); and (3) Does SF^2 quantitatively outperform recent state-of-the-art multi-step and distilled video generators? (Sec. 5.4).

5.1 Experimental Setup

Implementation details. We instantiate SF^2 across both bidirectional and causal paradigms to demonstrate its versatility. For the bidirectional setting, we employ two structurally distinct backbones: Wan2.1-T2V-1.3B [41] as our primary model, and HunyuanVideo-1.5 [45] to evaluate generalizability. Notably, HunyuanVideo-1.5 utilizes an MMDiT [11] architecture with strong text-rendering capabilities, serving as a stringent visual probe for spatial fidelity preservation during few-step distillation. For the causal autoregressive setting, we initialize the student from Causal-Forcing [53] ODE-regression checkpoint, enabling us to evaluate the chunk-by-chunk representation forcing under a genuine key-value (KV) cache rollout.

All configurations follow the established recipe [16, 48], wherein the two score critics are drawn from the student's model family. Specifically, for the Wan2.1-based students (both bidirectional and causal), we use the frozen Wan2.1-14B [41] as the real-score model and initialize the learnable fake-score model from Wan2.1-1.3B. For the HunyuanVideo-1.5 bidirectional student, the real-score model is the



Figure 2: **Qualitative comparison on causal few-step T2V generation.** Top row: multi-step bidirectional Wan2.1 teacher. Rows 2–5: few-step distilled causal students (CausVid, Self-Forcing, Causal-Forcing, SF^2). The four panels expose four facets of representation-level coherence that the rep loss is designed to compress into the student: articulated-motion plausibility (top-left), event irreversibility (top-right), object persistence (bottom-left), and scene-structural stability (bottom-right). SF^2 satisfies all four, whereas baselines fail in distinct ways on each.

frozen HunyuanVideo-1.5 [45] and the fake-score model is initialized from the same HunyuanVideo-1.5 checkpoint. We use $K{=}4$ denoising steps for both bidirectional and causal students. Student and teacher features are read at fixed depth ratios $\ell_\theta{=}0.3N_b$ and $\ell_{\theta'}{=}0.7N_b$, respectively, where N_b is the total number of transformer blocks. These features are aligned via an MLP projector (hidden dimension $4D$) with $\lambda_{\text{rep}}{=}0.5$. The EMA teacher uses momentum $\mu{=}0.99$ with a 100-step projector-only warmup, during which the projector's learning rate is amplified $50\times$. Optimization is performed using AdamW at $\text{lr}_G{=}2{\times}10^{-6}$ and $\text{lr}_C{=}4{\times}10^{-7}$ at a 5:1 critic-to-generator update ratio.

Training cost. To quantify the overhead of SF^2 on top of vanilla DMD, we profile one training cycle (1 generator update + 5 critic updates, per the 5:1 ratio) in Table 1. SF^2 branch (EMA teacher forward, projector, and cosine alignment) takes $\sim$6% of the cycle; the rest is the DMD loop the baseline already pays.

Evaluation metrics. Our primary evaluation protocol is VBench [17]: we run the official prompt suite and report the Total Score, Quality Score, and Semantic scores. We additionally report VisionReward [46], a learned human-preference model, on the curated 100-prompt set with rich motion and complex actions provided by [53], together with single-H100-equivalent throughput and latency. For ablations we use VBench Quality as the headline single-number proxy.

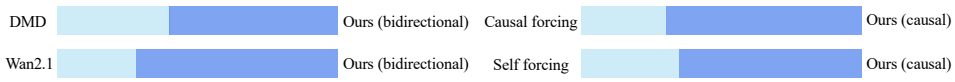
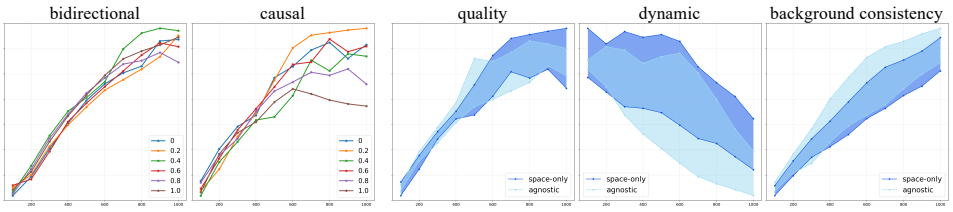


Figure 3: **Pairwise quantitative comparison.** All compared methods share the same Wan2.1-T2V-1.3B backbone. Each bar shows the per-prompt preference of VisionReward [46]; the dark-blue segment is the fraction of prompts on which SF^2 is preferred. SF^2 wins the majority of prompts against all four baselines.



(a) Masking ratio (b) Mask sampling

Figure 4: **Hyperparameter selection for asymmetric inputs. (a) Masking ratio $\rho_{\max}$.** The two paradigms peak at different optima — $\rho_{\max}$=0.4 for bidirectional and $\rho_{\max}$=0.2 for causal — which we adopt as defaults in the following experiments. **(b) Mask sampling pattern.** Space-only sampling (a single $H_p \times W_p$ mask broadcast across all frames) versus agnostic sampling (independent per-token masking) on VBench's Quality Score, Dynamic Degree, and Background Consistency.

