DynaForcing: Overcoming Dynamic Collapse in Self-Forcing Distillation for Streaming Avatar Generation

Anonymous Author(s)

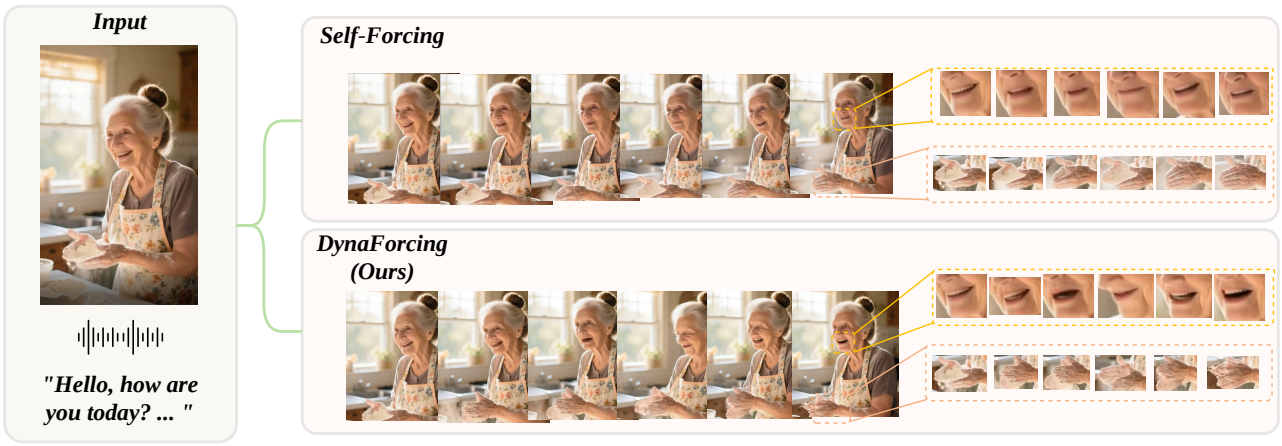

Figure 1: Dynamic collapse in self-forcing distillation for streaming avatar generation. Given the same reference image and driving audio (left), the self-forcing baseline (top) produces visually appealing but temporally frozen outputs—nearly identical lip shapes across 10 seconds of speech (see mouth crops). DynaForcing (bottom) restores faithful audio-driven dynamics with natural expression variation and accurate lip articulation, at 45.2 FPS real-time streaming.

Abstract

Audio-driven avatar generation requires realistic lip-sync, expressive motion, and real-time streaming. Recent work achieves the latter via self-forcing with Distribution Matching Distillation (DMD), but we uncover a critical failure: **dynamic collapse**, where the student model converges to a near-static optimum with high perceptual quality but severely suppressed temporal dynamics. We trace this to two causes: the reverse KL objective in DMD, which biases toward low-motion modes, and unanchored self-conditioning, which creates a feedback loop that amplifies collapse. This is especially harmful for avatars, where even subtle motion loss breaks lip-sync and expression. To address this, we propose **DynaForcing**, a training framework with three complementary strategies applied at different levels. Specifically, *Hybrid Forcing* anchors rollouts to ground-truth dynamics at the data level to break the feedback loop. *Dynamics-Aware Reward Regularization* introduces explicit motion rewards via the RL interpretation of DMD to counteract the reverse KL bias at the loss level. *Reference Perturbation* perturbs reference images to decouple identity from static details, forcing the model to rely on audio for motion at the conditioning level. We further introduce computation graph pruning and gradient replay, reducing the GPU footprint of self-forcing by over an order of magnitude. Experiments show that DynaForcing recovers dynamics to teacher-comparable levels (Dyn-Deg: 0.31→0.73, Sync-C: 7.03→7.68) while improving visual quality, resolving the quality–dynamics trade-off throughout training without early stopping.

CCS Concepts

• **Computing methodologies** → **Computer vision**; **Video generation**.

Keywords

interactive video generation, efficient video generation

1 Introduction

Audio-driven avatar generation aims to synthesize photorealistic talking-head video whose motion faithfully follows an input audio stream, serving as a cornerstone technology for interactive digital communication [7, 23, 29]. Deploying such systems in real-world scenarios, such as video conferencing and virtual assistants, requires satisfying two simultaneous demands: *real-time streaming* at interactive frame rates, and *temporally rich dynamics* including precise lip articulation, natural expressions, and responsive head motion. Recent large-scale video diffusion models [1, 31] have significantly advanced visual fidelity, and self-forcing [4, 9] has emerged as a leading distillation paradigm, leveraging Distribution Matching Distillation (DMD) [38] to convert these models into causal, few-step streaming generators that enable real-time, infinite-length avatar generation at 14B-parameter scale [10, 25].

Yet beneath these impressive capabilities lies a failure mode that has received limited formal study. We observe that self-forcing training systematically drives the student model toward what we term **dynamic collapse** (Figure 1), where generated videos exhibit high perceptual quality scores but near-zero temporal dynamics, with mouths barely move, expressions freezing, and the rich motion

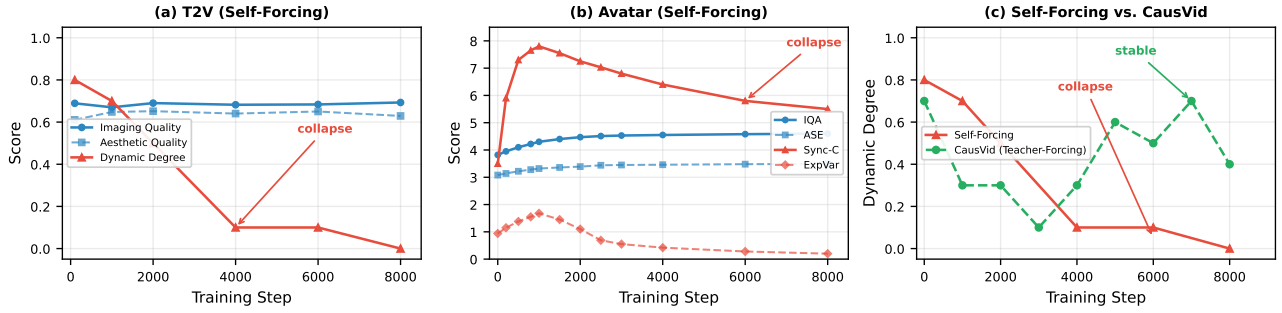

Figure 2: Dynamic collapse in self-forcing distillation. (a) Text-to-video (Self-Forcing [9]): VBench [11] *dynamic degree* drops from 0.80 to 0.00 while visual quality improves. (b) Audio-driven avatar (LiveAvatar [10]): Sync-C and ExpVar degrade while IQA and ASE improve. (c) Comparing self-forcing with CausVid [39] (GT-anchored): CausVid's dynamic degree fluctuates around 0.43 without collapsing, confirming that unanchored self-conditioning, not DMD distillation itself, drives the collapse.

present in the teacher model vanishes. Avatar generation is particularly susceptible because the task requires *fine-grained* temporal dynamics (precise lip shapes for each phoneme, micro-expressions, head motion responsive to prosody), yet the static-scene prior (fixed background, stable identity) means that a near-motionless output already satisfies most visual quality criteria, leaving little gradient pressure toward dynamics. While DMD's diversity–quality tradeoff is well-documented for images [37, 38], its impact on video is qualitatively different: collapsing temporal diversity does not merely reduce "variety" but fundamentally breaks the core requirement of faithful audio-visual correspondence.

