BRAVO:
Bridge Matching for Autoregressive Video Generation

Shihua Zhang* **Zhenxiong Tan*** **Xinchao Wang†**
Department of Electrical and Computer Engineering
National University of Singapore
suhzhang001@gmail.com zhenxiong@u.nus.edu xinchao@nus.edu.sg



Figure 1: Videos generated by noise-to-data (1st row) and data-to-data (2nd row) methods conditioned on the same text prompt and input image (left column, input image is generated by GPT-5.5 [30]). Noise-to-data produces increasing facial distortion in later frames, while our method produces less.

Abstract

Image-to-video generation aims to animate an image while preserving appearance details and producing visually coherent future frames. Autoregressive generation models this process as a sequence of next-frame predictions conditioned on the input image and text prompt. However, advanced autoregressive generators built on diffusion or flow models formulate each step as a noise-to-data process, despite the fact that the previous frame already provides an informative visual prior. This mismatch makes next-frame synthesis harder than necessary and can degrade prediction quality, since the model must recover scene layout and visual context from uninformative Gaussian noise at every step. To this end, we introduce **BRAVO**, a Brownian bridge matching framework for autoregressive video generation. BRAVO replaces the conventional noise-to-data path with a data-to-data bridge from the previous frame to the current one, turning next-frame synthesis into local frame-to-frame transport. The bridge state is formed by interpolating between adjacent real frames with a Brownian perturbation, so the learned velocity models the local transition from the previous frame to the next under the text prompt and causal visual history. BRAVO is trained with teacher-forcing and sampled autoregressively, where each bridge starts from the last generated frame and reuses historical messages through a causal KV-cache. Experiments on different datasets show that BRAVO surpasses the conventional noise-to-data baseline and achieves even competitive performance with substantially larger models. Code is available at `https://github.com/SuhZhang/BRAVO`.

*Equal contribution.
†Corresponding author.

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

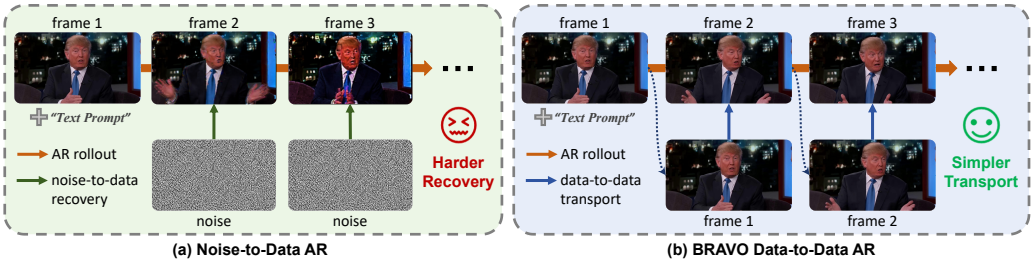

Figure 2: Comparison between normal noise-to-data and our data-to-data AR generation paradigms. (a) Conventional AR diffusion or flow generation predicts each future frame through a hard noise-to-data recovery path. (b) BRAVO instead uses the previous frame as the source state and models the next step as a simpler Brownian bridge-based data-to-data transport.

1 Introduction

Image-to-video (I2V) generation aims to generate a video sequence from a static input image, and recent generative models have made rapid progress on this task, producing videos with increasingly realistic content [3, 48, 41, 25, 43]. In this setting, the conditioning image acts as the visual anchor for the whole video, providing the objects, scene layout, and appearance that later frames should inherit [29, 5]. A practical solution for this problem is autoregressive (AR) video generation, which synthesizes the video causally through a chain of conditional transitions [1, 50]. Specifically, at each transition, the model observes the input image, the text prompt, and the frames that have been produced, allowing the generated visual history to guide next-frame prediction. However, AR rollout is sensitive to prediction errors at each step, where small artifacts, appearance drift, or inaccurate motion in intermediate predictions may be carried forward and amplified as the sequence grows [4].

To mitigate this issue, advanced methods therefore focus on improving the quality of AR video generation through several approaches, including causal architectures and training strategies that reduce the gap between training and self-conditioned inference [7, 16, 54]. These designs make AR rollout more practical, but they leave a more basic issue less explored. The generative process of each conditional transition is commonly borrowed from standard diffusion or flow models, where the model starts from Gaussian noise [13, 36, 21] and transports it toward the target frame under image, text, and historical conditions [5]. This noise-to-data trajectory is powerful for general generation, yet it is not the most natural choice for causal I2V synthesis. As illustrated in Figure 2(a), the input image and text prompt guide the denoising process, while the source of each next-frame trajectory is uninformative noise. However, when frame k is predicted, the previous frame $k-1$ already provides strong visual cues needed for the next frame. Starting again from Gaussian noise forces the model to recover visual information that is already available in the rollout history, which makes each transition unnecessarily difficult and can degrade next-frame prediction quality. This raises the central question of this work. *Can the previous frame serve as the source state for each AR transition, rather than merely as a condition signal, so that generation no longer has to recover from uninformative noise?*

Answering this question is not a simple change of the source state during the generation transition. The model needs a principled path between adjacent frames, a learning objective that describes the local transition along this path, and an inference procedure that can continue from generated frames while preserving the causal structure of AR generation. To this end, we introduce bridge matching for autoregressive video generation, abbreviated as **BRAVO**. BRAVO formulates each AR rollout step as Brownian bridge matching [20] between neighboring frames. Concretely, as shown in Figure 2(b), BRAVO uses the previous frame as the start endpoint and connects it to the next frame with a Brownian bridge, where intermediate states are sampled along this Brownian-perturbed data-to-data path. The model learns the corresponding velocity field conditioned on text and the input image, and each step focuses on the adjacent-frame transport rather than recovering from an unstructured Gaussian distribution. During inference, the latest generated frame becomes the new source endpoint of the next bridge, and historical frames are reused through a causal KV-cache. In this way, BRAVO keeps the practical rollout structure of AR video generation while turning each step from noise-to-data recovery into data-to-data transport, allowing the model to build on the existing visual prior and focus on the short-term changes needed for the next frame.

In summary, our main contributions are as follows.

- We identify the noise source in AR I2V transitions as a key bottleneck, since it ignores the informative visual prior already present in the historical frames.
- We propose **BRAVO**, which replaces noise-to-data next-frame prediction with Brownian bridge data-to-data transport between adjacent frames.
- We provide a practical training and inference scheme with causal teacher-forcing, weighted bridge velocity learning, and AR rollout with cached history.
- Experiments on VBench-I2V and reference-based rollout evaluation show improved quality and prediction stability over the noise-to-data AR baseline, with even competitive performance against substantially larger models.

2 Related Work

2.1 Image-to-video generation

Recent I2V generation can be broadly grouped into diffusion-based and autoregressive models, which differ in whether they generate videos through global denoising or causal rollout.

Diffusion models. Diffusion-based I2V models generate videos by denoising a global spatiotemporal latent from Gaussian noise under image and text conditions. Early methods adapt image or text-to-video diffusion backbones with temporal modules, image-conditioning injection, or motion-control mechanisms [29, 5, 8, 52, 47, 33, 11, 10, 9, 27, 32, 45, 46, 31, 34], while recent large-scale models further improve fidelity and prompt following through stronger video backbones and larger-scale training data [15, 40, 19]. Different from these diffusion methods, BRAVO focuses on AR rollout and applies a Brownian bridge from the previous frame to the current at each local transition.

Autoregressive models. Autoregressive video generation produces future frames or chunks causally from the generated history. Existing work improves this paradigm either by refining the causal diffusion formulation or by scaling AR video backbones. Diffusion-Forcing [7] trains a causal sequence model to denoise tokens with independent noise levels, thereby combining autoregressive next-token prediction with full-sequence diffusion. Self-Forcing [16], CausVid [49], and Causal-Forcing [54] further study train-test mismatch, causalization, and distillation in AR video diffusion. Other large-scale AR methods expand the paradigm in complementary directions: Cosmos targets physical world modeling [1], Magi-1 uses chunk-wise autoregressive diffusion for streaming generation [39], and VideoMAR combines temporal causal rollout with spatial masked generation over continuous tokens for I2V [50]. Unlike these autoregressive models, BRAVO focuses on the transport path of each local transition, replacing noise-to-data prediction with a Brownian bridge from the previous frame.

2.2 Bridge matching in vision applications

Bridge matching learns a generative transport path between two informative endpoints rather than from a fixed noise prior [2, 23, 35]. In image tasks, bridge formulations are commonly built between fixed source and target domains or between degraded and clean images, covering restoration, translation, and generation [20, 22, 53, 6]. For video-oriented bridge methods, most focus on video transformation tasks such as interpolation or translation [26, 38]. In particular, Lyu et al. [26] use consecutive Brownian bridges for interpolation between observed past and future frames, reducing sampling variance to reconstruct an intermediate frame. For I2V generation, FrameBridge [42] builds a global bridge from the repeated first frame to the target video. BRAVO instead uses bridge modeling inside AR generation, where no future boundary is observed and each predicted frame becomes the source of the next bridge. This exploits the data-to-data nature of bridges to preserve the visual context already present in the rollout, making next-frame synthesis easier and improving generation quality.

3 BRAVO

In this section, we present BRAVO, a Brownian-bridge formulation for autoregressive (AR) image-to-video (I2V) generation. We first formalize the latent AR generation problem (Section 3.1), and introduce how to apply bridge matching between adjacent video frames (Section 3.2). Then we

describe how BRAVO is trained with teacher-forcing under a causal mask (Section 3.3), and finally present the AR inference procedure with cached history (Section 3.4).