5.2 Ablation Experiments

We study the two design choices introduced by the asymmetric input construction (Sec. 4.1): the masking ratio $\rho_{\max}$ and the mask sampling pattern. Curves report VBench [17] metric heads on a held-out validation set of 400 motion-rich prompts generated by GPT-5 versus training step, exposing both final performance and convergence.

Masking ratio. Fig. 4(a) sweeps the the masking ratio $\rho_{\max} \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ on both settings (left: bidirectional Wan2.1-1.3B; right: causal Wan2.1-1.3B initialized from Causal-Forcing). The bidirectional student peaks at $\rho_{\max}$=0.4 and the causal student at $\rho_{\max}$=0.2. One explanation for the smaller causal optimum is that each causal chunk is already conditioned on its previously generated history through the KV cache. The student thus has cleaner context even at small ρ, and additional clean tokens within the chunk make its input closer to the teacher's and the prediction task easier. The bidirectional student, in contrast, denoises the entire window without such history and benefits from more clean tokens within the window.

Table 2: **Quantitative comparison with existing methods on VBench [17] and VisionReward [46].** Models are grouped by sampling regime — bidirectional multi-step, distilled bidirectional, autoregressive multi-step, and distilled autoregressive — and **bold** marks the best score within each comparable group. Throughput (FPS) and latency (s) are single-H100-equivalent values. All distilled models (in both distilled groups) share the same Wan2.1-T2V-1.3B backbone.

Model	Params	Throughput ↑	Latency ↓	Resolution	Total ↑	Quality ↑	Semantic ↑	Vision. ↑
Bidirectional Video Diffusion Models								
LTX-Video [13]	1.9B	8.98	13.5	768×512	79.83	81.88	71.62	–6.218
Wan2.1 [41]	1.3B	0.78	103	848×480	83.37	84.30	79.65	5.275
Distilled Bidirectional Video Models								
CausVid [50]	1.3B	16.2	4.22	848×480	83.19	84.08	79.63	5.310
Ours	1.3B	16.2	4.22	848×480	**84.15**	**84.92**	81.07	<u>7.060</u>
Autoregressive Video Diffusion Models								
Pyramid Flow [20]	2B	6.70	2.5	640×384	80.75	83.41	70.11	4.055
SkyReels-V2 [8]	1.3B	0.49	112	960×540	81.97	83.96	74.01	3.584
MAGI-1 [38]	4.5B	0.19	282	848×480	78.88	81.67	67.72	0.773
Distilled Autoregressive Video Models								
CausVid [50]	1.3B	**17.0**	**0.69**	848×480	81.33	83.98	70.72	5.741
Self-Forcing [16]	1.3B	**17.0**	**0.69**	848×480	83.74	84.48	80.77	5.820
Causal-Forcing [53]	1.3B	**17.0**	**0.69**	848×480	84.04	84.59	<u>81.84</u>	6.326
Ours	1.3B	**17.0**	**0.69**	848×480	**84.15**	<u>84.67</u>	**82.06**	**7.523**



Figure 5: **Qualitative comparison on bidirectional few-step T2V generation.** Both compared methods are distilled from the same HunyuanVideo-1.5 backbone. Each pair shows the DMD student (top) and SF^2 (bottom) on the same prompt and noise; zoomed crops highlight the rendered text. SF^2 produces sharp, legible glyphs where DMD yields blurred or scrambled characters.

Mask sampling pattern. Fig. 4(b) compares the two mask sampling patterns [12]: (i) **space-only**, where a single $H_p \times W_p$ binary mask is sampled and broadcast across all frames in the window; and (ii) **agnostic**, where each spatio-temporal token is masked independently. To decouple the comparison from $\rho_{\max}$, we run each scheme on the causal student at $\rho_{\max} \in \{0.2, 0.4, 0.6, 0.8\}$ and plot the $[\min, \max]$ envelope across the four runs.

The two patterns reach comparable Quality (with space-only attaining a higher ceiling at convergence), but split on the two temporal-coherence metrics: space-only yields a higher Dynamic Degree, while agnostic produces a higher Background Consistency. We attribute this trade-off to a temporal-copy shortcut: under agnostic sampling, each masked token is likely left clean at the same spatial location in an adjacent frame, so the student can minimize the rep loss by copying along the time axis — biasing it toward over-similar adjacent frames, which inflates background consistency and suppresses motion. Space-only sampling closes the shortcut: with the same mask broadcast across all frames, every spatial location is either clean or masked throughout the entire window, forcing reconstruction from spatial neighbors. We adopt it as the default.

5.3 Qualitative Results

Causal autoregressive. Fig. 2 compares autoregressive rollouts on identical prompts and seeds for the multi-step Wan2.1 teacher and four few-step distilled causal students: CausVid [50], Self-Forcing [16], Causal-Forcing [53], and SF^2. The gains of SF^2 surface as four distinct facets of representation-level coherence across the figure. (Top-left) Articulated-motion plausibility: the soccer player's pose stays anatomically valid frame to frame, while baselines produce snapping or rerouted limbs. (Top-right) Event irreversibility: the destroyed wall stays destroyed, whereas baselines reconstitute or even regrow it across consecutive frames. (Bottom-left) Object persistence: the red ribbon remains a single continuous object throughout the dance, instead of vanishing, splitting, or warping. (Bottom-right) Scene-structural stability: horse and rider counts, identities, and overall layout stay coherent.

