# WHERE COMPUTE MATTERS: HETEROGENEOUS ATTENTION FOR EFFICIENT VIDEO DIFFUSION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Efficient video generation requires reducing the quadratic cost of self-attention over long spatio-temporal token sequences. Existing efficient-attention methods typically apply the same computation pattern to every token, even though denoising difficulty varies substantially across video regions and evolves throughout the generation process. We introduce HetA-DiT, a heterogeneous attention mechanism that adaptively allocates computation according to token difficulty. A lightweight uncertainty branch predicts a token-wise estimate of denoising difficulty, which is used to route uncertain tokens through dense global attention while processing more reliable tokens with efficient local attention. The resulting routing is content- and timestep-adaptive, retains global context where it matters most, and provides a single parameter for controlling the quality-efficiency tradeoff. HetA-DiT is compatible with few-step distribution-matching distillation and introduces no additional Transformer evaluation at inference time by reusing uncertainty estimates from the preceding denoising step. We evaluate the method on DMD-distilled Wan2.2-5B and Wan2.1-1.3B models. Across VBench, VBench-2.0, and human preference evaluation, HetA-DiT maintains competitive generation quality while routing only approximately 20% of tokens through dense attention.

## 1 INTRODUCTION

Diffusion Transformers (DiTs) have become the dominant backbone for high-quality video generation, powering recent open systems such as Wan (Wan et al., 2025), HunyuanVideo (Hunyuan Foundation Model Team, 2024), CogVideoX (Yang et al., 2025b), LTX-Video (HaCohen et al., 2024), and Open-Sora Plan (Lin et al., 2024). By representing a video as a long sequence of latent spatio-temporal tokens, DiTs flexibly model appearance, motion, and temporal consistency in a single architecture. This flexibility, however, comes at a steep computational price: self-attention scales quadratically with the number of tokens, and video sequences are typically an order of magnitude longer than image sequences. As a result, a single denoising step is dominated by attention cost. The community has responded with a growing set of efficiency techniques, but essentially all of them make attention cheaper *uniformly*, treating every token as equally important to compute.

A key observation behind our work is that this uniformity is at odds with the structure of video content: *not all spatio-temporal tokens are equally hard to denoise*. Large regions such as static backgrounds, sky, or smoothly varying textures can be predicted reliably from local context very early in denoising. Other regions—moving objects, occlusions, object boundaries, faces, hands, and fine semantic details—remain ambiguous much longer and genuinely benefit from richer global context. Equally important, *which* tokens are hard is not fixed: it changes across denoising steps, across prompts, and across frames as motion and occlusion evolve. Any static sparsity pattern is therefore fundamentally mismatched to the underlying problem. Figure 1 illustrates this behavior.

Beyond making attention uniformly cheaper, a much less explored direction is to allocate compute *non-uniformly* across the tokens themselves. The few existing efforts in this direction leave a clear gap: token caching (Zou et al., 2024; Shmilovich et al., 2025) makes hard binary cache-or-compute decisions and relies on many-step redundancy that vanishes under step distillation; MoE DiTs (Fei et al., 2024; Wan et al., 2025) route only the FFN sub-layer, leaving the attention bottleneck untouched; and training-free adaptive sparsification(Peruzzo et al., 2025; Liu et al., 2026; Zhang et al.,



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

2025a) avoids tuning at the cost of measurable quality loss. We occupy the design point none of these cover: *adaptive, attention-level, DMD-compatible* token routing, obtained at a small tuning cost (appr. 12 GPU-days, four to five orders of magnitude below pretraining).

We propose a **heterogeneous attention** mechanism that performs non-uniform token processing. We augment the denoiser with a lightweight uncertainty prediction branch that produces a per-token scalar, trained with an uncertainty-aware reconstruction loss so that high values mark tokens whose clean-latent prediction is unreliable. After training, the branch is frozen and used only to produce a stable routing signal. At inference, an adaptive per-sample, per-timestep threshold sends high-uncertainty tokens to dense self-attention and reliable tokens to local self-attention, with a single hyperparameter trading efficiency for quality. Three properties distinguish this design: (i) routing is *content- and timestep-adaptive*, so dense compute follows difficult regions across frames and denoising steps; (ii) the mechanism is *drop-in* on transformer-based video diffusion backbones; and (iii) it is explicitly *compatible with step distillation* and is demonstrated on DMD-distilled Wan models.