3.1 Problem formulation

The I2V task takes a conditioning image $I_1 \in \mathbb{R}^{3\times H\times W}$ and a text prompt y as input, and aims to generate future frames $I_{2:T} = (I_2, \ldots, I_T)$ such that the complete video $I_{1:T}$ is visually coherent and semantically consistent with the conditioning inputs [48, 41]. Formally, the target distribution is:

$$p(I_{2:T} \mid I_1, y). \tag{1}$$

Directly modeling this distribution in pixel space is expensive and requires the model to represent substantial low-level visual redundancy. We therefore follow the standard latent-generation paradigm and encode the video in the latent space of a frozen video VAE encoder $\mathcal{E}$:

$$\mathbf{z}^{1:T} = \mathcal{E}(I_{1:T}), \quad \mathbf{z}^k \in \mathbb{R}^{C\times H_z\times W_z}, \tag{2}$$

where $\mathbf{z}^k$ denotes the latent slice associated with frame I_k. At inference time, only the conditioning image is observed, and its encoded latent $\mathbf{z}^1 = \mathcal{E}(I_1)$ serves as the anchor from which the remaining latent frames are generated. Under this representation, BRAVO models the latent I2V process autoregressively:

$$p(\mathbf{z}^{2:T} \mid \mathbf{z}^1, y) = \prod_{k=2}^{T} p(\mathbf{z}^k \mid c^k), \quad c^k = (\mathbf{z}^{<k}, y), \tag{3}$$

where $\mathbf{z}^{<k} = (\mathbf{z}^1, \ldots, \mathbf{z}^{k-1})$ is the causal history and y is the text prompt. In this work, we focus on this frame-wise rollout setting, where each autoregressive step predicts the next latent frame. Conventionally, such a step is modeled by a flow-matching process that generates $\mathbf{z}^k$ from Gaussian noise [21, 54]. BRAVO takes a different path: it generates $\mathbf{z}^k$ from the previous frame $\mathbf{z}^{k-1}$ via Brownian bridge matching [20, 53], turning each autoregressive step into a local, frame-to-frame transport problem, which will be introduced in detail in the following.

3.2 Bridge matching between adjacent frames

We begin with the per-step bridge construction, which is the main difference from standard autoregressive diffusion- or flow-based methods. Specifically, at each autoregressive step $k \geq 2$, the bridge endpoints are defined as $\mathbf{x}_0 = \mathbf{z}^{k-1}$ (the source, *i.e.*, the previous frame) and $\mathbf{x}_1 = \mathbf{z}^k$ (the target, *i.e.*, the next frame to be generated). We use a continuous bridge time $t \in [0, 1]$, where $t{=}0$ corresponds to the source and $t{=}1$ to the target; generation proceeds from $t{=}0$ toward $t{=}1$. To unify the notation of the bridge matching process, all equations in this section are written in terms of $\mathbf{x}_0$ and $\mathbf{x}_1$. We first review conventional noise-to-data flow matching as a baseline, and then give the Brownian bridge dynamics and training target used by BRAVO.

Noise-to-data flow matching. In standard noise-to-data flow matching [23], each autoregressive step models $p(\mathbf{z}^k \mid c^k)$ by transporting isotropic Gaussian noise to the target frame. The source $\mathbf{x}_0 = \boldsymbol{\epsilon}$ carries no information about the video, so the model must reconstruct the entire frame from scratch at every step. Generation follows the ODE:

$$d\mathbf{x}_t = \mathbf{v}_t\, dt, \quad \mathbf{v}_t = \frac{\mathbf{x}_1 - \mathbf{x}_t}{1-t}, \quad \mathbf{x}_{t=0} = \boldsymbol{\epsilon}, \quad t \in [0, 1), \tag{4}$$

where $\mathbf{v}_t$ denotes the time-dependent velocity field that transports the current state $\mathbf{x}_t$ toward the target endpoint $\mathbf{x}_1 = \mathbf{z}^k$ [21, 23]. The corresponding marginal path is $\mathbf{x}_t = (1{-}t)\,\boldsymbol{\epsilon} + t\,\mathbf{z}^k$, so the intermediate states are blends of noise and data that do not correspond to meaningful visual content.

However, in AR video generation, the model is not generating frames in isolation but rather *continuing* a video from an already informative visual context. The previous frame $\mathbf{z}^{k-1}$ shares scene layout, appearance, and local visual context with the target $\mathbf{z}^k$, so the inter-frame difference $\|\mathbf{z}^k - \mathbf{z}^{k-1}\|$ is typically much smaller than the full data-to-noise distance $\|\mathbf{z}^k - \boldsymbol{\epsilon}\|$. BRAVO exploits this structure by replacing the uninformative noise source with the previous frame, turning each step into a *local* transport problem.

Brownian bridge SDE. Concretely, BRAVO connects $\mathbf{x}_0 = \mathbf{z}^{k-1}$ to $\mathbf{x}_1 = \mathbf{z}^k$ via a Brownian bridge SDE [2, 42]:

$$d\mathbf{x}_t = \frac{\mathbf{x}_1 - \mathbf{x}_t}{1-t}\, dt + \boldsymbol{\sigma} \odot d\mathbf{W}_t, \quad \mathbf{x}_{t=0} = \mathbf{x}_0, \quad t \in [0, 1), \tag{5}$$

where $\mathbf{W}_t$ is a standard Wiener process and $\boldsymbol{\sigma} \in \mathbb{R}^C$ is a fixed per-channel noise scale hyperparameter. The drift $(\mathbf{x}_1 - \mathbf{x}_t)/(1-t)$ always points toward the target $\mathbf{x}_1$ and pins the process to the endpoint, so $\lim_{t\to 1} \mathbf{x}_t = \mathbf{x}_1$. The diffusion term $\boldsymbol{\sigma} \odot d\mathbf{W}_t$ injects stochasticity along the path. Setting $\boldsymbol{\sigma} = \mathbf{0}$ will reduce the SDE to a deterministic ODE of rectified flow [23] between adjacent frames. Conversely, as $\boldsymbol{\sigma}$ grows larger, the path becomes increasingly stochastic.

Then, the SDE in Eq. (5) admits the following closed-form marginal distribution at any time t:

$$\mathbf{x}_t = (1-t)\,\mathbf{x}_0 + t\,\mathbf{x}_1 + (\boldsymbol{\sigma} \odot \boldsymbol{\epsilon})\sqrt{t(1-t)}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{6}$$

Unlike the noise-to-data case, where $\mathbf{x}_t$ interpolates between noise and data, the bridge state $\mathbf{x}_t$ is a convex combination of two *real* frames plus a small stochastic perturbation, so every intermediate state remains a meaningful blend of adjacent video frames. The variance $t(1-t)$ vanishes at both endpoints, so $\mathbf{x}_0$ is recovered exactly at $t{=}0$ and $\mathbf{x}_1$ at $t{=}1$. The drift of the SDE is precisely the velocity target that we train the model to predict:

$$\mathbf{v}_t^\star = \frac{\mathbf{x}_1 - \mathbf{x}_t}{1-t}, \qquad \mathbf{x}_1 = \mathbf{x}_t + (1{-}t)\,\mathbf{v}_t^\star, \tag{7}$$

where the second expression shows that the target frame is always recoverable from any bridge state and its velocity. With Eqs. (6) and (7), this bridge construction provides the two quantities needed during learning: sampled intermediate states $\mathbf{x}_t$ and their supervised velocities $\mathbf{v}_t^\star$. We next describe how BRAVO trains a causal transformer to predict these per-frame velocities while preventing access to the clean target frame.

3.3 Training with teacher-forcing

With the bridge target fixed in (7), the training reduces to supervised velocity prediction under autoregressive conditioning. We train a causal transformer f_θ to predict the bridge velocity. At frame k and bridge time t, the model takes the bridge state, the bridge time, and the causal context as input and predicts:

$$\hat{\mathbf{v}}_t^k = f_\theta(\mathbf{x}_t,\, t,\, c^k), \quad c^k = (\mathbf{z}^{<k},\, y), \tag{8}$$

where $\hat{\mathbf{v}}_t^k$ denotes the velocity predicted for frame k at bridge time t.

During training, the model has access to ground-truth past frames. To provide each frame's bridge prediction with clean conditioning context (rather than other frames' noisy bridge states), in practice, we structure an input sequence $Z^{(0)}$ that contains two sets: a clean set $\mathcal{Z}$ of ground-truth frames and a noisy set $\mathcal{X}$ of each step's bridge states uniformly sampled on the corresponding trajectory, which is presented as follows:

$$Z^{(0)} = \big[\underbrace{\mathbf{z}^1, \ldots, \mathbf{z}^T}_{\text{clean set } \mathcal{Z}},\ \underbrace{\mathbf{x}_{t_1}^1,\, \mathbf{x}_{t_2}^2, \ldots, \mathbf{x}_{t_T}^T}_{\text{noisy set } \mathcal{X}}\big], \tag{9}$$

where, for each frame $k \geq 2$, $\mathbf{x}_{t_k}^k$ denotes the bridge state calculated by Eq. (6) at an independently sampled bridge time $t_k \sim \mathcal{U}(0, 1)$, with the per-frame endpoints $\mathbf{x}_0{=}\mathbf{z}^{k-1}$ and $\mathbf{x}_1{=}\mathbf{z}^k$. The first frame is the conditioning image with no bridge process, so we set $\mathbf{x}_{t_1}^1 \equiv \mathbf{z}^1$ for all t_1 and include it in $\mathcal{X}$ to maintain a uniform sequence structure, while cutting out its gradient during training. The information flow is controlled by a blockwise causal mask M. Particularly, write f_i for the frame index of token i and $s_i \in \{\mathcal{Z}, \mathcal{X}\}$ for its set membership. The mask M is defined as:

$$M_{ij} = (s_i = s_j = \mathcal{Z},\ f_j \leq f_i) \ \vee\ (s_i = \mathcal{X},\ s_j = \mathcal{Z},\ f_j < f_i) \ \vee\ (s_i = s_j = \mathcal{X},\ f_j = f_i) \tag{10}$$