We trace this failure to two contributing mechanisms. First, DMD's reverse KL objective $D_{\mathrm{KL}}(p_\theta \| p_{\mathrm{data}})$ is mode-seeking [20]: it penalizes the student for placing mass outside the teacher's support but imposes no cost for ignoring modes. Second, self-forcing performs *unanchored self-conditioning*: unlike GT-anchored approaches (e.g., CausVid [39]) that process all blocks in parallel with ground-truth latents as input, self-forcing builds the entire KV cache from the student's own outputs with no ground-truth anchoring whatsoever. This creates a compounding feedback loop: when the student begins producing low-dynamics content, all subsequent blocks condition on this self-generated static history, reinforcing the collapse across the entire sequence. We verify this empirically: under identical training horizons, CausVid's GT-anchored conditioning maintains stable dynamics while self-forcing collapses to near-zero, as demonstrated in Figure 2c.

To address dynamic collapse, we propose **DynaForcing**, a training framework with three complementary strategies operating at different levels of the training pipeline:

(1) **Hybrid Forcing** (§4.2), a *data-level* strategy that probabilistically replaces self-forcing rollout starting points with noised ground-truth latents. This provides a ground-truth anchor that counteracts mode collapse in the distillation objective.

(2) **Dynamics-Aware Reward Regularization** (§4.3), a *loss-level* strategy that leverages the RL interpretation of DMD [17] to introduce explicit dynamics rewards (lip-sync accuracy and expression variance) as auxiliary training signals that directly penalize static outputs.

(3) **Reference Perturbation** (§4.4), a complementary *conditioning-level* strategy that perturbs reference images to decouple identity from extraneous visual details (pose, background, lighting), preventing the model from taking a shortcut of copying the reference frame instead of generating audio-driven motion.

Additionally, we present **efficient self-forcing training** (§4.5) via computation graph pruning and gradient replay, reducing GPU requirements by over 10× while preserving quality.

In summary, the main contributions of this work are as follows:

- We identify and empirically characterize *dynamic collapse* in self-forcing distillation, tracing it to reverse-KL mode-seeking and the compounding effect of unanchored self-conditioning.
- We propose DynaForcing, combining three strategies at different levels (data, loss, conditioning) that jointly address dynamic collapse while preserving visual quality.
- We introduce computation graph pruning and gradient replay for self-forcing, achieving ∼10× GPU-hour reduction.
- Experiments on audio-driven avatar generation demonstrate substantial improvements in motion naturalness and audio-visual faithfulness on both short- and long-form benchmarks.

2 Related Work

Audio-Driven Avatar Generation. Audio-driven talking-head synthesis has evolved from GAN-based methods [23, 40] to diffusion-based approaches [3, 7, 8, 29] that leverage large-scale video diffusion models for dramatically improved fidelity. More recently, several works have achieved *real-time streaming* avatar generation by combining diffusion models with autoregressive distillation [10, 12, 25, 27, 32], demonstrating interactive-rate generation at scale. However, these streaming methods consistently exhibit *significantly degraded motion precision and naturalness* compared to their multi-step teachers, even though the underlying DMD distillation has been shown to produce students that match or exceed teacher quality on static image metrics [37, 38]. For avatars, this motion loss is particularly damaging: lip-sync accuracy, expression diversity, and gesture naturalness are the core requirements, and any dynamics degradation directly undermines the task objective.

Streaming Video Distillation. Distilling bidirectional multi-step diffusion models into few-step causal streaming generators has been

pursued through progressive distillation [24], consistency models [26], and DMD [37, 38]. CausVid [39] introduced GT-anchored autoregressive distillation, and Self-Forcing [4, 9] further removed ground-truth anchoring by conditioning entirely on the student's own outputs, better aligning train-time and test-time distributions. Subsequent works have refined this paradigm: Causal Forcing [41] bridges the architectural gap via AR-teacher ODE initialization, Rolling Forcing [15] suppresses error accumulation through joint multi-frame denoising, and LongLive [36] extends stable generation to longer horizons. The field has largely converged on DMD-based self-forcing as the dominant paradigm for streaming video generation. Yet self-forcing introduces an inherent *motion degradation* problem: the combination of reverse-KL mode-seeking and unanchored self-conditioning creates a feedback loop that systematically suppresses temporal dynamics. This has been independently observed by DiagDistill [14], which proposes diagonal forcing with implicit flow modeling to mitigate motion amplitude attenuation, and Reward Forcing [16], which introduces reward signals to encourage dynamics. Nevertheless, the problem remains largely unsolved, especially for tasks like avatar generation that demand fine-grained, audio-synchronized motion rather than coarse temporal coherence.

3 Preliminaries

3.1 Self-Forcing for Streaming Video Diffusion

Self-forcing [9] enables streaming video generation by combining autoregressive rollout with few-step diffusion sampling. The video is generated block-by-block, where each block B^i contains L consecutive frame latents. At each rollout step, the student model G_θ denoises the current block conditioned on a rolling KV cache of previous blocks:

$$\hat{B}_0^i = G_\theta(B_T^i, \underbrace{B_t^{(i-w):(i-1)}}_{\text{KV cache}}, I, a^i), \tag{1}$$

where $B_T^i \sim \mathcal{N}(0, \mathbf{I})$ is the initial noise, w is the cache window size, I is the reference (sink) frame, a^i is the audio embedding, and t denotes the shared noise level of the KV cache (timestep-forcing). The student produces a multi-step denoising trajectory $B_T^i \to B_{t_1}^i \to \cdots \to \hat{B}_0^i$, and the generated clean latent $\hat{B}_0^i$ is used to construct the KV cache for subsequent blocks.