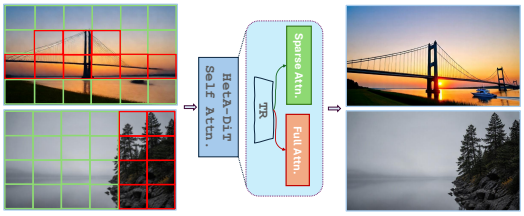


Figure 1: **Spend compute where it counts.** Not all video tokens are equally hard to denoise; fine structures demand global context, while smooth regions are handled by cheap local attention. Our heterogeneous attention method incorporates this via an adaptive uncertainty-based token router (TR) that dispatches each token to local or dense attention.

Applied to DMD-distilled Wan2.2-5B and Wan2.1-1.3B (Wan et al., 2025) at 704p and 480p resolution respectively, our routing keeps only $\sim$20% of tokens on dense attention while matching the base models on VBench (Huang et al., 2024) and VBench-2.0 (Zheng et al., 2025), and strictly outperforms random and fixed-pattern routing at matched sparsity.

Our main contributions are the following:

- We identify a gap in efficient video generation: existing efficiency methods either allocate a uniform compute budget to every spatio-temporal token regardless of its denoising difficulty, make hard binary compute-or-cache decisions, and/or are incompatible with the low-step, step-distilled regime that dominates modern deployment.
- We propose a heterogeneous attention mechanism that fills this gap by allocating compute per token. A lightweight branch predicts a per-token uncertainty score; an adaptive per-sample, per-timestep threshold then routes reliable tokens to cheap local attention and uncertain tokens to dense global attention. The mechanism exposes a single quality/efficiency knob and remains effective under step distillation.
- We validate our method on two popular open-source video diffusion models (Wan2.2-5B and Wan2.1-1.3B) across VBench, VBench-2.0, and a human preference study, and use ablations to isolate the contribution of each component.

## 2 Related Work

**Efficient Video Diffusion.** Open-source video diffusion models such as Wan (Wan et al., 2025), HunyuanVideo (Hunyuan Foundation Model Team, 2024), CogVideoX (Yang et al., 2025b), LTX-Video (HaCohen et al., 2024), and Open-Sora Plan (Lin et al., 2024) have driven rapid progress in text-to-video generation, but their DiT backbones incur a large computational cost that grows quadratically with the number of spatio-temporal tokens. A first line of efficiency work reduces the *number* of denoising steps via step distillation: DMD (Yin et al., 2024b;a) and follow-ups (Lin et al., 2025; Fan et al., 2026; Nie et al., 2026; Zheng et al., 2026) now yield $>30\times$ speedups and have become the de facto deployment regime. A parallel line targets deployment on edge and mobile hardware through step distillation, sparse/hybrid attention, model pruning and quantization, e.g., Neodragon (Karnewar et al., 2026), SnapGen-V (Wu et al., 2025b), MobileVD (Ben Yahia et al., 2025), AMD-Hummingbird (Isobe et al., 2025), Video DiT (Wu et al., 2025a), and Mobile-

Wan (Ghafoorian et al., 2026b). All of these methods, however, treat every token identically within a denoising step.