Bidirectional text-to-video generation. Fig. 5 compares two few-step HunyuanVideo-1.5 students — the plain DMD baseline and SF^2 — on prompts containing in-scene text. Since the HunyuanVideo-1.5 teacher renders non-trivial text, degradation under distillation directly probes retained fine spatial structure. The DMD student loses this ability, yielding blurred or scrambled glyphs and indicating that output-only supervision is too coarse for such structured spatial features, whereas SF^2 produces sharp, legible text across all prompts.

5.4 Quantitative Comparison

Table 2 reports the main comparison. At matched throughput and latency, SF^2 attains the best score on every reported metric within both the *Distilled Autoregressive* and *Distilled Bidirectional* groups.

Direct pairwise comparison. To check that the VisionReward gains in Table 2 hold per prompt rather than from a few outliers, we let VisionReward [46] pick the preferred video for each of the 100 motion-rich prompts in four pairings spanning both paradigms. As shown in Fig. 3, SF^2 is preferred on the majority of prompts against every baseline, including the Wan2.1 base model.

Probing representation quality. We curate 120 prompts depicting static scenes under smooth camera trajectories, reconstruct each generated video into a 3D Gaussian Splatting [22] representation

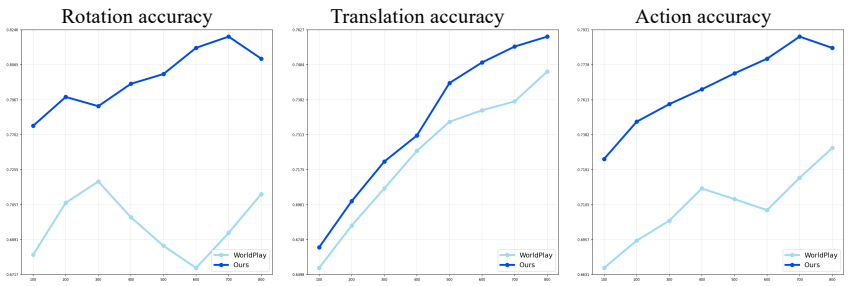


Figure 6: **Quantitative evaluation on interactive video generation.** Adherence of generated videos to the input camera and action conditions. SF^2 achieves higher translation, rotation, and action accuracy than the WorldPlay baseline across training steps.

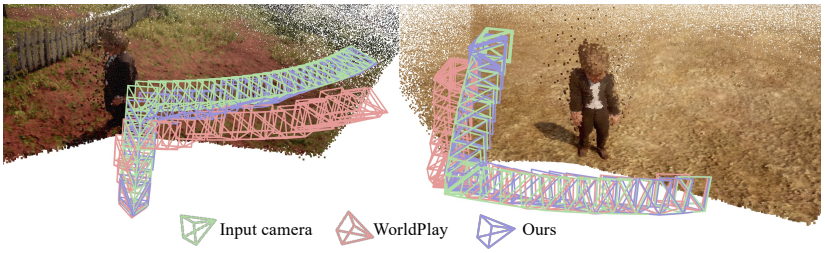


Figure 7: **Qualitative camera trajectory alignment.** Reconstructed 3D point clouds and camera trajectories from the generated videos. SF^2 (blue) tightly tracks the input condition (green), whereas the WorldPlay baseline (red) drifts.

Table 3: **Probing representation quality** via 3DGS re-rendering and VBench frame-level coherence (no 3D supervision in training).

Model	Params	Throughput	PSNR ↑	SSIM ↑	LPIPS ↓	Dynamic ↑	Flickering ↑	Consistency ↑
DMD (Ours w/o $\mathcal{L}_{\text{rep}}$)	8.3B	2.21	20.64	0.6982	0.2144	90.00	96.39	94.11
Ours	8.3B	2.21	**21.24**	**0.7110**	**0.2096**	90.00	**96.86**	**94.65**

via WorldMirror-2 [18], re-render along the original trajectory, and report PSNR / SSIM / LPIPS against the generated frames, alongside VBench Temporal Flickering and Background Consistency. On top of HunyuanVideo-1.5, SF^2 improves over vanilla DMD on every reconstruction and coherence metric at matched Dynamic Degree (Table 3). As no 3D supervision is used, this gain stems from the feature-level alignment, indicating that SF^2 shapes the internal representation rather than only the output distribution.

Application: Interactive long video generation. We further apply SF^2 to interactive video generation [9] with WorldPlay [37], a causal autoregressive model conditioned on camera and action inputs whose original distillation relies on output-space distillation akin to Self-Forcing [16]. We replace this distillation with SF^2, condition both models on input camera trajectories, recover the synthesized camera poses from the generated long videos with WorldMirror-2 [18], and measure their alignment with the input conditions. SF^2 attains higher translation, rotation, and action accuracy throughout training (Fig. 6), and its trajectories closely follow the input path where the baseline drifts (Fig. 7). Viewing interactive generation as state-transition learning, a more coherent internal representation yields more precise transitions and tighter adherence to control signals.