The mask here is designed to preserve teacher-forced context without leaking the target frame. Clean tokens form a causal history over ground-truth frames, while the noisy tokens for frame k can attend to earlier clean frames and to their own same-frame noisy states, but not to the clean target $\mathbf{z}^k$ or to any future frame. This prevents a degenerate shortcut in which the model copies $\mathbf{z}^k$ from the clean stream instead of learning the bridge velocity. Details of the masked-attention update are seen in

Algorithm 1 BRAVO Training

Require: latents $\mathbf{z}^{1:T}$, prompt y, scale $\boldsymbol{\sigma}$
  initialize $\mathcal{L} \leftarrow 0$
  for $k = 2$ to T **do**
    $\mathbf{x}_0 \leftarrow \mathbf{z}^{k-1},\ \mathbf{x}_1 \leftarrow \mathbf{z}^k$
    sample $t \sim \mathcal{U}(0,1),\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    sample $\mathbf{x}_t$ by Eq. (6)
    $\hat{\mathbf{v}}_t^k \leftarrow f_\theta(\mathbf{x}_t, t, (\mathbf{z}^{<k}, y))$
    $\mathcal{L} \leftarrow \mathcal{L} + (1-t)\|\hat{\mathbf{v}}_t^k - \mathbf{v}_t^{\star,k}\|_2^2$
  end for
  return $\mathcal{L}/(T-1)$

Algorithm 2 BRAVO Inference

Require: latent $\hat{\mathbf{z}}^1$, prompt y, schedule $\{t_n\}_{n=0}^N$, ω, $\boldsymbol{\sigma}$
  initialize KV-cache with $\hat{\mathbf{z}}^1$ at $t = 0$ by Eq. (13)
  for $k = 2$ to T **do**
    $\mathbf{x}^{(0)} \leftarrow \hat{\mathbf{z}}^{k-1}$
    for $n = 0$ to $N-1$ **do**
      compute $\hat{\mathbf{v}}^{(n)}$ by Eq. (13); update $\mathbf{x}^{(n+1)}$ by Eq. (12)
    end for
    $\hat{\mathbf{z}}^k \leftarrow \mathbf{x}^{(N)}$; append it to KV-cache at $t = 0$ by Eq. (13)
  end for
  return $\hat{\mathbf{z}}^{1:T}$

Appendix A.1. Note that the mask only governs visual self-attention. The text prompt y is injected through a separate cross-attention layer and is visible to all visual tokens. After the transformer blocks, we discard the $\mathcal{Z}$ part of the final sequence and apply the output head only to the $\mathcal{X}$ part.

Given this leakage-free conditioning structure, the optimization objective can be written directly as a frame-wise bridge-matching loss with a weighted velocity error objective:

$$\mathcal{L} = \frac{1}{T-1} \sum_{k=2}^{T} w(t_k) \left\| \hat{\mathbf{v}}_{t_k}^k - \mathbf{v}_{t_k}^{\star,k} \right\|_2^2, \quad w(t) = 1 - t, \tag{11}$$

where $\hat{\mathbf{v}}_{t_k}^k$ is the predicted velocity at timestamp t_k as defined in Eq. (8), $\mathbf{v}_{t_k}^{\star,k}$ is the corresponding target velocity in Eq. (7). Since each t_k is sampled randomly, Eq. (11) can theoretically supervise the velocity over the full bridge path. The weight $w(t) = 1-t$ downweights samples near t=1, where the bridge state is already close to the target and the stochastic component of the velocity dominates. This weighting controls the contribution of high-variance targets to the loss (Appendix A.2) to avoid overwhelming the gradient, and its empirical effect is evaluated in Section 4.3. The complete training procedure is given in Algorithm 1.

3.4 Autoregressive inference

We now turn to the AR rollout with BRAVO, conditioning on its own generated history because of the lack of ground truth. Thus, the clean set $\mathcal{Z}$ in Eq. (9) is replaced by a causal KV-cache built from the model's own predictions. For frame $k \geq 2$, we set the bridge source to the model's previous prediction $\mathbf{x}_0 = \hat{\mathbf{z}}^{k-1}$ (where $\hat{\mathbf{z}}^1 = \mathbf{z}^1$ is the image condition) and integrate the learned bridge dynamics from t=0 to t=1. Denoting the discrete solver state at time t_n by $\mathbf{x}^{(n)}$, we initialize $\mathbf{x}^{(0)} = \hat{\mathbf{z}}^{k-1}$ and take the terminal state $\mathbf{x}^{(N)}$ as the predicted latent frame $\hat{\mathbf{z}}^k$.

Sampling discretization. To make the bridge sampler explicit, we use a first-order stochastic discretization of the SDE. Given a schedule $0{=}t_0 < t_1 < \cdots < t_N{=}1$, we discretize Eq. (5) using an Euler-Maruyama-style update with a Brownian-bridge correction [18, 38]:

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \underbrace{(t_{n+1} - t_n)\, \hat{\mathbf{v}}^{(n)}}_{\text{drift (learned)}} + \underbrace{(\boldsymbol{\sigma} \odot \boldsymbol{\epsilon}^{(n)}) \sqrt{(t_{n+1} - t_n)\, \frac{1 - t_{n+1}}{1 - t_n}}}_{\text{diffusion (Brownian bridge)}}, \tag{12}$$

where $\boldsymbol{\epsilon}^{(n)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The drift term uses the learned velocity $\hat{\mathbf{v}}^{(n)}$ in place of the true drift $(\mathbf{x}_1 - \mathbf{x}^{(n)})/(1 - t_n)$, while the diffusion term samples the zero-mean Brownian-bridge residual with its exact conditional variance between adjacent solver times (derivation in Appendix A.3). Dropping the diffusion term yields a deterministic ODE sampler, analogous to the probability-flow ODE in diffusion models [37].

In practice, at each step, the velocity is computed with classifier-free guidance (CFG) [12], which operates identically to its use in diffusion models:

$$\hat{\mathbf{v}}^{(n)} = (1 - \omega)\, f_\theta\big(\mathbf{x}^{(n)}, t_n, \hat{c}^{k,\emptyset}\big) + \omega\, f_\theta\big(\mathbf{x}^{(n)}, t_n, \hat{c}^k\big), \tag{13}$$

where $\hat{c}^k{=}(\hat{\mathbf{z}}^{<k}, y)$, $\hat{c}^{k,\emptyset}{=}(\hat{\mathbf{z}}^{<k}, \emptyset)$, and ω is the guidance weight. After the final step, we commit $\hat{\mathbf{z}}^k = \mathbf{x}^{(N)}$, then run one more forward pass on $\hat{\mathbf{z}}^k$ with timestep 0 to store the resulting keys and values in the KV-cache (see Appendix A.5). The inference procedure is given in Algorithm 2.

3.5 Implementation details

We implement BRAVO on top of the causal `Wan2.1-T2V-1.3B` transformer [40], where the text encoder and the video VAE are kept frozen. All training datasets and settings are the same as those used in stage 1 of [54]. The model is initialized from `Wan2.1-T2V-1.3B` by default, while it can also be initialized from a trained AR I2V model (here is the stage-1 checkpoint in [54]) to boost performance, which will be discussed in Section 4.1.1. We train BRAVO in the latent space with 21 latent frames of shape $16 \times 60 \times 104$, using one frame per AR block. The bridge uses $1k$ training timesteps with no timestep shift and a 16-dimensional channel-wise noise scale $\boldsymbol{\sigma}$, whose values are calculated by the rules in Appendix A.4. For optimization, we use AdamW [24] with learning rate 2×10^{-6}, betas $(0.0, 0.999)$, weight decay 0.01, and batch size 8. We train the model for $80k$ steps on 8 NVIDIA H200 GPUs, which takes about 6 days in total.

4 Experiments

In this section, we evaluate BRAVO's effectiveness in improving AR I2V generation. First, we conduct a standard benchmark VBench-I2V [17], including a basic comparison among the conventional noise-to-data transition, deterministic data-to-data flow matching [23], and our data-to-data Brownian bridge, as well as a wider comparison with other advanced I2V generation methods. Second, we propose a reference-based quality evaluation to examine next-frame prediction quality during rollout. Finally, we analyze inference hyperparameters, noise calibration, and loss weighting. Unless otherwise stated, in the following experiments, all models generate videos at the resolution of 480×832 and the frame count of 81, with the first frame and a text prompt given as conditions. For BRAVO, we set CFG weight $\omega = 2$ in Eq. (13) and sampling steps 50 during inference. We also use a bridge noise multiplier $\gamma = 1.2$ by default on $\boldsymbol{\sigma}$ in Eq. (12) rather than the same scale in Eq. (6) during training. We discuss the choice of ω and γ in Section 4.3. Experiments are conducted on a single H200 GPU.

4.1 Standard benchmark evaluation

In this section, we evaluate I2V generation on a widely used standard benchmark, namely VBench-I2V [17]. The benchmark provides a curated suite with multiple aspect-ratio crops and manually refined prompts, and evaluates videos on multiple dimensions of generation quality, including subject (I2V-Subj.) and background (I2V-Bg.) consistency with the input image, camera-motion alignment with the prompt (Cam.), subject (Subj.) and background (Bg.) consistency across frames, motion smoothness (Smooth), dynamic degree (Dyn.), aesthetic quality (Aes.), and imaging quality (Img.). Additionally, VBench-I2V also reports the aggregate image-video alignment score (I2V), video quality score (Quality), and overall score (Total). Higher values are better for all dimensions. Following the official protocol, we use the 9:16 cropping, which is close to our generation resolution of 480×832. Practically, we organize the evaluation into two parts. The first is a controlled transition-path comparison that isolates the effect of replacing noise-to-data generation with data-to-data bridge generation under the same model architecture and training settings. The second places our final BRAVO model in a broader context by comparing it with advanced I2V generation methods.