**Efficient Attention.** A large body of work reduces the cost of self-attention within a step through content-agnostic mechanisms. Fixed sparse patterns date back to classical efficient transformers (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020) and have been adapted to video DiTs through Sliding Tile Attention (Zhang et al., 2025c). Trainable-sparse variants such as VSA (Zhang et al., 2025b) and SLA (Zhang et al., 2026a) learn a sparse mask, while linear-attention backbones (Katharopoulos et al., 2020; Wang et al., 2020; Choromanski et al., 2021; Xiong et al., 2021) have recently been extended to video with SANA-Video (Chen et al., 2026) and HLA (Ackermann et al., 2026). Hybrid recurrent designs such as ReHyAt (Ghafoorian & Habibian, 2026) and Attention Surgery (Ghafoorian et al., 2026a), and architectural swaps like Mamba-based M4V (Huang et al., 2025b), follow the same philosophy of making attention globally cheaper. All of these methods, however, apply the same attention operator to every token, regardless of how easy or hard that token is to denoise.

**Caching.** Feature caching methods exploit temporal redundancy across denoising steps. ToCa (Zou et al., 2024) caches activations for tokens deemed stable and recomputes the rest; LiteAttention (Shmilovich et al., 2025) does the same at the level of attention; Clockwork Diffusion (Habibian et al., 2024) and object-centric editing (Kahatapitiya et al., 2024) reuse coarse features across timesteps. Two properties distinguish caching from our method. First, caching makes a *hard binary* cache-vs-compute decision, whereas we allocate a *soft* dense-vs-local budget that still updates every token. Second, caching amortizes skipped compute over many denoising steps; in the low-step regime this redundancy largely vanishes, making caching ineffective in precisely the deployment regime that matters most.

**Mixture of Experts.** Mixture-of-Experts DiTs (Fei et al., 2024; Wan et al., 2025) route different tokens through different compute paths, which is conceptually close to our routing. However, existing MoE-DiTs place experts on the *feed-forward* sub-layer. In video DiTs, the FFN is not the bottleneck; self-attention is, because of quadratic scaling in the token count. For instance in Wan2.2 A14B model at 720p resolution, 82% of the compute is spent at self-attention while only 13% goes to FFN. The (sparse) existing works on video MoE therefore leave the dominant cost untouched. Our method is complementary: it routes at the *attention* sub-layer and could, in principle, be combined with FFN-level MoE.

# 3 METHOD

In this work, we propose allocating compute *where it matters* on a per-token basis. Generated videos suggest unequal denoising complexity: some regions are smooth and uniform, while others contain complex patterns and textures. Motivated by this observation, we allocate computational budget according to token complexity, focusing on the attention operation, one of the main bottlenecks of DiTs.

To this end, we equip a few-step DMD model with an additional uncertainty branch that serves as a proxy for complexity prediction. We then use this branch to route tokens to either sparse local attention or full attention within the underlying DiT model, which we then term HetA (Heterogeneous self Attention) DiT. This preserves global context for difficult regions while reducing unnecessary dense computation on easier tokens. We first briefly review self-attention and DMD training for video diffusion models, and then describe our proposed uncertainty-based heterogeneous HetA DiT model.

## 3.1 PRELIMINARIES.

### 3.1.1 SELF-ATTENTION.

The key operation in video DiT models is self-attention, which relates all tokens to one another. Formally, given hidden tokens $Z \in \mathbb{R}^{N \times d}$, a transformer block computes queries, keys, and values:

$$Q = ZW_Q, \qquad K = ZW_K, \qquad V = ZW_V. \tag{1}$$

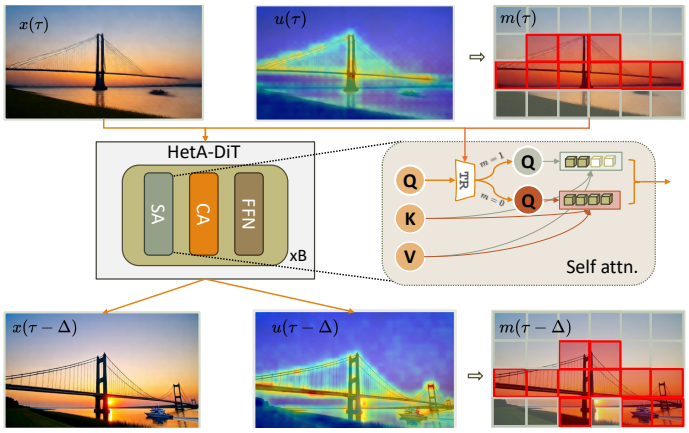


Figure 2: **HetA-DiT.** Our approach assigns heterogeneous self-attention computation to tokens with different denoising complexities. Given the uncertainty map $u(\tau)$ predicted at step $\tau$ by the dedicated branch, we construct a routing mask $m(\tau)$ by thresholding the predicted uncertainties. Based on this mask $m(\tau)$ and the latent $x(\tau)$, tokens with low uncertainty are processed using cheaper local self-attention in the following denoising step $\tau - \Delta$, while high-uncertainty tokens are processed with full attention. The resulting outputs are then used to determine heterogeneous self-attention processing in the next denoising step.

Self-attention then updates token $i$ using all tokens:

$$\mathrm{Attn}_{\mathrm{dense}}(i) = \sum_{j=1}^{N} a_{ij} v_j, \qquad a_{ij} = \frac{\exp(q_i^\top k_j/\sqrt{d_h})}{\sum_{\ell=1}^{N} \exp(q_i^\top k_\ell/\sqrt{d_h})}, \tag{2}$$

where $d_h$ is the channel dimension. This operation is expressive but has quadratic cost in the number of video tokens, $\mathcal{O}(N^2 d_h)$, making it expensive for long or high-resolution videos. **Local** self-attention, on the other hand, restricts the key-value set to a spatio-temporal neighborhood $\mathfrak{N}(i)$ around token $i$:

$$\mathrm{Attn}_{\mathrm{local}}(i) = \sum_{j \in \mathfrak{N}(i)} \tilde{a}_{ij} v_j. \tag{3}$$

If $|\mathfrak{N}(i)| = M \ll N$, local attention has cost $\mathcal{O}(NMd_h)$. However, purely local attention may lose global information that is important for object identity, occlusions, fast motion, and temporal consistency. Our goal is therefore to use dense attention only where it is needed.

### 3.1.2 Distribution Matching Distillation.

Distribution Matching Distillation (DMD) trains a student generator to match the distribution of a pretrained diffusion teacher using fewer denoising steps (Yin et al., 2024b). Given a prompt $y$ and noise $z$, the student produces a latent video

$$\tilde{x}_0 = G_\theta(z, y). \tag{4}$$

The sample is then re-noised to timestep $\tau$:

$$\tilde{x}_\tau = \alpha_\tau \tilde{x}_0 + \sigma_\tau \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{5}$$

where $\alpha_\tau$ and $\sigma_\tau$ define the teacher diffusion schedule. The teacher score $s_{\mathrm{T}}(\tilde{x}_\tau, \tau, y)$ provides the direction that moves the student sample toward the teacher distribution. DMD optimizes the student using a distribution-matching gradient of the form

$$\nabla_\theta \mathcal{L}_{\mathrm{DMD}} = \mathbb{E}_{z,y,\tau,\epsilon}\Big[w(\tau)\Big(s_{\mathrm{S}}(\tilde{x}_\tau, \tau, y) - s_{\mathrm{T}}(\tilde{x}_\tau, \tau, y)\Big)\nabla_\theta \tilde{x}_0\Big], \tag{6}$$

where $s_{\mathrm{S}}$ is the student score and $w(\tau)$ is a timestep-dependent weight. Thus, the student is updated so that its generated video distribution matches the teacher distribution. For video generation, DMD

is applied in the video latent space, and supervision comes from the pretrained teacher rather than from paired ground-truth videos.

Although DMD fine-tuning already drastically reduces the number of denoising steps required to obtain high-quality generated videos, we further optimize the underlying DiT model to enable even more efficient generation.

### 3.2 Proposed Approach.

Overall, our proposed approach consists of two stages. First, we train a full DiT model augmented with an uncertainty branch using a modified DMD objective. Then, we rely on this branch to introduce heterogeneous attention computation into the underlying DiT - resulting in our final proposed HetA-DiT model. We describe both stages in detail below.