3.2 Distribution Matching Distillation as RL

Distribution Matching Distillation (DMD) [38] trains the student by minimizing the reverse KL divergence between the student's induced distribution p_θ and the teacher's data distribution p_{data}, averaged over noise levels t:

$$\mathcal{L}_{\text{DMD}} = \mathbb{E}_t \left[D_{\text{KL}}\left(p_{\theta,t} \,\|\, p_{\text{data},t}\right) \right]. \tag{2}$$

The gradient of this objective takes the form:

$$\nabla_\theta \mathcal{L}_{\text{DMD}} = -\mathbb{E}_{t,\mathbf{z}} \left[\left(s_{\text{real}}(\mathbf{x}_t, t) - s_{\text{fake}}(\mathbf{x}_t, t)\right)^\top \frac{\partial G_\theta(\mathbf{z})}{\partial \theta} \right], \tag{3}$$

where s_{real} is the teacher (pretrained) score and s_{fake} is a learned score on student-generated data, and $\mathbf{x}_t = \Psi(\hat{\mathbf{x}}, t)$ is the noise-perturbed student output.

Crucially, Luo *et al.* [17] frame DMD within an RL/RLHF objective: the score difference in Eq. 3 serves as an Integral KL regularization toward the teacher, and additional reward signals can be incorporated by augmenting the objective with an explicit reward term. Building on this, Lu *et al.* [16] propose *reward-weighted* distillation, where an external reward $R(\mathbf{x})$ exponentially reweights the DMD gradient:

$$\nabla_\theta \mathcal{L}_{\text{Re-DMD}} = -\mathbb{E}_{t,\mathbf{z}} \left[\exp\left(\alpha \cdot \overline{R}\right) \cdot \left(s_{\text{real}} - s_{\text{fake}}\right)^\top \frac{\partial G_\theta(\mathbf{z})}{\partial \theta} \right], \tag{4}$$

where $\overline{R} = \text{sg}(R)$ is the stopped-gradient reward and α controls the reward strength. This formulation amplifies the DMD gradient for high-reward samples while suppressing it for low-reward ones, effectively steering the student toward reward-rich modes without requiring backpropagation through the reward model.

This reward-weighted framework is central to our approach: by defining a dynamics-specific reward, we can bias the self-forcing training loop away from degenerate static modes (§4.3).

4 Method: DynaForcing

We first analyze the dynamic collapse phenomenon (§4.1), then present our three strategies: Hybrid Forcing (§4.2), Dynamics-Aware Reward Regularization (§4.3), and Reference Perturbation (§4.4). Finally, we describe efficient training (§4.5).

4.1 Understanding Dynamic Collapse

We observe that self-forcing training can severely suppress temporal dynamics despite producing visually appealing outputs. We term this phenomenon **dynamic collapse** and analyze its causes.

Empirical evidence. Figure 2 demonstrates that dynamic collapse is not task-specific but a general pathology of self-forcing distillation.[1] On text-to-video generation (Figure 2a), VBench [11] *dynamic degree* monotonically decreases from 0.80 to 0.00 over 8K training steps, while *imaging quality* and *aesthetic quality* steadily improve. On audio-driven avatar generation (Figure 2b), the same pattern emerges: visual quality (IQA, ASE) improves steadily while dynamics metrics (Sync-C, ExpVar) peak early and then decline sharply, eventually reaching severely suppressed levels. In both cases, the generated videos look sharp and artifact-free but exhibit minimal temporal variation.

Analysis: Mode-seeking reverse KL. DMD minimizes $D_{\text{KL}}(p_\theta \| p_{\text{data}})$, which is mode-seeking [20]: it penalizes mass outside the teacher's support but incurs no cost for ignoring modes. In the video domain, static content is an attractive mode because the DMD score difference $s_{\text{real}} - s_{\text{fake}}$ provides limited gradient signal for dynamics, as the teacher's score function primarily captures distributional realism, not temporal richness. As the student concentrates on a narrow, low-motion subset of the teacher's distribution, the fake score s_{fake} quickly matches s_{real} in this region, eliminating the gradient signal needed to explore other modes.

Analysis: Unanchored self-conditioning. Self-forcing builds the entire KV cache from the student's own outputs, with no ground-truth anchoring during the rollout. Once the student begins producing low-motion blocks, all subsequent blocks condition on this static history, creating a compounding feedback loop that reinforces

[1] Reproduction details for Figure 2 are provided in the supplementary material.

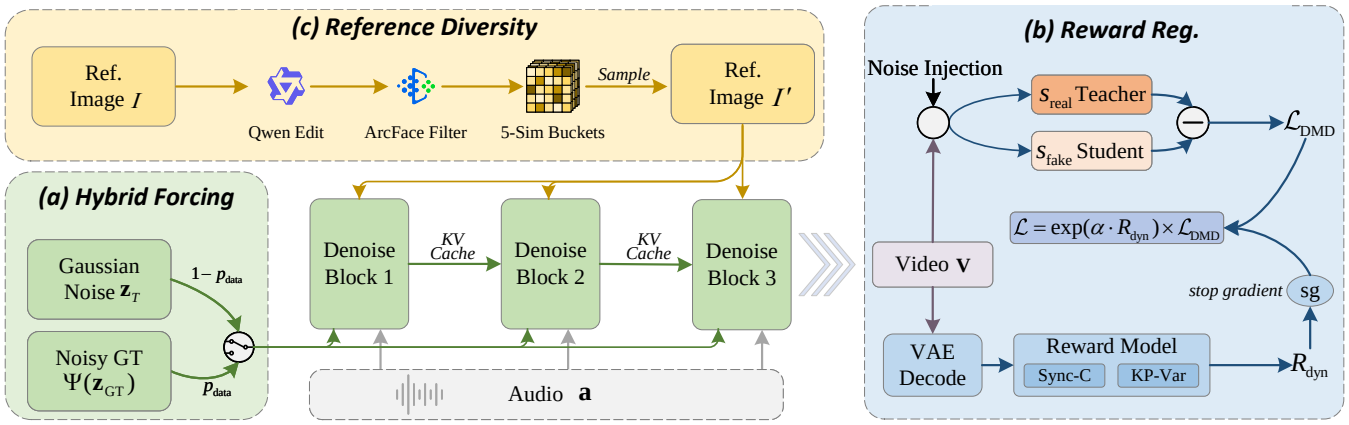

Figure 3: Overview of DynaForcing. Three strategies intervene at different levels of the self-forcing training pipeline: (a) Hybrid Forcing at the input level, (b) dynamics-aware reward regularization at the loss level, and (c) reference perturbation at the conditioning level.

collapse across the sequence. The key insight is not autoregression per se, but the *absence of any ground-truth signal in the conditioning path*. Figure 2c validates this analysis: under identical evaluation, CausVid [39], which conditions each block on noisy ground-truth frames, maintains stable dynamics (mean 0.43) while self-forcing collapses to 0.

Analysis: Why avatar generation is particularly affected. While dynamic collapse can occur in general video distillation, audio-driven avatar generation is particularly susceptible because: (i) the task requires *fine-grained* temporal dynamics (precise lip shapes for each phoneme, micro-expressions), making any dynamics reduction immediately noticeable; and (ii) the static-scene prior (fixed background, stable identity) means that a zero-motion output already satisfies most of the visual quality requirements, leaving little gradient pressure toward dynamics.