4.1.1 Noise-to-data versus data-to-data

Compared variants. We compare three variants initialized from `Wan-T2V-1.3B` [40]: The noise-to-data baseline starts each transition from Gaussian noise; The data-to-data flow-matching baseline starts from the previous latent frame but uses the deterministic $\boldsymbol{\sigma} = \mathbf{0}$ path; BRAVO* uses the same previous-frame source but with the Brownian perturbation described in Section 3.2. All models are trained on the same dataset under the same settings (as seen in Section 3.5). They receive the same image, text, and causal history, and use the same 50-step sampling budget.

Results. Table 1 reports the controlled comparison among these transition choices, where deterministic data-to-data flow matching performs substantially worse than the noise-to-data baseline. BRAVO* is evaluated under the same controlled setup (*e.g.*, initialization and training) as the baseline, outperforming the other variants. These results show that data-to-data generation is beneficial for AR I2V rollout when the previous-frame source is paired with Brownian bridge stochasticity. Albeit fair, this setting is not favorable to our method, because starting from a parallel noise-to-data diffusion model `Wan-T2V-1.3B` has to learn causal AR rollout and the data-to-data transition together in the

Table 1: Comparison among noise-to-data, deterministic data-to-data, and BRAVO. Bold numbers mark the best among the first three variants. BRAVO* uses the same training setting as the first two variants, while the gray row is the final BRAVO initialized from a trained noise-to-data AR baseline.

Method	Total	I2V	Quality	Cam.	I2V-Subj.	I2V-Bg.	Subj.	Bg.	Smooth	Dyn.	Aes.	Img.
Noise-to-data (baseline)	86.44	**94.84**	78.04	**34.60**	97.51	**98.98**	92.57	93.73	98.59	49.19	58.85	66.95
Data-to-data (flow matching)	76.53	92.74	60.32	30.54	95.29	97.95	68.41	82.39	98.59	4.07	38.56	55.97
Data-to-data (**BRAVO***)	**87.64**	**94.84**	**80.45**	28.05	**98.22**	98.85	**94.87**	**95.55**	**98.87**	**55.28**	**59.84**	**70.03**
Data-to-data (**BRAVO**)	87.86	95.43	80.29	33.94	98.42	99.16	92.71	96.00	98.90	58.13	59.35	70.05

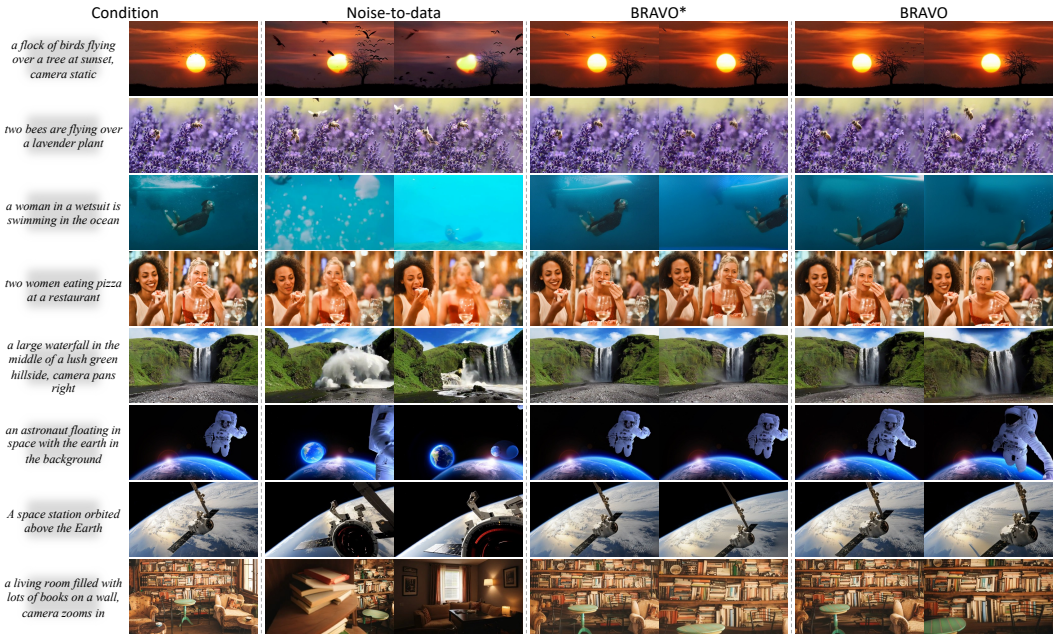

Figure 3: Qualitative visualization. Each column uses the same condition as input.

same training stage. Thus, we add the final BRAVO model in gray for reference here, which is initialized from a trained noise-to-data AR baseline instead. This initialization has already absorbed much of the gap between parallel diffusion and causal AR generation and further improves the performance. Figure 3 provides several qualitative results, where the noise-to-data baseline suffers from distortion, artifacts, and abrupt objects during rollout. In contrast, the data-to-data BRAVO preserves scene layout and object identity more consistently, alleviating these quality issues by grounding each transition on the previous generated frame.

4.1.2 Comparison with advanced I2V generation methods

Compared methods. We compare the final version of BRAVO with representative advanced I2V generation methods to place BRAVO in a broader benchmark context. We group the compared methods into diffusion models, including Step-Video-TI2V [15], Wan2.1-I2V, Wan2.2-I2V [40], and HunyuanVideo-I2V [19], and autoregressive models, including Cosmos [1], VideoMAR [50], Magi-1 [39], and Causal-Forcing [54] (we adopt the stage-1 checkpoint with 50 diffusion steps). Since autoregressive models are the closest generation paradigm to BRAVO, we will mainly focus on this group.

Results. Table 2 reports the comparison with advanced I2V generation methods. Compared with Causal-Forcing and VideoMAR, which have a similar model size and the same AR paradigm, BRAVO improves the performance on multiple dimensions and gets the best total score. Furthermore, BRAVO also remains competitive with much larger models, both diffusion and autoregressive. All these results demonstrate the effectiveness of BRAVO and its data-to-data design.

Table 2: Comparison with advanced I2V generation methods. In the autoregressive group, the best, second-best, and third-best are highlighted from darker to lighter purple.

Method	Size	Total	I2V	Quality	I2V-Subj.	I2V-Bg.	Cam.	Subj.	Bg.	Smooth	Dyn.	Aes.	Img.
Diffusion models													
HunyuanVideo-I2V [19]	13B	86.82	95.10	78.54	98.53	97.37	49.91	95.26	96.70	99.23	22.20	62.55	70.14
Step-Video-TI2V [15]	30B	88.37	95.51	81.22	97.86	98.63	49.32	96.02	97.06	99.24	48.78	62.29	70.44
Wan2.1-I2V-14B-720P [40]	14B	86.86	92.90	80.83	96.95	96.44	34.76	94.86	97.07	97.90	51.38	64.75	70.44
Wan2.2-I2V-A14B [40]	14B	88.16	95.82	80.50	97.52	99.03	54.44	94.68	96.72	98.32	52.85	61.53	70.36
Autoregressive models													
Cosmos [1]	13B	84.22	92.60	75.83	96.17	97.35	25.43	97.69	96.77	99.40	18.70	55.84	60.15
VideoMAR [50]	1.4B	84.82	94.02	75.61	97.85	98.38	21.62	97.13	97.20	99.57	10.98	55.81	62.34
Magi-1 [39]	24B	89.28	96.12	82.44	98.39	99.00	50.85	93.96	96.74	98.68	68.21	64.74	69.71
Causal-Forcing [54]	1.3B	87.08	94.67	79.49	97.62	99.03	29.10	94.03	95.20	98.70	50.41	59.77	69.32
BRAVO (Ours)	1.3B	87.86	95.43	80.29	98.42	99.16	33.94	92.71	96.00	98.90	58.13	59.35	70.05

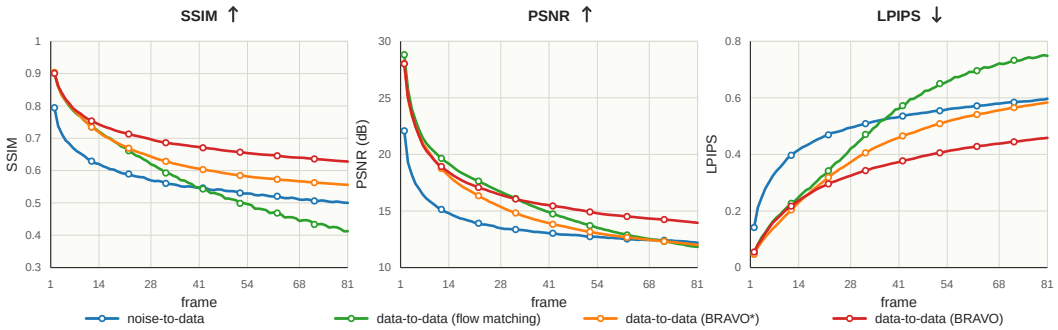

Figure 4: Reference-based quality curves on an OpenVid-1M subset [28]. Arrows indicate the preferred direction. BRAVO* uses the same controlled training setting as the baselines, while BRAVO denotes the final model initialized from a trained noise-to-data AR baseline.

4.2 Reference-based quality evaluation

Motivation and protocol. Standard I2V benchmarks mostly evaluate the final generated video, but they do not reveal how frame-wise prediction quality changes as an autoregressive model repeatedly feeds its own outputs back into the next transition. To this end, given a real video, we take its first frame and caption as the condition, generate the following frames, and plot frame-wise agreement curves against the original video. These curves are not meant to measure absolute perceptual quality, since many valid videos can follow the same condition. Instead, they provide a view that early frames depend on a short generated history, while later frames reveal accumulated autoregressive drift. We average the curves over a randomly sampled OpenVid-1M subset [28] using `SSIM` [44], `PSNR` [14], and `LPIPS` [51]. Higher `SSIM` and `PSNR` indicate better reference agreement, while lower `LPIPS` is better. Compared with a single aggregate score, these curves show the degradation pattern across the 81-frame rollout. We use the same variants as in Section 4.1.1.