#### 3.2.1 Stage 1: Uncertainty Prediction.

Our goal is to differentiate between tokens that are easier or more difficult to denoise. To this end, we extend the transformer with an additional branch that estimates the complexity of each latent token (after patchification). Formally, for each noisy latent $x_\tau$, the model predicts both a denoised latent $\tilde{x}_0$ and a token-wise uncertainty map $u$:

$$(\tilde{x}_0, u) = f_\theta(x_\tau, \tau, y), \quad u = \{u_i\}_{i=1}^N, \quad u_i \geq 0. \tag{7}$$

The scalar $u_i$ estimates the denoising uncertainty of token $i$.

To integrate the uncertainty prediction with the DMD gradient (6), we note first that the latter can be interpreted as a gradient of the MSE loss between the predicted clean sample $\tilde{x}_0$ and the pseudo-target $\hat{x}_0 = \texttt{stopgrad}\left[\tilde{x}_0 - w(\tau)\left(s_{\mathrm{S}}\left(\tilde{x}_\tau, \tau, y\right) - s_{\mathrm{T}}\left(\left(\tilde{x}_\tau, \tau, y\right)\right)\right)\right]$,

$$\nabla_\theta \mathcal{L}_{\mathrm{DMD}} = \mathbb{E}_{z,y,\tau,\epsilon}\left[\nabla_\theta \left\|\tilde{x}_0 - \hat{x}_0\right\|^2\right]. \tag{8}$$

This formulation can be naturally generalized to per-token negative Gaussian log-likelihood with predicted standard deviation,

$$\mathcal{L}_{\mathrm{DMD}}^{\mathrm{unc}} = \mathbb{E}_{z,y,\tau,\epsilon}\left[-\sum_{i=1}^{N} \log \mathcal{N}\left(\hat{x}_{0,i} \mid \tilde{x}_{0,i}, u_i\right)\right]. \tag{9}$$

As commonly known, Gaussian log-likelihood consists of the two non-constant terms,

$$-\log \mathcal{N}\left(\hat{x}_{0,i} \mid \tilde{x}_{0,i}, u_i\right) = \lambda_{\mathrm{rec}} \frac{\|\tilde{x}_{0,i} - \hat{x}_{0,i}\|^2}{u_i^2} + \lambda_{\log} \log u_i + \mathrm{const}. \tag{10}$$

The first term is proportional to the former MSE objective (8) and encourages larger uncertainty where the prediction error is high, while the second term prevents the trivial solution of predicting arbitrarily large uncertainty. Notably, we found that with this uncertainty-based weighting we do not need the per-sample DMD loss weighting which is otherwise commonly used in practice (Yin et al., 2024b). In the next stage, we use this predictive uncertainty as a proxy for token denoising difficulty.

#### 3.2.2 Stage 2: Uncertainty-Guided Token Densification.

After pretraining the uncertainty branch, we freeze it and use its predictions during the second DiT fine-tuning stage to construct a routing mask that assigns heterogeneous attention patterns to tokens, as illustrated in Figure 2.

Specifically, given a timestep $\tau$, the branch predicts token-wise uncertainties $\{u_i(\tau)\}_{i=1}^N$. We compute their mean $\mu_u(\tau)$ and standard deviation $\sigma_u(\tau)$, and define a per-step threshold for heterogeneous attention assignment:

$$T(\tau) = \mu_u(\tau) + c\,\sigma_u(\tau). \tag{11}$$

Here, $c$ is a hyperparameter controlling the number of tokens assigned to full attention. Smaller values of $c$ lower the threshold and route more tokens to full attention, whereas larger values make full-attention assignment less likely. Based on this threshold, we define a binary mask:

$$m_i(\tau) = \mathbf{1}\left[u_i(\tau) < T(\tau)\right]. \tag{12}$$




Figure 3: **Qualitative comparison of HetA-DiT to baseline DMD**. Both models run 3-step denoising. Our method maintains the high quality of the full step-distilled model while being substantially more efficient in an already low-compute regime.