6 Conclusion

We presented SF^2, which closes a structural blind spot of DMD by adding a predictive representation objective on top of its output loss. Through asymmetric inputs and a self-bootstrapped EMA teacher, the student aligns its internal features with the teacher's under information loss. The mechanism is paradigm-agnostic, adds negligible cost, and consistently improves spatial fidelity and temporal coherence across bidirectional and causal generators on distinct backbones. We view SF^2 as evidence that, in the few-step regime, what the student learns to represent matters as much as what it learns to output.

Acknowledgments and Disclosure of Funding

This work was done during Junyi Chen's research internship at Hunyuan, Tencent, which provided computational resources and research support.

References

[1] M. Assran, A. Bardes, D. Fan, Q. Garrido, R. Howes, M. Komeili, M. Muckley, A. Rizvi, C. Roberts, K. Sinha, A. Zholus, S. Arnaud, A. Gejji, A. Martin, F. Robert Hogan, D. Dugas, P. Bojanowski, V. Khalidov, P. Labatut, F. Massa, M. Szafraniec, K. Krishnakumar, Y. Li, X. Ma, S. Chandar, F. Meier, Y. LeCun, M. Rabbat, and N. Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[2] J. Bai, X. Wu, X. Wang, X. Fu, Y. Zhang, Q. Wang, X. Shi, M. Xia, Z. Liu, H. Hu, et al. Semanticgen: Video generation in semantic space. *arXiv preprint arXiv:2512.20619*, 2025.

[3] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[4] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[5] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

[6] H. Chefer, P. Esser, D. Lorenz, D. Podell, V. Raja, V. Tong, A. Torralba, and R. Rombach. Self-supervised flow matching for scalable multi-modal synthesis. *arXiv preprint arXiv:2603.06507*, 2026.

[7] B. Chen, D. Martí Monsó, Y. Du, M. Simchowitz, R. Tedrake, and V. Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2025.

[8] G. Chen, D. Lin, J. Yang, C. Lin, J. Zhu, M. Fan, H. Zhang, S. Chen, Z. Chen, C. Ma, et al. Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025.

[9] J. Chen, H. Zhu, X. He, Y. Wang, J. Zhou, W. Chang, Y. Zhou, Z. Li, Z. Fu, J. Pang, et al. Deepverse: 4d autoregressive video generation as a world model. *arXiv preprint arXiv:2506.01103*, 2025.

[10] J. Chen, T. He, Z. Fu, P. Wan, K. Gai, and W. Ye. Vino: A unified visual generator with interleaved omnimodal context. *arXiv preprint arXiv:2601.02358*, 2026.

[11] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[12] C. Feichtenhofer, H. Fan, Y. Li, and K. He. Masked autoencoders as spatiotemporal learners. *arXiv:2205.09113*, 2022.

[13] Y. HaCohen, N. Chiprut, B. Brazowski, D. Shalem, D. Moshe, E. Richardson, E. Levin, G. Shiran, N. Zabari, O. Gordon, P. Panet, S. Weissbuch, V. Kulikov, Y. Bitterman, Z. Melumian, and O. Bibi. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

[14] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[15] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models. *arXiv:2204.03458*, 2022.

[16] X. Huang, Z. Li, G. He, M. Zhou, and E. Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

[17] Z. Huang, Y. He, J. Yu, F. Zhang, C. Si, Y. Jiang, Y. Zhang, T. Wu, Q. Jin, N. Chanpaisit, Y. Wang, X. Chen, L. Wang, D. Lin, Y. Qiao, and Z. Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[18] T. HY-World. Hy-world 2.0: A multi-modal world model for reconstructing, generating, and simulating 3d worlds. *arXiv preprint*, 2026.

[19] D. Jiang, M. Wang, L. Li, L. Zhang, H. Wang, W. Wei, G. Dai, Y. Zhang, and J. Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. *arXiv preprint arXiv:2505.02831*, 2025.

[20] Y. Jin, Z. Sun, N. Li, K. Xu, K. Xu, H. Jiang, N. Zhuang, Q. Huang, Y. Song, Y. Mu, and Z. Lin. Pyramidal flow matching for efficient video generative modeling. 2024.

[21] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.

[22] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

[23] X. Leng, J. Singh, Y. Hou, Z. Xing, S. Xie, and L. Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. *arXiv preprint arXiv:2504.10483*, 2025.

[24] J. Li, W. Feng, T.-J. Fu, X. Wang, S. Basu, W. Chen, and W. Y. Wang. T2v-turbo: Breaking the quality bottleneck of video consistency model with mixed reward feedback. *Advances in neural information processing systems*, 37:75692–75726, 2024.

[25] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[26] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[27] X. Liu, X. Zhang, J. Ma, J. Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023.

[28] S. Luo, Y. Tan, L. Huang, J. Li, and H. Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

[29] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[30] W. Peebles and S. Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

[31] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[32] T. Salimans and J. Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[33] A. Sauer, D. Lorenz, A. Blattmann, and R. Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024.

[34] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

[35] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

[36] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[37] W. Sun, H. Zhang, H. Wang, J. Wu, Z. Wang, Z. Wang, Y. Wang, J. Zhang, T. Wang, and C. Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world model. *arXiv preprint*, 2025.