Results. Figure 4 shows that BRAVO maintains better reference agreement throughout the AR rollout. The baseline behavior is consistent with Section 4.1.1, where deterministic data-to-data flow matching improves short-term agreement but degrades quickly, while adding Brownian bridge stabilizes the curves. The final BRAVO model remains the best across all three measures and shows the clearest advantage in late frames, where accumulated autoregressive drift is most visible. These curves again support the central motivation of BRAVO by showing that a stochastic data-to-data bridge from the previous frame provides a more reliable local transition than restarting each step from noise.

4.3 Analysis

Inference hyperparameters. We analyze two inference parameters, the CFG weight ω and the multiplier γ applied to the bridge noise scale $\boldsymbol{\sigma}$. ω controls text-conditioning strength, where a too-small value weakens prompt following and a too-large value can hurt image quality.

Table 3: VBench-I2V [17] Total score under different inference hyperparameters. The best is bold.

$\omega\backslash\gamma$	1.0	1.2	1.4
1.0	87.57	87.53	87.40
2.0	87.54	**87.64**	87.51
3.0	87.14	87.31	87.54

γ controls inference-time stochasticity along the data-to-data path. Since training uses clean sources but inference starts from the previous generated frame, we allow the inference bridge to carry slightly more uncertainty, following the spirit of [7]. A moderate γ helps avoid over-committing to an imperfect source frame, while excessive stochasticity can introduce drifting. Table 3 reports the VBench-I2V Total score, with the full results provided in Appendix B.2, and we choose $\omega = 2.0$ and $\gamma = 1.2$.

Dynamics and quality. In Table 4, we vary γ while keeping $\boldsymbol{\sigma}$ and ω fixed. Increasing γ from 1.0 to 1.4 lowers Dynamic Degree from 65.45 to 39.43, while Total remains between 87.51 and 87.64 and consistently exceeds the baseline's 86.44. As γ increases, subject and background consistency and motion smoothness improve. This suggests that smaller γ values favor more dynamic videos, while larger values favor temporal consistency. At $\gamma = 1.3$, BRAVO* closely matches the baseline in Dynamic Degree (48.78 vs. 49.19), while improving Quality from 78.04 to 80.29 and Total from 86.44 to 87.62. At $\gamma = 1.0$, its Dynamic Degree is higher than the baseline's (65.45 vs. 49.19), with higher Quality and Total scores as well. These results show that BRAVO* improves quality at both similar and higher Dynamic Degree.

Table 4: Sensitivity to the noise multiplier γ on VBench-I2V. The highest score is bold.

Method	γ	Dyn.	Total	I2V	Quality	I2V-Subj.	I2V-Bg.	Cam.	Subj.	Bg.	Smooth	Aes.	Img.
Noise-to-data (baseline)	–	49.19	86.44	94.84	78.04	97.51	98.98	**34.60**	92.57	93.73	98.59	58.85	66.95
Data-to-data (**BRAVO***)	1.0	**65.45**	87.54	94.48	80.59	97.89	98.52	28.83	94.09	94.57	98.73	58.83	69.47
	1.1	62.20	87.62	94.58	**80.67**	98.04	98.64	27.52	94.42	95.04	98.80	59.33	69.74
	1.2	55.28	**87.64**	94.84	80.45	98.22	98.85	28.05	94.87	95.55	98.87	59.84	70.03
	1.3	48.78	87.62	94.94	80.29	98.46	98.99	25.43	95.46	95.96	98.95	60.41	70.34
	1.4	39.43	87.51	**95.13**	79.89	**98.68**	**99.12**	25.03	**96.02**	**96.36**	**99.05**	**61.09**	**70.59**

Ablation on loss weighting. As derived in Appendix A.2, the stochastic variance of the bridge velocity target grows as t approaches 1, motivating the weight $w(t) = 1 - t$. We compare this weighting with an unweighted objective, $w(t) = 1$, using the same training settings for 40k steps. Table 5 shows that weighting improves I2V, Quality, and Total, with Total increasing from 85.95 to 86.82. The average global gradient norm also decreases from 0.238 to 0.088. These results support downweighting high-variance targets near the bridge endpoint to stabilize the training process and improve generation quality.

Table 5: Ablation of loss weighting for BRAVO* at 40k training steps, tested on VBench-I2V with the same inference settings. The highest score is bold.

Loss weight	Total	I2V	Quality	I2V-Subj.	I2V-Bg.	Cam.	Subj.	Bg.	Smooth	Dyn.	Aes.	Img.	Grad. norm
$w(t) = 1$	85.95	93.60	78.30	97.77	97.37	27.26	90.51	**94.41**	98.81	54.88	57.24	**67.89**	0.238
$w(t) = 1 - t$	**86.82**	**94.59**	**79.05**	**97.96**	**98.36**	**32.50**	**91.47**	93.79	**98.90**	**61.79**	**57.82**	67.38	0.088

5 Conclusion

We presented BRAVO, an autoregressive I2V generation framework that reconsiders how each next-frame prediction is initialized. Instead of treating every step as a new recovery problem from uninformative Gaussian noise, BRAVO uses the latest generated frame as the source endpoint of a Brownian bridge and learns the local transport toward the next frame under text and causal visual messages. This formulation turns autoregressive synthesis into a sequence of data-to-data updates, allowing the model to preserve the visual state in the rollout history while concentrating on the short-term change between adjacent frames. Together with causal teacher-forced training, weighted bridge velocity learning, and cached autoregressive inference, BRAVO improves generation quality and rollout stability over the conventional noise-to-data baseline and remains competitive with larger video generation models.

Acknowledgements

This project is supported by the Ministry of Education, Singapore, under the Academic Research Fund Tier 1 (FY2026, WBS: A-8004345-00-00), the National Research Foundation, Singapore, and Cyber Security Agency of Singapore under its National Cybersecurity R&D Programme and CyberSG R&D Cyber Research Programme Office (Award: CRPO-GC1-NTU-002), as well as NUS IT's High Performance Cluster.

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[2] Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209): 1–80, 2025.

[3] Nuha Aldausari, Arcot Sowmya, Nadine Marcus, and Gelareh Mohammadi. Video generative adversarial networks: a review. *ACM Computing Surveys*, 55(2):1–25, 2022.

[4] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 28, pages 1–9, 2015.

[5] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[6] Clément Chadebec, Onur Tasar, Sanjeev Sreetharan, and Benjamin Aubin. Lbm: Latent bridge matching for fast image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 29086–29098, 2025.

[7] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 37, pages 24081–24125, 2024.

[8] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7310–7320, 2024.

[9] Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. Dmax: Aggressive parallel decoding for dllms. In *Proceedings of the Neural Information Processing Systems*, 2026.

[10] Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. dparallel: Learnable parallel decoding for dllms. In *Proceedings of the International Conference on Learning Representations*, volume 2026, pages 53193–53211, 2026.

[11] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *Proceedings of the International Conference on Learning Representations*, pages 1–19, 2024.

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *Proceedings of the Advances in Neural Information Processing Systems Workshop*, pages 1–8, 2021.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[14] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *Proceedings of the IEEE International Conference on Pattern Recognition*, pages 2366–2369, 2010.

[15] Haoyang Huang, Guoqing Ma, Nan Duan, Xing Chen, Changyi Wan, Ranchen Ming, Tianyu Wang, Bo Wang, Zhiying Lu, Aojie Li, et al. Step-video-ti2v technical report: A state-of-the-art text-driven image-to-video generation model. *arXiv preprint arXiv:2503.11251*, 2025.

[16] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 1–19, 2025.

[17] Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):3268–3285, 2026.

[18] Peter E Kloeden and RA Pearson. The numerical solution of stochastic differential equations. *The ANZIAM Journal*, 20(1):8–12, 1977.

[19] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[20] Bo Li, Kaitao Xue, Bin Liu, and Yu-Kun Lai. Bbdm: Image-to-image translation with brownian bridge diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1952–1961, 2023.

[21] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *Proceedings of the International Conference on Learning Representations*, pages 1–28, 2023.

[22] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I^2SB: Image-to-image schrödinger bridge. In *Proceedings of the International Conference on Machine Learning*, pages 22042–22062, 2023.

[23] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *Proceedings of the International Conference on Learning Representations*, pages 1–33, 2023.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*, pages 1–18, 2019.

[25] Haiquan Lu, Gongfan Fang, Xinyin Ma, and Xinchao Wang. Vgedit: Unlocking video generation priors for reasoning-informed image editing. In *Proceedings of the European Conference on Computer Vision*, pages 372–389, 2026.

[26] Zonglin Lyu, Ming Li, Jianbo Jiao, and Chen Chen. Frame interpolation with consecutive brownian bridge diffusion. In *Proceedings of the ACM International Conference on Multimedia*, pages 3449–3458, 2024.

[27] Xinyin Ma, Julius Berner, Chao Liu, Arash Vahdat, Weili Nie, and Xinchao Wang. Flex-forcing: Towards a unified autoregressive and bidirectional video diffusion model. volume 306, pages 84076–84091, 2026.

[28] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *Proceedings of the International Conference on Learning Representations*, pages 1–20, 2025.

[29] Haomiao Ni, Changhao Shi, Kai Li, Sharon X Huang, and Martin Renqiang Min. Conditional image-to-video generation with latent flow diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 18444–18455, 2023.

[30] OpenAI. GPT-5.5. `https://developers.openai.com/api/docs/models/gpt-5.5`, 2026. Large language model.

[31] Zhibin Qin, Zhenxiong Tan, Zeqing Wang, Songhua Liu, and Xinchao Wang. Spotedit: Selective region editing in diffusion transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 18683–18692, 2026.