4.2 Hybrid Forcing: Data-Anchored Rollout

To break the feedback loop of dynamic collapse, we propose *Hybrid Forcing*, which probabilistically injects ground-truth dynamics into the self-forcing rollout at the *starting point* level, providing an anchor that counteracts mode collapse under unanchored self-conditioning.

At each training step, we sample a mode for the entire rollout (i.e., the decision is per-sample, applied uniformly to all blocks). Denoting the full sequence starting point as $\mathbf{z} = [B^1_{\text{start}}, \ldots, B^K_{\text{start}}]$, each block's starting point is:

$$B^i_{\text{start}} = \begin{cases} B^i_T \sim \mathcal{N}(0, \mathbf{I}), & \text{w.p. } 1 - p_{\text{data}}, \\ \Psi(B^i_{\text{GT}}, t_{\text{start}}), & \text{w.p. } p_{\text{data}}, \end{cases} \tag{5}$$

where B^i_{GT} is the ground-truth video latent, $\Psi(\cdot, t)$ is the forward noise scheduler, and $t_{\text{start}} \sim \{t_0, \ldots, t_{K-1}\}$ is sampled from the K noise levels in the few-step denoising schedule. In data-anchored mode, the student starts from a noised ground-truth rather than pure noise, preserving temporal dynamics (motion patterns, lip movements) as structural information that the student must refine.

The DMD loss is then computed on the student's denoised output regardless of the starting mode. When starting from data-anchored points, the loss gradient points toward dynamically-rich outputs, providing an escape direction from the zero-motion basin. The KV cache always uses the student's own representations (noisy, per timestep-forcing), maintaining train-test alignment.

Why data-anchored starting points help. A noised ground-truth latent $\Psi(x_{\text{GT}}, t) = \alpha_t x_{\text{GT}} + \sigma_t \varepsilon$ retains recoverable structure (motion cues, temporal dynamics, spatial layout) as long as the signal-to-noise ratio α_t/σ_t is not too small. A pure Gaussian start $z \sim \mathcal{N}(0, \mathbf{I})$ carries none of this information, forcing the student to reconstruct everything from its imperfect learned prior. We formalize this intuition in the supplementary material, showing that data-anchored starts yield a tighter Wasserstein bound on the denoised output.

Starting-step probability redistribution. Starting from t_{start} means the model only traverses the denoising sub-trajectory $t_{\text{start}} \to \cdots \to t_0$ (Eq. 5), so noisier levels receive less training coverage. Backward simulation [37], which self-forcing inherits, addresses a similar coverage issue for pure-noise starts. To compensate in the data-anchored case, we redistribute the sampling probability of t_{start}: since t_0 already operates at high SNR where strong dynamic priors are preserved and little denoising correction is needed, we concentrate mass on early steps, which retain the most dynamic structure from the ground truth while still requiring non-trivial denoising.

4.3 Dynamics-Aware Reward Regularization

Framework. As discussed in §3.2, DMD can be viewed through an RL lens where the score difference serves as an implicit reward [17]. However, this implicit reward can be fully satisfied by mode collapse (concentrating on high-likelihood static content), since it measures distributional realism, not temporal richness. We introduce an explicit dynamics reward R_{dyn} as an additional return signal that directly penalizes static outputs. Following the reward-weighted distillation framework [16, 17], we integrate R_{dyn} by exponentially

reweighting the DMD gradient:

$$\nabla_\theta \mathcal{L}_{\text{DynaForcing}} = -\mathbb{E}_{t,\mathbf{z}}\left[\exp\big(\alpha \cdot \text{sg}(R_{\text{dyn}})\big) \cdot \big(s_{\text{real}} - s_{\text{fake}}\big)^\top \frac{\partial G_\theta(\mathbf{z})}{\partial \theta}\right], \quad (6)$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator and α controls the reward strength. This amplifies the DMD gradient for samples with rich dynamics and suppresses it for static outputs, approximating optimization toward a reward-tilted target $\tilde{p}(\mathbf{x}) \propto p_{\text{teacher}}(\mathbf{x}) \exp(\alpha R_{\text{dyn}}(\mathbf{x}))$, analogous to reward-weighted regression in offline RL [22].

Reward implementation. Let $\hat{\mathbf{V}} = [\hat{B}_0^1, \ldots, \hat{B}_0^K]$ denote the student-rolled video. We define the dynamics reward as:

$$R_{\text{dyn}}(\hat{\mathbf{V}}, \mathbf{a}) = \lambda_{\text{sync}} \cdot \widetilde{\text{Sync-C}}(\hat{\mathbf{V}}, \mathbf{a}) + \lambda_{\text{exp}} \cdot \widetilde{\text{ExpVar}}(\hat{\mathbf{V}}), \quad (7)$$

where $\widetilde{\text{Sync-C}}$ is the normalized lip-sync confidence from a pretrained SyncNet [2] and $\widetilde{\text{ExpVar}}$ is the normalized standard deviation of 3DMM expression coefficients across frames [42]. Both are computed on decoded video frames and normalized to $[0, 1]$ via running statistics. R_{dyn} is treated as a constant during backpropagation (stop-gradient), so only the score terms and generator Jacobian carry gradient flow, making this fully compatible with standard DMD training.

4.4 Reference Perturbation

We identify that standard training practice induces a complementary source of dynamics suppression at the conditioning level: the reference image leaks far more information than identity alone.

The copy-dominance problem. In avatar generation, the reference image I is typically sampled from the same video clip as the target. Since reference and target share scene, background, viewpoint, and often similar pose, the model learns a shortcut: *copy the reference frame* with minimal modification, rather than generating genuine audio-driven dynamics. The reference is meant to supply only identity, but in practice it leaks far more, suppressing the need for temporal generation.

Perturbing references to decouple identity. To decouple identity from extraneous visual details, we perturb reference images so that they retain identity but vary in other attributes. Specifically, we use an off-the-shelf image editing model [34] to generate perturbed reference images from the training set. Given a reference frame I, we apply semantic-preserving edits (viewpoint variation, background modification, lighting changes) that break pixel-level correspondence between reference and target while preserving identity. By training with these perturbed references, the model is forced to extract only identity information from the reference and rely on the audio signal for temporal dynamics, rather than copying static content. Detailed prompts are provided in the supplementary material.

Similarity-calibrated sampling. Not all perturbed references are equally useful. We filter and organize them using a two-stage criterion: (1) *Identity filtering*: we compute ArcFace [5] cosine similarity between the perturbed reference and the original, retaining only pairs with similarity $> \tau_{\text{arc}}$ to ensure identity consistency. (2) *Perturbation-aware bucketing*: for retained pairs, we compute the structural similarity (SSIM [33]) between the perturbed and original images and partition pairs into 5 buckets of width 0.15 spanning $[0.25, 1.0]$. During training, we uniformly sample across buckets, ensuring the model sees a balanced distribution from near-identical to substantially different reference-target pairs. This balanced curriculum prevents the model from relying on the copy shortcut, forcing it to generate genuine audio-driven dynamics.