[38] H. Teng, H. Jia, L. Sun, L. Li, M. Li, M. Tang, S. Han, T. Zhang, W. Zhang, W. Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.

[39] Z. Tong, Y. Song, J. Wang, and L. Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022.

[40] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[41] A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, J. Wang, J. Zhang, J. Zhou, J. Wang, J. Chen, K. Zhu, K. Zhao, K. Yan, L. Huang, M. Feng, N. Zhang, P. Li, P. Wu, R. Chu, R. Feng, S. Zhang, S. Sun, T. Fang, T. Wang, T. Gui, T. Weng, T. Shen, W. Lin, W. Wang, W. Wang, W. Zhou, W. Wang, W. Shen, W. Yu, X. Shi, X. Huang, X. Xu, Y. Kou, Y. Lv, Y. Li, Y. Liu, Y. Wang, Y. Zhang, Y. Huang, Y. Li, Y. Wu, Y. Liu, Y. Pan, Y. Zheng, Y. Hong, Y. Shi, Y. Feng, Z. Jiang, Z. Han, Z.-F. Wu, and Z. Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[42] F.-Y. Wang, Z. Huang, A. W. Bergman, D. Shen, P. Gao, M. Lingelbach, K. Sun, W. Bian, G. Song, Y. Liu, et al. Phased consistency model. *arXiv preprint arXiv:2405.18407*, 2024.

[43] W. Wang and Y. Yang. Vidprom: A million-scale real prompt-gallery dataset for text-to-video diffusion models. 2024. URL https://openreview.net/forum?id=pYNl76onJL.

[44] X. Wang, S. Zhang, H. Zhang, Y. Liu, Y. Zhang, C. Gao, and N. Sang. Videolcm: Video latent consistency model. *arXiv preprint arXiv:2312.09109*, 2023.

[45] B. Wu, C. Zou, C. Li, D. Huang, F. Yang, H. Tan, J. Peng, J. Wu, J. Xiong, J. Jiang, et al. Hunyuanvideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025.

[46] J. Xu, Y. Huang, J. Cheng, Y. Yang, J. Xu, Y. Wang, W. Duan, S. Yang, Q. Jin, S. Li, et al. Visionreward: Fine-grained multi-dimensional human preference learning for image and video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 11269–11277, 2026.

[47] J. Yao, B. Yang, and X. Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[48] T. Yin, M. Gharbi, T. Park, R. Zhang, E. Shechtman, F. Durand, and W. T. Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024.

[49] T. Yin, M. Gharbi, R. Zhang, E. Shechtman, F. Durand, W. T. Freeman, and T. Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024.

[50] T. Yin, Q. Zhang, R. Zhang, W. T. Freeman, F. Durand, E. Shechtman, and X. Huang. From slow bidirectional to fast autoregressive video diffusion models. In *CVPR*, 2025.

[51] S. Yu, S. Kwak, H. Jang, J. Jeong, J. Huang, J. Shin, and S. Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

[52] B. Zheng, N. Ma, S. Tong, and S. Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

[53] H. Zhu, M. Zhao, G. He, H. Su, C. Li, and J. Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026.

Algorithm 1 Training of Self Flow Forcing.

Require: Denoising timesteps $\{t_1, \dots, t_K\}$ with $t_1 > \cdots > t_K$
Require: Student generator G_θ, EMA teacher $G_{\theta'}$ ($\theta' \leftarrow \theta$)
Require: Fake-score critic s_θ, frozen real-score s_{target}
Require: Per-layer projectors $\{\pi_\ell\}_{\ell=1}^{L}$, layer indices $\mathcal{S}, \mathcal{T}$
Require: Loss weight λ_{rep}, mask ratio $\rho_{\max}$, EMA momentum μ
Require: Warmup steps $T_{\text{proj}}, T_{\text{ema}}$
$k \leftarrow 0$
loop
  // (1) DMD rollout with trajectory recording
  Sample noise $z \sim \mathcal{N}(0, \mathbf{I})$ and condition c
  Roll out G_θ along $\{t_1, \dots, t_K\}$ and record $\mathcal{R} = \{(x_{t_i}, t_i)\}_{i=1}^{K}$, final prediction $\hat{x}_0$
  // (2) Heterogeneous noise mixing
  Sample two distinct steps (t, s) from $\mathcal{R}$ with $t > s$
  Sample masking ratio $\rho \sim \mathcal{U}(0, \rho_{\max})$
  Sample per-token mask $M \in \{0,1\}^{B \times F \times H_p \times W_p}$, $M_{ijkl} \sim \text{Bern}(\rho)$
  $\tilde{x}^{\text{stu}} \leftarrow M \odot x_s + (1 - M) \odot x_t, \quad \tilde{x}^{\text{tea}} \leftarrow x_s$
  // (3) Asymmetric forward and feature extraction
  $\{h^{\text{stu}}_{s_\ell}\} \leftarrow G_\theta(\tilde{x}^{\text{stu}}, t, c\,;\, \mathcal{S})$ ▷ with grad; chunk-AR if causal
  $\{h^{\text{tea}}_{t_\ell}\} \leftarrow G_{\theta'}(\tilde{x}^{\text{tea}}, s, c\,;\, \mathcal{T})$ ▷ no grad
  // (4) Loss computation
  if $k < T_{\text{proj}}$ **then** ▷ Projector-only warmup: only $\{\pi_\ell\}$ receives gradient
    $\mathcal{L}_{\text{DMD}} \leftarrow 0$; detach $h^{\text{stu}}_{s_\ell}$ before π_ℓ for all ℓ
  else
    Compute $\mathcal{L}_{\text{DMD}}$ from $\hat{x}_0$ via score matching with s_θ, s_{target}
  end if
  $\mathcal{L}_{\text{rep}} \leftarrow \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{E}_n \left[1 - \cos\big(\pi_\ell(h^{\text{stu}}_{s_\ell, n}),\, \text{sg}[h^{\text{tea}}_{t_\ell, n}]\big)\right]$
  $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{DMD}} + \lambda_{\text{rep}} \mathcal{L}_{\text{rep}}$
  // (5) Parameter updates
  Update θ and $\{\pi_\ell\}$ by backpropagating $\mathcal{L}_{\text{total}}$
  Update fake-score critic s_θ on G_θ's rollout (denoising loss)
  if $k < T_{\text{ema}}$ **then**
    $\theta' \leftarrow \theta$ ▷ Direct copy: avoids fp32 EMA drift
  else
    $\theta' \leftarrow \mu\, \theta' + (1 - \mu)\, \theta$ ▷ Standard EMA
  end if
  $k \leftarrow k + 1$
end loop