[32] Jiangwei Ren, Xingyu Jiang, Zizhuo Li, Dingkang Liang, Xin Zhou, and Xiang Bai. Minima: Modality invariant image matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 23059–23068, 2025.

[33] Weiming Ren, Huan Yang, Ge Zhang, Cong Wei, Xinrun Du, Wenhao Huang, and Wenhu Chen. Consisti2v: Enhancing visual consistency for image-to-video generation. *Transactions on Machine Learning Research*, pages 1–28, 2024.

[34] Qiuhong Shen, Xingyi Yang, Xinyin Ma, Gongfan Fang, and Xinchao Wang. SPEED: Sharpened-Teacher Distillation for Parallel Decoding of Diffusion Language Models. In *Proceedings of the International Conference on Machine Learning*, 2026.

[35] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 36, pages 62183–62223, 2023.

[36] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of the International Conference on Learning Representations*, pages 1–20, 2021.

[37] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations*, pages 1–36, 2021.

[38] Zhenxiong Tan, Zeqing Wang, Xingyi Yang, Songhua Liu, and Xinchao Wang. Vision bridge transformer at scale. In *Proceedings of the European Conference on Computer Vision*, pages 503–521, 2026.

[39] Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.

[40] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[41] Yimu Wang, Xuye Liu, Wei Pang, Li Ma, Shuai Yuan, Paul Debevec, and Ning Yu. Survey of video diffusion models: Foundations, implementations, and applications. *Transactions on Machine Learning Research*, pages 1–93, 2025.

[42] Yuji Wang, Zehua Chen, Xiaoyu Chen, Yixiang Wei, Jun Zhu, and Jianfei Chen. Framebridge: Improving image-to-video generation with bridge models. In *Proceedings of the International Conference on Machine Learning*, pages 1–31, 2025.

[43] Zeqing Wang, Bowen Zheng, Xingyi Yang, Zhenxiong Tan, Yuecong Xu, and Xinchao Wang. Minute-long videos with dual parallelisms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 10358–10366, 2026.

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

[45] Zeyu Xiao and Xinchao Wang. Exploiting blurry representations for event-guided video super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 11032–11041, 2026.

[46] Zeyu Xiao and Xinchao Wang. Event-guided online video super-resolution. *IEEE Transactions on Image Processing*, 2026.

[47] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *Proceedings of the European Conference on Computer Vision*, pages 399–417, 2024.

[48] Zhen Xing, Qijun Feng, Haoran Chen, Qi Dai, Han Hu, Hang Xu, Zuxuan Wu, and Yu-Gang Jiang. A survey on video diffusion models. *ACM Computing Surveys*, 57(2):1–42, 2024.

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

[50] Hu Yu, Biao Gong, Hangjie Yuan, DanDan Zheng, Weilong Chai, Jingdong Chen, Kecheng Zheng, and Feng Zhao. Videomar: Autoregressive video generation with continuous tokens. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 1–23, 2025.

[51] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[52] Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023.

[53] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *Proceedings of the International Conference on Learning Representations*, pages 1–12, 2024.

[54] Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. In *Proceedings of the International Conference on Machine Learning*, pages 1–19, 2026.

A Technical Derivations and Details

A.1 Teacher-forced causal mask

Let i and j index visual tokens in the concatenated sequence in Eq. (9). In transformer block ℓ, let $Z^{(\ell-1)}$ be the input hidden states and let $Q^{(\ell)}$, $K^{(\ell)}$, and $V^{(\ell)}$ be the corresponding query, key, and value projections. With attention logits $S^{(\ell)} = Q^{(\ell)}(K^{(\ell)})^\top/\sqrt{d_h}$, the blockwise mask in Eq. (10) is applied before the row-wise softmax:

$$A_{ij}^{(\ell)} = \frac{\exp(S_{ij}^{(\ell)})\, M_{ij}}{\sum_{j'} \exp(S_{ij'}^{(\ell)})\, M_{ij'}}, \quad \bar{Z}^{(\ell)} = A^{(\ell)} V^{(\ell)}. \tag{14}$$

Equivalently, entries with $M_{ij} = 0$ are removed from the normalization for query token i. In this way, clean tokens attend only to clean tokens from the current or earlier frames, so the clean stream remains causal. A noisy token for frame k attends to clean tokens from frames $< k$, which provides the ground-truth history $\mathbf{z}^{<k}$, and to noisy tokens from frame k, which allows computation within the current bridge state. It cannot attend to the clean target $\mathbf{z}^k$, later clean frames, or noisy tokens from other frames. This preserves the teacher-forced history while preventing a direct path from the clean target $\mathbf{z}^k$ to its own velocity prediction.

A.2 Derivation of the loss weight

We derive the loss weight in Eq. (11) from the stochastic scale of the bridge velocity target. Substituting the marginal in Eq. (6) into Eq. (7) gives:

$$\mathbf{v}_t^\star = \frac{\mathbf{x}_1 - \mathbf{x}_t}{1-t} = \mathbf{x}_1 - \mathbf{x}_0 - (\boldsymbol{\sigma} \odot \boldsymbol{\epsilon})\sqrt{\frac{t}{1-t}}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{15}$$

Conditioned on the endpoints $\mathbf{x}_0$ and $\mathbf{x}_1$, the residual $\mathbf{x}_1 - \mathbf{x}_0$ is fixed. Thus only the Brownian term contributes to the conditional covariance given the two bridge endpoints:

$$\mathrm{Cov}(\mathbf{v}_t^\star \mid \mathbf{x}_0, \mathbf{x}_1) = \mathrm{Diag}(\boldsymbol{\sigma}^2)\frac{t}{1-t}. \tag{16}$$

Hence the stochastic variance of the velocity target grows as $\mathcal{O}(1/(1-t))$ as $t \to 1$. In an unweighted squared loss, samples close to the target endpoint would therefore contribute disproportionately to the objective, even though the bridge state itself is already close to $\mathbf{x}_1$.

We use the weight $w(t) = 1 - t$ to remove this endpoint singularity. Multiplying the squared error by $w(t)$ scales the covariance term as:

$$w(t)\,\mathrm{Cov}(\mathbf{v}_t^\star \mid \mathbf{x}_0, \mathbf{x}_1) = \mathrm{Diag}(\boldsymbol{\sigma}^2)\, t. \tag{17}$$

The factor $1/(1-t)$ is thus cancelled, leaving a bounded stochastic contribution for all bridge times.

A.3 Derivation of the Brownian perturbation term

We write the stochastic part of the bridge in Eq. (6) as $\mathbf{x}_t = \bar{\mathbf{x}}_t + \boldsymbol{\sigma} \odot \boldsymbol{\beta}_t$, where $\bar{\mathbf{x}}_t = (1-t)\mathbf{x}_0 + t\,\mathbf{x}_1$ is the deterministic mean. Here $\{\boldsymbol{\beta}_t\}_{t\in[0,1]}$ is the zero-endpoint Brownian bridge induced by a standard vector Brownian motion $\mathbf{W}_t$:

$$\boldsymbol{\beta}_t = \mathbf{W}_t - t\,\mathbf{W}_1, \quad \mathbf{W}_0 = \mathbf{0}, \quad \mathbf{W}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{18}$$

Moreover, for the standard vector Brownian motion $\mathbf{W}_t$, when $t \le t'$ we have $\mathbf{W}_{t'} = \mathbf{W}_t + (\mathbf{W}_{t'} - \mathbf{W}_t)$ with an independent increment, so $\mathrm{Cov}(\mathbf{W}_t, \mathbf{W}_{t'}) = t\mathbf{I}$. By symmetry, $\mathrm{Cov}(\mathbf{W}_t, \mathbf{W}_{t'}) = \min(t, t')\mathbf{I}$, and therefore:

$$\begin{aligned} \mathrm{Cov}(\boldsymbol{\beta}_t, \boldsymbol{\beta}_{t'}) &= \mathrm{Cov}(\mathbf{W}_t - t\mathbf{W}_1, \mathbf{W}_{t'} - t'\mathbf{W}_1) \\ &= \big(\min(t,t') - t'\,t - t\,t' + t\,t'\big)\mathbf{I} \\ &= (\min(t,t') - tt')\mathbf{I}. \end{aligned} \tag{19}$$

For two adjacent timestamps, let $a = t_n$ and $b = t_{n+1}$ with $0 < a < b < 1$. The pair $(\boldsymbol{\beta}_a, \boldsymbol{\beta}_b)$ is jointly Gaussian, and the covariance in Eq. (19) gives:

$$\begin{aligned} \mathrm{Var}(\boldsymbol{\beta}_a) &= a(1-a)\mathbf{I}, \\ \mathrm{Var}(\boldsymbol{\beta}_b) &= b(1-b)\mathbf{I}, \\ \mathrm{Cov}(\boldsymbol{\beta}_b, \boldsymbol{\beta}_a) &= a(1-b)\mathbf{I}. \end{aligned} \tag{20}$$

For jointly Gaussian variables Y and X, we have:

$$Y \mid X \sim \mathcal{N}\left(\Sigma_{YX}\Sigma_{XX}^{-1}X, \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}\right). \tag{21}$$

Applying this formula with $Y = \boldsymbol{\beta}_b$ and $X = \boldsymbol{\beta}_a$ yields:

$$\boldsymbol{\beta}_b \mid \boldsymbol{\beta}_a \sim \mathcal{N}\left(\frac{1-b}{1-a}\boldsymbol{\beta}_a, \frac{(1-b)(b-a)}{1-a}\mathbf{I}\right). \tag{22}$$