4.5 Efficient Self-Forcing Training

Self-forcing training is memory-intensive because the full rollout graph must be retained for backpropagation [9]. We introduce two optimizations that substantially reduce memory requirements.

Computation graph pruning. We empirically observe that cross-block gradients (propagated through the KV cache) have negligible impact on training outcomes, yet significantly increase memory pressure. We therefore detach the KV cache from the computation graph, removing cross-block gradient dependencies.

Block-wise gradient replay. With cross-block gradients eliminated, we adopt a two-phase training procedure. In the *rollout phase*, the entire sequence is generated with gradients disabled, and intermediate results (latent outputs, KV cache entries) are stored in memory. In the *replay phase*, we iterate over blocks one at a time: for each block, we re-enable gradients and run a single forward pass using the stored intermediates, then compute and accumulate the DMD gradient. This reduces peak activation memory by $\sim K\times$ (where K is the number of blocks), at the cost of K additional forward passes.

5 Experiments

5.1 Experimental Setup

Implementation details. Our approach is built upon the WanS2V [8] (14B). The training process is divided into two stages. For Stage 1, we strictly follow the diffusion-forcing pretraining protocol introduced by LiveAvatar [10], while our proposed modifications are exclusively applied during Stage 2. During the inference phase, we employ the TPP strategy from [10] to generate videos at a resolution of 720×400. Regarding the optimization hyperparameters, we set the probability p_{data} to 0.3. We intentionally concentrate the starting-step probability mass on the early noise levels by assigning $p(t_1) = 0.70$, alongside $p(t_0) = p(t_2) = 0.15$. Furthermore, the reward weights are configured as follows: the synchronization weight λ_{sync} is 1.0, the expression weight λ_{exp} is 0.5, and the balancing factor α is 0.1. We also establish an ArcFace threshold of $\tau_{\text{arc}} = 0.9$. Finally, our model is trained for 4,000 steps on eight NVIDIA H100 GPUs. For the DMD optimization, we use a learning rate of 1×10^{-5} for the student and 2×10^{-6} for the fake score, with an update frequency ratio of 5.

Datasets. We train our model on the AVSpeech dataset [6], utilizing the preprocessing pipeline introduced by OmniAvatar [7]. This resulting training corpus consists of 400,000 samples, with all video clips having a duration of more than 10 seconds. For evaluation, we adopt two benchmark datasets from [10]. The first is GenBench-ShortVideo, which includes 100 samples that are approximately 10 seconds long, and the second is GenBench-LongVideo, which comprises 15 samples with durations exceeding 5 minutes.

Evaluation metrics. We adopt standard metrics: Q-Align [35] for perceptual quality (IQA) and aesthetic appeal (ASE), Sync-C and Sync-D [2] for audio-visual synchronization, DINOv2 [21] cosine similarity (Dino-S) for identity consistency. We additionally

Table 1: Quantitative comparison on GenBench-ShortVideo (~10 s) and GenBench-LongVideo (>5 min) [10]. Bold: best. Underline: second best. ↑/↓: higher/lower is better. FPS is resolution- and hardware-dependent (720×400, single H800 node) and identical across benchmarks.

Method	Standard Metrics						Dynamics	
	ASE↑	IQA↑	Sync-C↑	Sync-D↓	Dino-S↑	FPS↑	ExpVar↑	Dyn-Deg↑
GenBench-ShortVideo								
Non-realtime								
WanS2V [8]	3.36	4.29	5.89	9.08	0.95	0.25	1.81	0.72
OmniAvatar [7]	3.53	4.49	6.77	8.22	0.95	0.16	1.95	**0.75**
Hallo3 [3]	3.12	3.97	4.74	10.19	0.94	0.26	1.48	0.57
StableAvatar [30]	3.52	4.47	3.42	11.33	0.93	0.64	1.23	0.51
EchoMimic-V2 [19]	2.82	3.61	5.57	9.13	0.79	0.53	1.56	0.60
Realtime								
Ditto [13]	3.31	4.24	4.09	10.76	**0.99**	21.80	0.94	0.37
LiveAvatar [10]	3.44	4.51	7.03	8.30	0.96	**45.2**	0.69	0.31
SoulX-FlashTalk [25]	3.48	4.53	7.12	8.25	0.96	32.5	0.81	0.35
DynaForcing (Ours)	**3.55**	**4.58**	**7.68**	**7.89**	0.96	**45.2**	**2.02**	0.73
GenBench-LongVideo								
Non-realtime								
WanS2V	2.63	3.99	6.04	9.12	0.80	0.25	1.63	0.65
OmniAvatar	2.36	2.86	8.00	**7.59**	0.66	0.16	**2.06**	**0.71**
Hallo3	2.65	4.04	6.18	9.29	0.83	0.26	1.36	0.53
StableAvatar	3.00	4.66	1.97	13.57	0.94	0.64	1.07	0.44
Realtime								
Ditto	2.90	4.48	3.98	10.57	**0.97**	21.80	0.76	0.35
LiveAvatar	3.42	4.76	7.16	8.31	**0.97**	**45.2**	0.57	0.28
SoulX-FlashTalk	3.40	4.72	7.08	8.18	0.96	32.5	0.67	0.32
DynaForcing (Ours)	**3.50**	**4.82**	**8.05**	8.02	**0.97**	**45.2**	1.93	0.68

report **dynamics metrics**: **ExpVar** (standard deviation of 3DMM expression coefficients across frames [42], measuring expression dynamics) and VBench [11] **Dyn-Deg** (dynamic degree, measuring temporal variation diversity).

Baselines. We compare with methods spanning two paradigms: (1) *Non-realtime* multi-step diffusion models: WanS2V [8] (teacher), OmniAvatar [7], Hallo3 [3], StableAvatar [30], EchoMimic-V2 [19]; (2) *Realtime* streaming models: Ditto [13], LiveAvatar [10], SoulX-FlashTalk [25]. All benchmarked at 720×400 on a single H800 node. For methods with strict requirements on reference image aspect ratio or subject positioning (SoulX-FlashTalk and EchoMimic-V2), we use their official open-source inference pipelines without modification to ensure fair comparison. The "self-forcing baseline" in ablations denotes standard self-forcing trained for 4,000 steps, identical to DynaForcing but without our three proposed components.