Appendix

A Training Algorithm

Algorithm 1 gives one training iteration of SF^2. Each iteration reuses the DMD rollout to record the denoising trajectory, builds the asymmetric student and teacher inputs from two of its states (Sec. 4.1), computes $\mathcal{L}_{\text{total}}$ (Eq. 4), and updates the student, the projectors, the fake-score critic, and the EMA teacher. During the first T_{proj} steps only the projectors are trained (Sec. 4.3). During the first T_{ema} steps the teacher is copied from the student instead of updated by EMA; we set $T_{\text{ema}}{=}T_{\text{proj}}$ in all experiments.

B Application to Bidirectional and Causal Generators

SF^2 only requires that hidden states can be read from given transformer blocks of the generator. Below we describe how the asymmetric inputs of Sec. 4.1 are built for the two paradigms in Sec. 5.

Bidirectional text-to-video. The whole latent window is denoised in one forward pass. We take two states (x_t, x_s) with $t > s$ from the DMD rollout and form $\tilde{x}^{\text{stu}}$ and $\tilde{x}^{\text{tea}}$ as in Eq. 6. The student G_θ processes $\tilde{x}^{\text{stu}}$ with per-token timesteps (s where $M{=}1$ and t where $M{=}0$), and the EMA teacher $G_{\theta'}$ processes $\tilde{x}^{\text{tea}}$ at timestep s. The hidden states at blocks $\mathcal{S}$ and $\mathcal{T}$ are passed to the projectors $\{\pi_\ell\}$ and aligned with Eq. 8. The extra cost over DMD does not depend on the number of denoising steps K; Table 1 reports its share of the training time.

Causal autoregressive generation. A causal student generates the video chunk by chunk and conditions on its previously generated chunks through a key-value (KV) cache. Applying the rep loss to the full window in a single pass would break this autoregressive structure and couple chunks that the student never processes jointly at inference. We therefore replay the chunk-wise rollout of Self-Forcing [16] in the representation branch:

- **Separate KV caches.** The student and the teacher each keep their own KV cache, filled chunk by chunk. The two caches share no state, so the student's gradient never reaches teacher activations.
- **Chunk-wise asymmetric inputs.** For chunk b, the student receives the mixed chunk $\tilde{x}^{\text{stu}}_b$ with per-token timesteps, and the teacher receives the cleaner chunk $x_s|_b$ at timestep s. RoPE positions match those of the chunk in the rollout.
- **Feature collection.** Hidden states at $\mathcal{S}$ and $\mathcal{T}$ are collected per chunk and concatenated along the token axis before computing Eq. 8, so the projectors and the loss are the same as in the bidirectional case.

C Hyperparameters and Implementation Details

Table 4 lists the hyperparameters of the three settings: Wan2.1-T2V-1.3B (bidirectional), HunyuanVideo-1.5 (bidirectional), and Wan2.1-T2V-1.3B (causal). The causal student is initialized from the Causal-Forcing [53] ODE-regression checkpoint.

Layer pair. Following Self-Flow [6], we specify the student and teacher blocks by relative depth rather than by absolute index. On Wan2.1 ($N_b{=}30$) we use blocks $(\{9\}, \{21\})$, i.e., depth ratios $(0.30, 0.70)$. On HunyuanVideo-1.5 ($N_b{=}54$) we use blocks $(\{18\}, \{42\})$, i.e., depth ratios $(0.33, 0.78)$.

Projector. Each projector π_ℓ is a 3-layer MLP, following REPA [51], with input and output width D (the backbone hidden size) and hidden width $4D$.