Substituting $a = t_n$ and $b = t_{n+1}$ gives:

$$\boldsymbol{\beta}_{t_{n+1}} \mid \boldsymbol{\beta}_{t_n} \sim \mathcal{N}\left(\frac{1-t_{n+1}}{1-t_n}\boldsymbol{\beta}_{t_n}, \frac{(1-t_{n+1})(t_{n+1}-t_n)}{1-t_n}\mathbf{I}\right). \tag{23}$$

The mean term in this conditional distribution is already accounted for by the drift update as in Eq. (12). Specifically, using $\mathbf{x}_{t_n} = \bar{\mathbf{x}}_{t_n} + \boldsymbol{\sigma} \odot \boldsymbol{\beta}_{t_n}$, the conditional mean of the full bridge state is:

$$\mathbb{E}\left[\mathbf{x}_{t_{n+1}} \mid \mathbf{x}_{t_n}, \mathbf{x}_0, \mathbf{x}_1\right] = \mathbf{x}_{t_n} + (t_{n+1} - t_n)\frac{\mathbf{x}_1 - \mathbf{x}_{t_n}}{1-t_n}. \tag{24}$$

This is the drift step in Eq. (12) when the true velocity is used. In inference, the learned velocity $\hat{\mathbf{v}}^{(n)}$ replaces this true drift. Therefore the term added explicitly is only the zero-mean residual innovation, whose covariance is $\mathrm{Diag}(\boldsymbol{\sigma}^2)(1-t_{n+1})(t_{n+1}-t_n)/(1-t_n)$:

$$\mathbf{x}^{(n+1)} \leftarrow \mathbf{x}^{(n+1)} + (\boldsymbol{\sigma} \odot \boldsymbol{\epsilon}^{(n)})\sqrt{\frac{(1-t_{n+1})(t_{n+1}-t_n)}{1-t_n}}, \quad \boldsymbol{\epsilon}^{(n)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{25}$$

which matches the perturbation term used in the main text.

A.4 Channel-wise noise scale calibration

The bridge SDE uses a Brownian noise scale $\boldsymbol{\sigma} \in \mathbb{R}^C$ in Eq. (5) and Eq. (6). We calibrate this scale in the Wan latent space, where $C = 16$ [40]. Since the latent channels have different value ranges, using a single scalar noise level would be poorly balanced. It would add too much noise to channels that change little between adjacent frames, and too little noise to channels with larger natural changes. For this reason, we use a channel-wise vector $\boldsymbol{\sigma}$.

The goal is to make the Brownian perturbation comparable to the typical one-step motion in each channel. For an adjacent latent-frame pair, let $\boldsymbol{\delta} = \mathbf{x}_1 - \mathbf{x}_0$ denote the displacement from the source frame to the target frame. For channel c, we estimate its motion scale by:

$$s_c^2 \triangleq \mathbb{E}\left[(x_{1,c} - x_{0,c})^2\right], \tag{26}$$

where the expectation is taken over adjacent frame pairs and spatial locations. Thus s_c is the root mean square size of the local change that the bridge needs to model in channel c. We then choose σ_c so that the noise has the same relative strength across channels. Using the displacement energy as the signal and the Brownian variance as the noise, we define:

$$\mathrm{SNR}_c \triangleq \frac{\mathbb{E}\left[\delta_c^2\right]}{\sigma_c^2} = \frac{s_c^2}{\sigma_c^2}, \tag{27}$$

and set this ratio to a shared target ρ. This gives:

$$\frac{s_c^2}{\sigma_c^2} = \rho \quad \Longrightarrow \quad \sigma_c = \frac{s_c}{\sqrt{\rho}}. \tag{28}$$

In our experiments, we set $\rho = 1$, so $\sigma_c = s_c$. We estimate s_c once from all adjacent latent-frame differences $(\mathbf{z}^{k+1} - \mathbf{z}^k)$ and all spatial locations in the training set. This produces the fixed channel-wise scale:

$$\begin{aligned} \boldsymbol{\sigma} = (&0.195654, 0.346082, 0.272466, 0.277743, 0.405165, 0.316656, 0.211535, 0.306381, \\ &0.192703, 0.286887, 0.191546, 0.318505, 0.315418, 0.414367, 0.233660, 0.314419). \end{aligned} \tag{29}$$

During training, we keep this vector fixed. During inference, we multiply it by γ as discussed in Section 4.3. The vector $\boldsymbol{\sigma}$ fixes the relative noise level across channels, while γ adjusts the overall amount of stochasticity.

A.5 Inference-time KV-cache implementation

At inference time, the clean stream used during teacher forcing is replaced by a cache of committed frames. Before generating frame k, the cache contains keys and values for $\hat{\mathbf{z}}^{1:k-1}$. Each committed latent is forwarded once at bridge time 0 through the same transformer call used in Eq. (13), as in Algorithm 2. Let $Z_s^{(\ell)}$ denote the hidden states of cached frame s at transformer block ℓ. We store

$$K_s^{(\ell)} = W_K^{(\ell)} Z_s^{(\ell)}, \quad V_s^{(\ell)} = W_V^{(\ell)} Z_s^{(\ell)}, \tag{30}$$

and stack them as $K_{1:k-1}^{(\ell)} = [K_1^{(\ell)}, \dots, K_{k-1}^{(\ell)}]$ and $V_{1:k-1}^{(\ell)}$.

At solver step n, the current bridge state $\mathbf{x}^{(n)}$ produces current-frame projections $Q_k^{(\ell,n)}$, $K_k^{(\ell,n)}$, and $V_k^{(\ell,n)}$. Its queries attend to the cached history and to the current-frame tokens:

$$\bar{Z}_k^{(\ell,n)} = \mathrm{SelfAttn}^{(\ell)}\Big(Q_k^{(\ell,n)},\, [K_{1:k-1}^{(\ell)}, K_k^{(\ell,n)}],\, [V_{1:k-1}^{(\ell)}, V_k^{(\ell,n)}]\Big). \tag{31}$$

This aligns with the training-time mask in Eq. (10): the current bridge state can use cached committed frames and its own tokens, but not future frames. The current-frame keys and values are recomputed at each solver step. After the final step, we commit $\hat{\mathbf{z}}^k = \mathbf{x}^{(N)}$, forward it once at bridge time 0, and append the resulting keys and values:

$$K_{1:k}^{(\ell)} = [K_{1:k-1}^{(\ell)}, K_k^{(\ell)}], \quad V_{1:k}^{(\ell)} = [V_{1:k-1}^{(\ell)}, V_k^{(\ell)}]. \tag{32}$$

The updated cache is then used when generating frame $k{+}1$, avoiding recomputation of the previous frames.

B Additional experimental results

B.1 Additional visualizations

Figures 5 and 6 provide additional qualitative comparisons under the same I2V conditions. For each method, we show four snapshots sampled every 20 frames from the generated 81-frame video, corresponding to frames 21, 41, 61, and 81. The first frame is given as the known image condition shown in the first column together with the text prompt. Across these examples, BRAVO shows fewer artifacts in later snapshots and better preserves the main subject than the noise-to-data baseline.

B.2 Full results for inference hyperparameter analysis

Table 6 here reports the full VBench-I2V [17] metrics for the inference-time hyperparameter analysis summarized in the main paper.

B.3 Limitations and future work

BRAVO improves AR I2V generation by replacing noise-to-data recovery with a Brownian bridge between adjacent frames, but its current sampler still uses 50 bridge steps for each generated frame. This leads to a longer generation time than fully distilled autoregressive video models. An important

Table 6: Full inference hyperparameter analysis for BRAVO* on VBench-I2V.

ω	γ	Total	I2V	Quality	I2V-Subj.	I2V-Bg.	Cam.	Subj.	Bg.	Smooth	Dyn.	Aes.	Img.
1.0	1.0	87.57	94.71	80.43	98.18	98.81	26.34	95.42	95.44	98.86	54.07	59.96	69.94
1.0	1.2	87.53	95.07	79.98	98.71	99.07	24.25	96.47	96.33	99.00	39.43	61.23	70.61
1.0	1.4	87.40	95.34	79.45	99.10	99.31	22.02	97.49	97.04	99.16	25.20	62.36	71.00
2.0	1.0	87.54	94.48	80.59	97.89	98.52	28.83	94.09	94.57	98.73	65.45	58.83	69.47
2.0	1.2	**87.64**	94.84	80.45	98.22	98.85	28.05	94.87	95.55	98.87	55.28	59.84	70.03
2.0	1.4	87.51	95.13	79.89	98.68	99.12	25.03	96.02	96.36	99.05	39.43	61.09	70.59
3.0	1.0	87.14	94.24	80.05	97.62	98.28	30.14	92.59	93.74	98.65	69.92	57.48	68.78
3.0	1.2	87.31	94.64	79.98	98.06	98.59	29.23	93.41	94.63	98.76	60.98	58.39	69.41
3.0	1.4	87.54	94.97	80.11	98.44	98.98	26.47	94.65	95.60	98.93	51.63	59.72	70.13