5.2 Main Results

GenBench-ShortVideo. Table 1 (top) presents results. Realtime self-forcing models (LiveAvatar, SoulX-FlashTalk) achieve strong visual quality but suffer severely on dynamics: LiveAvatar's ExpVar (0.69) is 62% lower than its teacher WanS2V (1.81), and its Dyn-Deg (0.31) is less than half of OmniAvatar (0.75), confirming dynamic collapse as a systematic issue. DynaForcing recovers dynamics to teacher-comparable levels: Dyn-Deg rises from 0.31 to 0.73 (teacher: 0.72), ExpVar from 0.69 to 2.02, and Sync-C from 7.03 to 7.68. That ExpVar slightly exceeds the teacher (2.02 vs. 1.81) is consistent with the known tendency of DMD-distilled models to concentrate output distributions [18, 28]; our reward regularization further steers this concentration toward dynamically-rich modes. Note that Sync-C and ExpVar are both components of our training reward, so gains on these two metrics alone cannot rule out reward hacking. We therefore highlight Dyn-Deg, which is *not* optimized during training yet recovers from 0.31 to 0.73 (matching the teacher's 0.72), as independent evidence of genuine dynamics recovery. The user study (§5.4) further corroborates this with strong human preference on motion naturalness. Moreover, by stabilizing dynamics throughout training, DynaForcing avoids the early quality–dynamics trade-off that forces vanilla self-forcing to stop prematurely, enabling full convergence and thus improved visual quality as a byproduct (ASE/IQA: 3.55/4.58 vs. 3.44/4.51). FPS remains 45.2 as our contributions are training-side only.

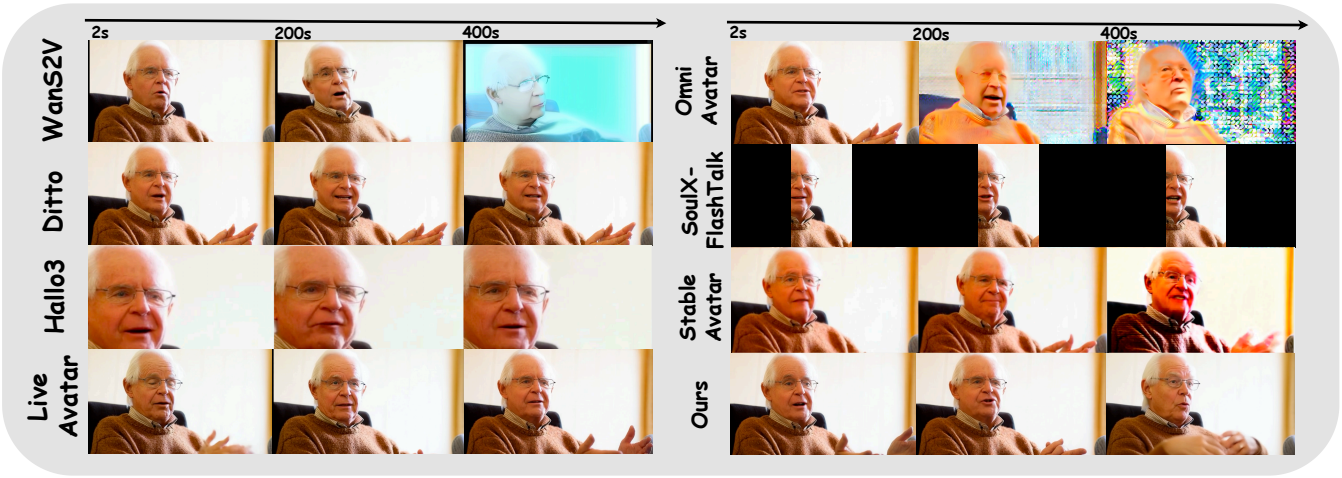

Figure 4: Qualitative comparison on GenBench-LongVideo. Frames sampled at 2 s, 200 s, and 400 s from the same driving audio. Please refer to the supplementary video for dynamics comparison.

GenBench-LongVideo. Table 1 (bottom) shows that the dynamics advantage of DynaForcing holds on long-form generation (>5 min). LiveAvatar's ExpVar further drops from 0.69 (short) to 0.57 (long) as static representations accumulate in the KV cache, while DynaForcing maintains strong dynamics (ExpVar 1.93) alongside the best visual quality and synchronization.

Qualitative results. Figure 4 shows representative frames from all compared methods. We refer readers to the supplementary video for a more faithful comparison of expression dynamics, which static frames cannot fully convey.

Training stability. Figure 5 shows the training dynamics of our full DynaForcing. Unlike vanilla self-forcing (Figure 2b), where dynamics metrics peak early and then collapse, DynaForcing's Sync-C and ExpVar rise within the first ~1,000 steps and remain stable throughout training. Meanwhile, quality metrics (ASE, IQA) steadily improve and plateau around 4,000 steps. This confirms that our method resolves the quality–dynamics trade-off throughout the entire training process, rather than relying on early stopping at a favorable checkpoint.

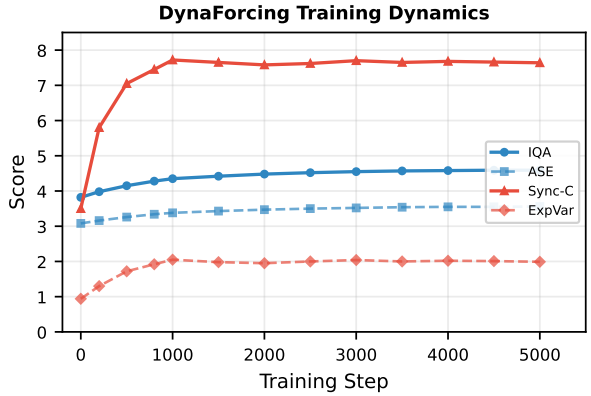

Figure 5: Training dynamics of DynaForcing. Unlike vanilla self-forcing (Figure 2b), dynamics metrics (Sync-C, ExpVar) stabilize after ~1K steps without degradation, while quality metrics improve steadily.

5.3 Ablation Studies

All ablations are evaluated on GenBench-ShortVideo unless otherwise noted.

Diagnostic ablation: isolating rollout feedback. To test our hypothesis that unanchored self-conditioning amplifies collapse (§4.1), we compare two training paradigms under identical settings (4,000 steps, same DMD loss, no DynaForcing components): (i) *Self-forcing*, where each block's context is built from the student's own outputs; (ii) *GT-anchored (CausVid-style)*, which follows CausVid [39] and processes all blocks in parallel on noised ground-truth latents with a blockwise causal mask. The core difference between the two paradigms is the conditioning source (student output vs. ground-truth).

Table 2: Diagnostic ablation: isolating rollout feedback. GT-anchored training conditions on ground-truth latents; self-forcing conditions on the student's own outputs.