Implementation notes. Two details matter for stable training. (i) Teacher features are detached before the cosine loss, so no gradient reaches the teacher; without this, the EMA update becomes unstable. (ii) During warmup the projector learning rate is multiplied by 50 so that the randomly initialized projectors can fit the fixed targets quickly. At step T_{proj} we reset the multiplier to 1. Without this reset, the projector learning rate dominates the joint phase and training slowly diverges.

Compute. Each setting is trained for 750 steps on 32 GPUs with hybrid FSDP sharding and bf16 mixed precision (Table 4). One training cycle consists of one generator update and five critic updates, and the representation branch takes about 6% of its wall-clock time (Table 1).

D Evaluation Protocol Details

VBench [17]. We follow the official VBench T2V protocol: each model generates 5 videos for each of the 946 prompts, and we report the official Total, Quality, and Semantic scores. Dynamic Degree, Temporal Flickering, and Background Consistency, wherever reported, are computed with the official VBench metric heads without modification.

VisionReward [46]. The VisionReward score in Table 2 is the average over the 100 motion-rich prompts from Causal-Forcing [53], with one video per prompt at each model's default resolution. In the pairwise comparison of Fig. 3, the bidirectional SF^2 student is compared with DMD and with the multi-step Wan2.1, and the causal student with Causal-Forcing and Self-Forcing. For each

Table 4: **Full hyperparameters for the three SF^2 instantiations.** The rep-forcing branch is identical in spirit across settings; the only paradigm-dependent quantities are the masking ratio $\rho_{\max}$ and the chunk size F_b (causal only). All bidirectional models use $K{=}4$ denoising steps; the causal student also uses $K{=}4$ unless otherwise noted.

	Wan2.1-1.3B (Bid.)	HunyuanVideo-1.5 (Bid.)	Wan2.1-1.3B (Causal)
Backbones			
Student backbone	Wan2.1-T2V-1.3B	HunyuanVideo-1.5	Wan2.1-T2V-1.3B
Real-score critic (frozen)	Wan2.1-T2V-14B	HunyuanVideo-1.5	Wan2.1-T2V-14B
Fake-score critic (init.)	Wan2.1-T2V-1.3B	HunyuanVideo-1.5	Wan2.1-T2V-1.3B
Student init. ckpt	Wan2.1-T2V-1.3B	HunyuanVideo-1.5	Causal-Forcing ODE-regression
Backbone depth N_b	30	54	30
Diffusion / sampling			
Denoising steps K	4	4	4
Timestep schedule	flow-matching	flow-matching	flow-matching
Timestep shift	7	7	7
Guidance scale (training)	3.0	3.0	3.0
Resolution	848×480	848×480	848×480
Frames per video	81 (21 latent)	121 (31 latent)	81 (21 latent)
Chunk size F_b (latent)	–	–	3
Representation forcing			
Layer pair $(\mathcal{S}, \mathcal{T})$	$(\{9\}, \{21\})$	$(\{18\}, \{42\})$	$(\{9\}, \{21\})$
Depth ratio $(\ell_\theta, \ell_{\theta'})/N_b$	$(0.30, 0.70)$	$(0.33, 0.78)$	$(0.30, 0.70)$
Projector π_ℓ	3 layers MLP, hidden $4D$	3 layers MLP, hidden $4D$	3 layers MLP, hidden $4D$
Loss type	cosine $(1-\cos)$	cosine $(1-\cos)$	cosine $(1-\cos)$
Rep weight λ_{rep}	0.5	0.5	0.5
Mask ratio $\rho_{\max}$	0.4	0.4	0.2
Mask sampling pattern	space-only	space-only	space-only
EMA teacher and warmup			
EMA momentum μ	0.99	0.99	0.99
Projector-only warmup T_{proj}	100	100	100
EMA direct-copy steps T_{ema}	100	100	100
Projector lr multiplier (during warmup)	50×	50×	50×
Optimization			
Optimizer	AdamW	AdamW	AdamW
(β_1, β_2)	$(0.0, 0.999)$	$(0.0, 0.999)$	$(0.0, 0.999)$
Weight decay	0.01	0.01	0.01
Generator lr	$2{\times}10^{-6}$	$2{\times}10^{-6}$	$2{\times}10^{-6}$
Critic lr	$4{\times}10^{-7}$	$4{\times}10^{-7}$	$4{\times}10^{-7}$
Critic / generator updates	5:1	5:1	5:1
Mixed precision	bf16	bf16	bf16
Grad checkpointing	yes	yes	yes
Max grad norm	10	10	10
Per-device batch size	1	1	1
Distributed training			
Sharding strategy	FSDP (hybrid)	FSDP (hybrid)	FSDP (hybrid)
Wrap policy	size	size	size
GPUs(global batch size)	32	32	32
Training steps	750	750	750

prompt, both models generate one video and the one with the higher VisionReward score is counted as preferred. All four pairings use the same prompts, seed, and sampling settings.

Validation set for ablations. The ablation curves in Sec. 5.2 are computed on 400 held-out prompts generated by GPT-5 with instructions to produce complex motion: articulated subjects, fast camera or subject motion, multi-object interaction, and irreversible events. We use this set instead of the full VBench suite because it targets motion, the main failure mode of few-step students, and because it is cheap enough to evaluate every 100 training steps.