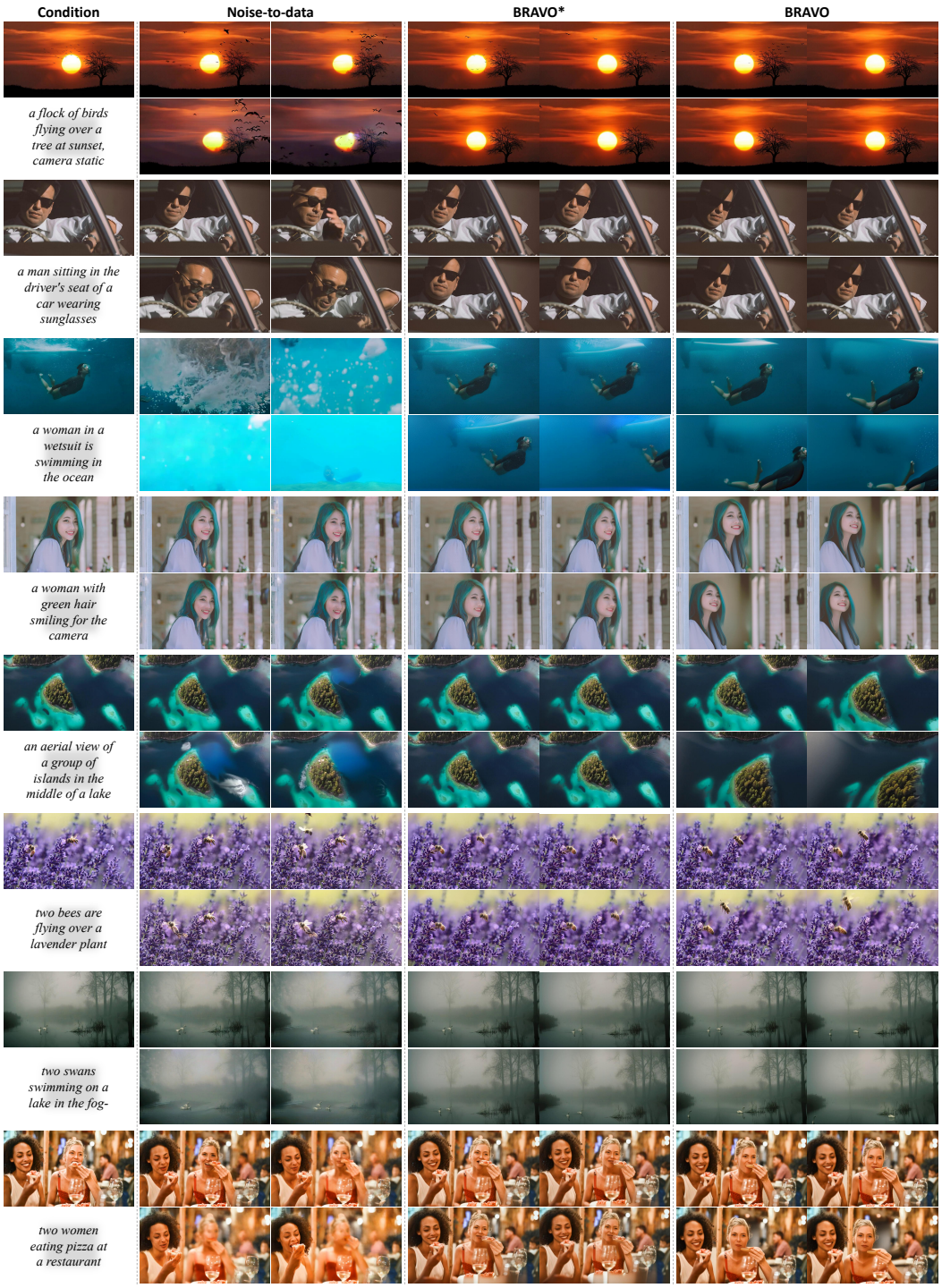

Figure 5: More qualitative visualization. Each column uses the same condition as input. Although some cases are shown in the main paper, here we give different snapshots of the generated videos, which are sampled every 20 frames from the 81-frame video.

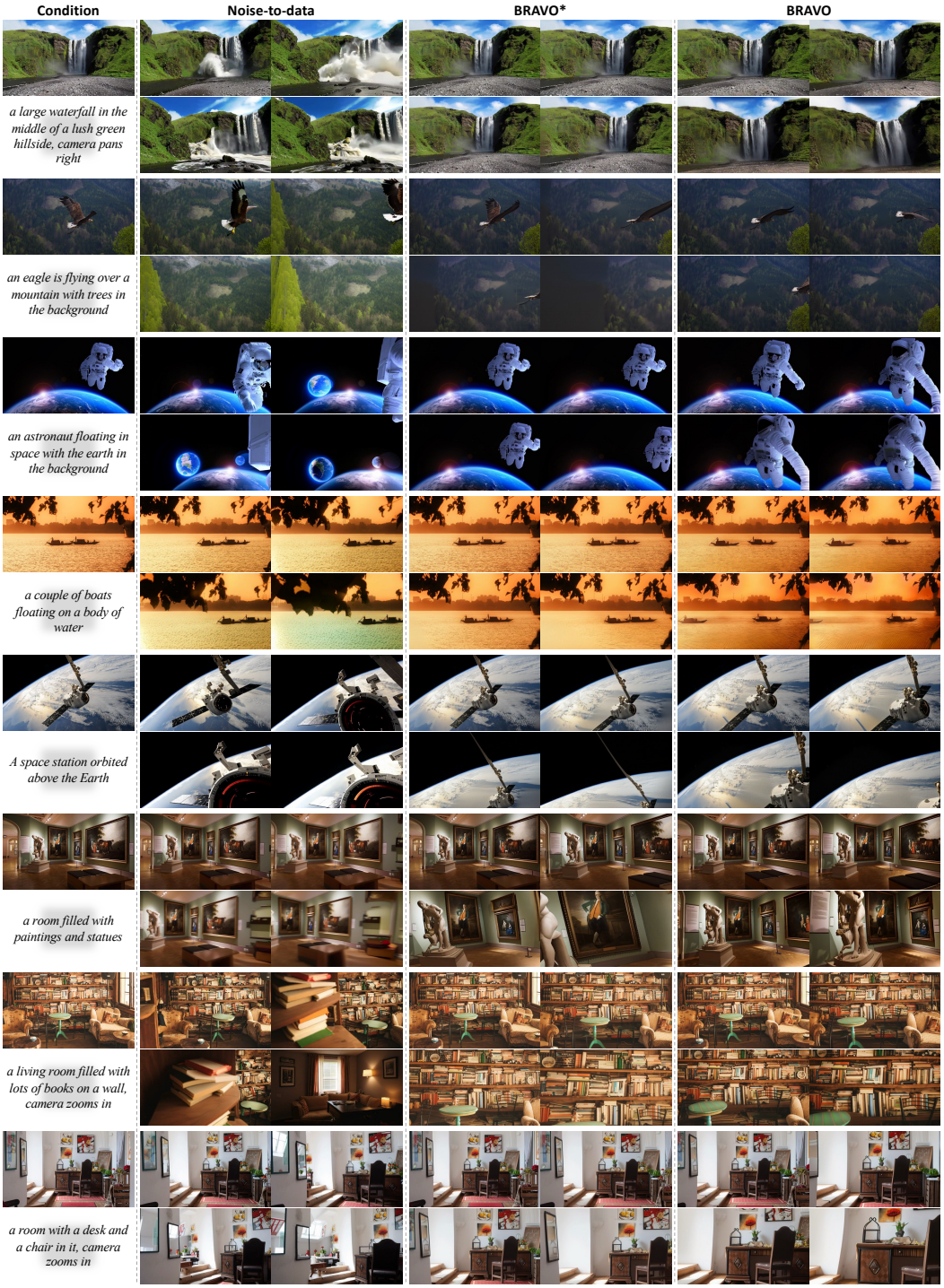

Figure 6: More qualitative visualization. Each column uses the same condition as input. Although some cases are shown in the main paper, here we give different snapshots of the generated videos, which are sampled every 20 frames from the 81-frame video.

direction is therefore to distill the bridge sampler into fewer steps, for example, following the multi-stage distillation strategy used by Causal-Forcing [54]. Another limitation is the model scale. Our experiments mainly use a relatively small Wan-based (1.3B) model, which can struggle with complex prompts, compositional instructions, or fine-grained text. Future work should validate BRAVO on larger video backbones to test whether the bridge transition brings similar gains when the base model has stronger modeling capacity.

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 abstract and introduction state the main motivation, BRAVO's Brownian bridge paradigm, and the experimental claims. The scope is supported by the related-work discussion, method description, VBench-I2V results, reference-based rollout curves, and the limitations presented in the appendix.

   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 appendix includes a limitations and future work subsection.

   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 present formal theorems. It only provides the derivations needed for the method.

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 method section gives the model formulation, training objective, inference procedure, and algorithms, while the implementation details and experiments specify the backbone, optimizer, training length, sampling steps, evaluation protocol, and hyperparameters.

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: The current submission does not include open access code. We provide method, implementation, and evaluation details in the main text and appendix, and plan to release the code upon acceptance.

   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 experimental section describes the VBench-I2V protocol, compared variants and methods, rollout evaluation, metrics, generation resolution and frame count, inference hyperparameters, and the hyperparameter study. The implementation details subsection specifies the optimizer, learning rate, batch size, training steps, backbone, latent shape, and compute setup.

   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 paper reports benchmark scores and averaged reference-based curves, but it does not report confidence intervals, standard deviations, or statistical significance tests. The experiments are computationally expensive video generation runs, so we leave this as a limitation of the current evaluation.

   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: [Yes].

   Justification: The experimental section states that the reported experiments are conducted on a single NVIDIA H200 GPU and specifies the inference sampling budget of 50 steps per generated frame. The implementation details also report the training compute used for BRAVO.

   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 work uses existing video generation datasets, benchmarks, and models with citations, and does not involve human-subject experiments or crowdsourcing. We are not aware of any aspect of the research that violates 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: [No].

    Justification: The paper focuses on the technical method and does not include a dedicated broader-impact discussion.

    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: [N/A].

    Justification: The current submission does not describe releasing a new dataset or model checkpoint.

    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: [No].

Justification: Existing assets such as Wan, VBench-I2V, OpenVid-1M, and compared baselines are credited through citations. However, the paper does not currently enumerate the licenses or terms of use for each asset.

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: [N/A].

Justification: The paper introduces a method and trained experimental models, but it does not release a new dataset, benchmark, codebase, or model asset as part of the current submission.

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 experiments or 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 experiments or 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: [N/A].

Justification: LLMs are not used as an important, original, or non-standard component of the core method. Any use of generative tools for writing, editing, formatting, or preparing illustrative material is outside the 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.