Configuration	ASE↑	Sync-C↑	ExpVar↑	Dyn-Deg↑
Self-forcing	**3.44**	5.48	0.69	0.31
GT-anchored (CausVid-style)	3.38	**6.85**	**1.36**	**0.52**

Table 2 reveals a stark contrast. Self-forcing achieves slightly higher visual quality (ASE) but severely suppressed dynamics (Sync-C: 5.48, Dyn-Deg: 0.31), confirming dynamic collapse under unanchored self-conditioning. GT-anchored training, which anchors every block to ground-truth context, recovers substantial dynamics (ExpVar: 0.69→1.36, Dyn-Deg: 0.31→0.52) and lip-sync accuracy (Sync-C: 5.48→6.85). This supports our claim that the conditioning source, not DMD distillation itself, is the primary driver of collapse.

Table 3: Component ablation. Each row removes one component from full DynaForcing.

Configuration	ASE↑	Sync-C↑	ExpVar↑
DynaForcing w/o Hybrid Forcing	**3.57**	6.78	1.18
DynaForcing w/o Reward Reg.	3.55	7.35	1.75
DynaForcing w/o Ref. Perturbation	3.52	7.58	1.92
DynaForcing (full)	3.55	**7.68**	**2.02**

Table 4: Effect of Hybrid Forcing probability p_{data} on full DynaForcing (varying p_{data} only).

p_{data}	ASE↑	Sync-C↑	ExpVar↑	Dino-S↑
0.1	**3.56**	7.19	1.42	**0.96**
0.3	3.55	**7.68**	2.02	**0.96**
0.5	3.53	7.64	2.03	0.95
1.0	3.42	7.62	**2.10**	0.93

Table 5: Training efficiency: full DynaForcing with and without our efficiency optimizations (§4.5), both trained for 4,000 steps.

	GPUs	GPU·h	ASE↑	Sync-C↑	ExpVar↑
w/o eff. optim.	128×H100	7,111	**3.56**	7.66	**2.04**
w. eff. optim.	**8×H100**	**667**	3.55	**7.68**	2.02

However, GT-anchored training alone does not fully close the gap to the teacher (ExpVar still below 1.81), indicating that reverse-KL mode-seeking contributes independently.

Component-wise ablation. Table 3 presents component contributions. Removing Hybrid Forcing causes the largest drop in both dynamics (ExpVar 2.02→1.18) and lip-sync (Sync-C 7.68→6.78), confirming data-level anchoring as the most critical component; notably, ASE slightly increases (3.55→3.57), consistent with the quality–dynamics trade-off observed in the p_{data} sweep. Removing Reward Regularization most degrades Sync-C among the remaining components (7.68→7.35), as it directly optimizes lip-sync via the dynamics reward. Removing Reference Perturbation has the smallest individual effect but still contributes (ExpVar 2.02→1.92). All three together achieve the best overall balance.

Hybrid Forcing probability. Table 4 sweeps the mixing probability. $p_{\text{data}} = 0.3$ achieves the best overall balance. Reducing p_{data} to 0.1 yields slightly higher visual quality (ASE 3.56) but substantially degrades lip-sync (Sync-C 7.68→7.19) and dynamics (ExpVar 2.02→1.42), as the model receives insufficient ground-truth anchoring. Increasing p_{data} beyond 0.3 continues to improve ExpVar (reaching 2.10 at $p_{\text{data}} = 1.0$) thanks to stronger ground-truth anchoring, but at the cost of significant train-test mismatch: the model gets less self-forcing practice, degrading visual quality (ASE 3.55→3.42) and identity consistency (Dino-S 0.96→0.93) at inference. $p_{\text{data}} = 0.3$ thus strikes the best balance between dynamics recovery and generation quality.

Table 6: User study: preference rate (%) for DynaForcing over competing realtime methods.

vs.	Visual	Lip-Sync	Motion	Overall
LiveAvatar	58.4	71.2	80.6	72.8
SoulX-FlashTalk	55.2	64.8	74.2	66.4

Training efficiency. Table 5 compares full DynaForcing trained with and without our efficiency optimizations. Graph pruning combined with block-wise gradient replay reduces the GPU requirement from 128 to 8 H100s, and total GPU-hours from 7,111 to 667 (~10.7× reduction) despite a ~1.5× per-step wall-clock overhead from replay. Quality metrics remain essentially unchanged (ASE differs by 0.01, Sync-C by 0.02, ExpVar by 0.02), confirming negligible degradation from the approximations.

5.4 User Study

We conduct a pairwise user study comparing DynaForcing against the two strongest competing realtime methods (LiveAvatar and SoulX-FlashTalk). 25 participants evaluate 20 randomly selected GenBench-ShortVideo samples across four dimensions: visual quality, lip-sync accuracy, motion naturalness, and overall preference. Videos are presented side-by-side with randomized left/right placement and synchronized audio. Paired t-tests confirm statistical significance ($p < 0.01$) across all dimensions.

Table 6 shows that DynaForcing is consistently preferred, with the largest advantage on motion naturalness (80.6% vs. LiveAvatar). Visual quality preference is more modest (58.4%), consistent with the small ASE/IQA gap in Table 1.

6 Conclusion

In the context of audio-driven avatar generation, we have identified and empirically characterized *dynamic collapse*, a failure mode of self-forcing distillation where the reverse KL objective and autoregressive feedback jointly drive the student toward near-static outputs, suppressing the fine-grained temporal dynamics essential for faithful talking-head synthesis. To address this, we proposed **DynaForcing**, combining three strategies at different levels of the training pipeline: Hybrid Forcing (data-level), Dynamics-Aware Reward Regularization (loss-level), and Reference Perturbation (conditioning-level). Together with computation graph pruning and gradient replay that reduce GPU requirements by ~10×, DynaForcing makes dynamically-faithful self-forcing training both more effective and accessible. Experiments demonstrate substantial improvements in motion naturalness and audio-visual faithfulness while preserving visual quality.

Limitations. Our dynamics reward relies on pretrained SyncNet and 3DMM expression estimation models, whose biases may influence training outcomes. The current analysis focuses on audio-driven avatar generation; while we hypothesize that dynamic collapse occurs more broadly in video self-forcing, systematic evaluation across other domains (e.g., text-to-video) is left for future work. The ~1.5× wall-clock overhead from gradient replay, while acceptable, may be further reducible with more sophisticated memory management strategies.

References

[1] Tim Brooks, Bill Peebles, Connor Holmes, et al. 2024. Video generation models as world simulators. OpenAI Technical Report. https://openai.com/research/video-generation-models-as-world-simulators

[2] Joon Son Chung and Andrew Zisserman. 2017. Out of time: automated lip sync in the wild. In *Computer Vision–ACCV 2016 Workshops*. Springer, 251–263.

[3] Jiahao Cui, Hui Li, Yun Zhan, Hanlin Shang, Kaihui Cheng, Yuqi Ma, Shan Mu, Hang Zhou, Jingdong Wang, and Siyu Zhu. 2025. Hallo3: Highly dynamic and realistic portrait image animation with video diffusion transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 21086–21095.

[4] Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. 2025. Self-Forcing++: Towards Minute-Scale High-Quality Video Generation. *arXiv preprint arXiv:2510.02283* (2025).