3D re-rendering probe. For Table 3, we use 120 prompts describing mostly static scenes under smooth camera motion and generate one video per prompt with each HunyuanVideo-1.5 student. WorldMirror-2 [18] reconstructs each video into a 3D Gaussian Splatting [22] scene, which we re-render along the camera trajectory of the video and compute PSNR, SSIM, and LPIPS against the generated frames. Temporal Flickering and Background Consistency are computed with VBench on the generated frames. Both methods have the same Dynamic Degree (90.00), so the lower re-rendering error of SF^2 is not explained by generating less motion.

Throughput and latency. Throughput and latency in Table 2 are single-H100-equivalent values with bf16 inference. All distilled models use $K=4$ sampling steps, and multi-step baselines use the number of inference steps reported in their papers.

E Datasets

Training prompts. All students are trained on text prompts from VidProM [43]. Since DMD is data-free, no real videos are used in training.

Evaluation prompt sets. We use four prompt sets, none of which is used in training:

- **VBench** [17]: the full 946-prompt T2V suite, used for Table 2.
- 100 **motion-rich prompts** from Causal-Forcing [53]: used for the VisionReward scores in Table 2 and Fig. 3.
- 400 **motion-rich prompts** generated by GPT-5: used for the ablation curves in Fig. 4.
- 120 **static-scene prompts** generated by GPT-5: used for the 3D re-rendering probe in Table 3.

Licenses of existing assets. Wan2.1 [41] (code and weights), VBench [17], Self-Forcing [16], Causal-Forcing [53], and the VisionReward [46] code are released under Apache-2.0. HunyuanVideo-1.5 [45], WorldMirror-2 [18], and WorldPlay [37] are released under their respective Tencent community licenses. The VisionReward-Video weights are released under the CogVLM License, the VidProM [43] prompts under CC BY-NC 4.0, and the CausVid [50] code under CC BY-NC-SA 4.0. We use all assets in accordance with their terms.

F Limitations

Bounded by the teacher. SF^2 changes how the student is distilled, not what the teacher knows. Prompt adherence on long compositional prompts and on rare concepts is bounded by the multi-step teacher, and biases of the teacher and its training data, such as cultural or demographic skew, carry over to the student. The rep loss improves the student's internal features but cannot add concepts that the teacher does not represent.

Training overhead. The EMA teacher is a full copy of the student and adds one set of generator weights to the training memory. The representation branch takes about 6% of the training time (Table 1).

Per-paradigm masking ratio. The best masking ratio differs between paradigms ($\rho_{\max}=0.4$ for bidirectional and 0.2 for causal students, Fig. 4), so a new setting may require a small sweep over $\rho_{\max}$.

Evaluation. Our results rely on automatic metrics (VBench, VisionReward, and 3D re-rendering). We do not conduct a human study or report variance across training seeds.

Backbone coverage. We evaluate SF^2 on two transformer backbones (Wan2.1 and HunyuanVideo-1.5) in both bidirectional and causal settings. Other architectures, such as state-space or hybrid backbones, are not tested and may need a different layer pair.

G Broader Impacts

Positive impacts. SF^2 yields 4-step video generators with more than $20\times$ the throughput of the multi-step Wan2.1 (Table 2). Faster sampling reduces the energy and compute cost per generated video and enables real-time and interactive applications such as the world model in Sec. 5.4.

Negative impacts. The same speed-up lowers the cost of producing realistic synthetic video, which can be misused for disinformation, impersonation, or non-consensual content, and real-time generation makes such content easier to produce at scale. SF^2 does not give the student capabilities beyond those of its teacher (Appendix F), and the Wan2.1 and HunyuanVideo-1.5 teachers are publicly released; the main added risk is therefore lower cost and latency rather than new content. Biases of the teachers and of the training prompts also carry over to the distilled models.

Safeguards. We will release the code and the distilled checkpoints. Each checkpoint is released under terms consistent with the licenses of its base model and training data (Appendix E) and comes with a model card that states the intended research use, the limitations in Appendix F, and prohibited uses such as generating deceptive or non-consensual content. Training uses only text prompts from VidProM, and no human-subject data is collected. We recommend that downstream deployments add content moderation and label generated videos as synthetic.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discuss the limitations of the work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper fully disclose all the information needed to reproduce the main experimental results of the paper.

   Guidelines:

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

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

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

Answer: [No]

Justification: We will release all the code upon acceptance of the paper.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper specify all the training and test details.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The results are not accompanied by error bars, confidence intervals, or statistical significance tests. While formal statistical tests are computationally prohibitive for large video models, we ensure robustness by evaluating over large-scale, diverse prompt sets.

   Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: For each setting, we report the number of training steps, the per-device and global batch sizes, the number of GPUs, and the parallelism and precision settings, as well as the per-cycle cost breakdown. The GPU model and the training time are not disclosed for confidentiality reasons.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses both potential positive societal impacts and negative societal impacts of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

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

11. **Safeguards**

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

    Answer: [Yes]

    Justification: The paper describes safeguards.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The paper cites the original paper that produced the code package or dataset.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We will release distilled checkpoints upon acceptance.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: The paper describes the usage of LLMs. GPT-5 is used solely to generate text prompts for evaluation set construction. No LLM is used in core methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.