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4690–4699.

[6] Ariel Ephrat, Inbar Mosseri, Oran Lang, et al. 2018. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619* (2018).

[7] Qijun Gan, Ruizi Yang, Jianke Zhu, Shaofei Xue, and Steven Hoi. 2025. Omni-Avatar: Efficient Audio-Driven Avatar Video Generation with Adaptive Body Animation. *arXiv preprint arXiv:2506.18866* (2025).

[8] Xin Gao, Li Hu, Siqi Hu, et al. 2025. Wan-S2V: Audio-Driven Cinematic Video Generation. arXiv:2508.18621

[9] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. 2025. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion. *arXiv preprint arXiv:2506.08009* (2025).

[10] Yubo Huang, Hailong Guo, Fangtai Wu, Shifeng Zhang, Shijie Huang, Qijun Gan, Lin Liu, Sirui Zhao, Enhong Chen, Jiaming Liu, and Steven Hoi. 2025. Live Avatar: Streaming Real-time Audio-Driven Avatar Generation with Infinite Length. *arXiv preprint arXiv:2512.04677* (2025).

[11] Ziqi Huang, Yinan He, et al. 2023. VBench: Comprehensive Benchmark Suite for Video Generative Models. arXiv:2311.17982

[12] Taekyung Ki, Sangwon Jang, Jaehyeong Jo, Jaehong Yoon, and Sung Ju Hwang. 2026. Avatar Forcing: Real-Time Interactive Head Avatar Generation for Natural Conversation. *arXiv preprint arXiv:2601.00664* (2026).

[13] Tianqi Li, Ruobing Zheng, Minghui Yang, Jingdong Chen, and Ming Yang. 2024. Ditto: Motion-space diffusion for controllable realtime talking head synthesis. *arXiv preprint arXiv:2411.19509* (2024).

[14] Jinxiu Liu, Xuanming Liu, Kangfu Mei, Yandong Wen, Ming-Hsuan Yang, and Weiyang Liu. 2026. Streaming Autoregressive Video Generation via Diagonal Distillation. In *ICLR*. arXiv:2603.09488.

[15] Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. 2025. Rolling Forcing: Autoregressive Long Video Diffusion in Real Time. *arXiv preprint arXiv:2509.25161* (2025).

[16] Yunhong Lu, Yanhong Zeng, Haobo Li, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jiapeng Zhu, Hengyuan Cao, Zhipeng Zhang, Xing Zhu, Yujun Shen, and Min Zhang. 2025. Reward Forcing: Efficient Streaming Video Generation with Rewarded Distribution Matching Distillation. *arXiv preprint arXiv:2512.04678* (2025).

[17] Weijian Luo. 2024. Diff-instruct++: Training one-step text-to-image generator model to align with human preferences. *arXiv preprint arXiv:2410.18881* (2024).

[18] Yihong Luo, Tianyang Hu, Jiacheng Sun, Yujun Cai, and Jing Tang. 2025. Learning Few-Step Diffusion Models by Trajectory Distribution Matching. *arXiv preprint arXiv:2503.06674* (2025).

[19] Rang Meng, Xingyu Zhang, Yuming Li, and Chenguang Ma. 2025. EchoMimicV2: Towards Striking, Simplified, and Semi-Body Human Animation. arXiv:2411.10061

[20] Kevin P Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. MIT press.

[21] Maxime Oquab, Timothée Darcet, Théo Moutakanni, et al. 2024. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research* (2024).

[22] Jan Peters and Stefan Schaal. 2008. Reinforcement learning of motor skills with policy gradients. *Neural Networks* 21, 4 (2008), 682–697.

[23] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. 2020. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM international conference on multimedia*. 484–492.

[24] Tim Salimans and Jonathan Ho. 2022. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*.

[25] Le Shen, Qian Qiao, Tan Yu, Ke Zhou, Tianhang Yu, Yu Zhan, Zhenjie Wang, Ming Tao, Shunshun Yin, and Siyuan Liu. 2025. SoulX-FlashTalk: Real-Time Infinite Streaming of Audio-Driven Avatars via Self-Correcting Bidirectional Distillation. *arXiv preprint arXiv:2512.23379* (2025).

[26] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. 2023. Consistency models. In *International Conference on Machine Learning (ICML)*.

[27] Zhiyao Sun et al. 2025. StreamAvatar: Streaming Diffusion Models for Real-Time Interactive Human Avatars. *arXiv preprint arXiv:2512.22065* (2025).

[28] Image Team et al. 2025. Z-Image: An Efficient Image Generation Foundation Model with Single-Stream Diffusion Transformer. arXiv:2511.22699

[29] Linrui Tian, Qi Wang, Bang Zhang, and Liefeng Bo. 2024. Emo: Emote portrait alive generating expressive portrait videos with audio2video diffusion model under weak conditions. In *European Conference on Computer Vision*. Springer, 244–260.

[30] Shuyuan Tu, Yueming Pan, Yinming Huang, Xintong Han, Zhen Xing, Qi Dai, Chong Luo, Zuxuan Wu, and Yu-Gang Jiang. 2025. Stableavatar: Infinite-length audio-driven avatar video generation. *arXiv preprint arXiv:2508.08248* (2025).

[31] Team Wan, Ang Wang, Baole Ai, et al. 2025. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314* (2025).

[32] Lizhen Wang, Yongming Zhu, Zhipeng Ge, Youwei Zheng, Longhao Zhang, and Tianshu Hu. 2026. FlowAct-R1: Towards Interactive Humanoid Video Generation. *arXiv preprint arXiv:2601.10103* (2026).

[33] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.

[34] Chenfei Wu et al. 2025. Qwen-Image Technical Report. arXiv:2508.02324

[35] Haoning Wu, Zicheng Zhang, Weixia Zhang, et al. 2023. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *arXiv preprint arXiv:2312.17090* (2023).

[36] Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. 2025. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622* (2025).

[37] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. 2024. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems* 37 (2024), 47455–47487.

[38] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. 2024. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6613–6623.

[39] Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. 2025. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 22963–22974.

[40] Wenxuan Zhang, Xiaodong Cun, Xuan Wang, Yong Zhang, Xi Shen, Yu Guo, Ying Shan, and Fei Wang. 2023. Sadtalker: Learning realistic 3d motion coefficients for stylized audio-driven single image talking face animation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8652–8661.

[41] Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. 2026. Causal Forcing: Autoregressive Diffusion Distillation Done Right for High-Quality Real-Time Interactive Video Generation. *arXiv preprint arXiv:2602.02214* (2026).

[42] Yongming Zhu, Longhao Zhang, Zhengkun Rong, Tianshu Hu, Shuang Liang, and Zhipeng Ge. 2025. INFP: Audio-driven interactive head generation in dyadic conversations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 10667–10677.