Given this mask, uncertain tokens with $m_i = 0$ are processed with full attention, while certain tokens with $m_i = 1$ are processed locally. Furthermore, to increase token-wise coverage of full attention, at each transformer block $b$, we additionally select a subset of uncertain tokens and swap them with randomly selected certain tokens. This exposes more tokens to global processing over the course of generation, while ensuring that uncertain tokens still receive full attention most of the time.

Finally, the predicted mask is used for the next denoising step $\tau - \Delta$ to route the heterogeneous self-attention computation, where the resulting self-attention update can be written as:

$$z_i' = m_i \mathrm{Attn}_{\mathrm{local}}(i) + (1 - m_i)\mathrm{Attn}_{\mathrm{dense}}(i). \tag{13}$$

We fine-tune the resulting sparsified model using the standard DMD objective, as described in Sec. 3.1.2.

In terms of computational savings, let $\rho = \frac{1}{N}\sum_i (1 - m_i)$ be the fraction of dense-query tokens, and let $M = |\mathfrak{N}(i)|$ be the local window size. The resulting attention cost then can be estimated as:

$$\mathcal{O}\left(\rho N^2 d_h + (1 - \rho) N M d_h\right), \tag{14}$$

compared with $\mathcal{O}(N^2 d_h)$ for fully dense attention. Thus, when only a small fraction of tokens is uncertain and $M \ll N$, the model reduces attention cost while preserving dense global context for the tokens that actually require it. In our experiments the dense ratio $\rho \approx 0.2$.

## 4 Experimental Setup

**Training Setting.** We train our proposed method in two stages. First, we train a DMD model augmented with an additional uncertainty branch using the modified DMD objective described in Sec. 3.2.1. The first-stage training is performed for 15k iterations. In the second stage, we freeze the pre-trained uncertainty branch and fine-tune the remaining DiT parameters for 6k iterations using the standard DMD objective, with heterogeneous sparsification guided by the branch predictions. Since the routing mask at each denoising step is based on uncertainty from the previous step, training uses an additional full-attention student forward pass without gradient propagation to estimate it. Specifically, for a randomly sampled timestep, we predict the uncertainty at an earlier timestep and use it to construct the sparse attention mask for the current step. At inference time, the first step is run without sparsification, while for the later steps no extra transformer pass is needed, as uncertainty predictions from the previous denoising step are reused to build the mask for the next step.

For both stages, we use 4 H100 GPUs and the original pretrained Wan model as the teacher. We also follow a prompt-only training protocol: visual supervision is provided by the pretrained generative model rather than ground-truth videos. This allows us to train the model without requiring direct access to video annotations. For prompts, we generate captions for a subset of videos from the

VIPE1M dataset (Huang et al., 2025a) using the Qwen3 model (Yang et al., 2025a). We provide further training details in the supp. material.

**Evaluation Setting.** To evaluate our proposed approach, we fine-tune HetA-DiT starting from DMD models based on Wan2.2 5B and Wan2.1 1.3B (Wan et al., 2025), demonstrating the generalizability of our method. For a fair comparison with other efficient diffusion variants, we fine-tune several training-based methods under DMD step distillation, as well as evaluate training-free approaches under a comparable setting. For all methods, we use 3 denoising steps at inference time and generate videos at resolutions of $121\times704\times1280$ for Wan2.2-based models and $81\times480\times832$ for Wan2.1-based models. We conduct the quantitative evaluation on the full set of extended VBench (Huang et al., 2024) and VBench-2.0 Zheng et al. (2025) prompts.

To complement the automatic metrics, we conduct a **method-blinded human preference study**. Evaluators were presented with GPT-enhanced long prompts from VBench and asked to select the preferred video, or mark the comparison as a tie, based on semantic coherence and visual quality. For each prompt, videos from the two methods were presented side by side, with their left-right placement randomized independently across questions. In total, the study comprised 1,248 paired comparisons from 19 distinct evaluators, covering 1,082 unique videos and 667 unique prompts. The numbers of videos and prompts differ because some prompts were evaluated with multiple random seeds.

### 4.1 Compute Complexity Assessment.

We analyze how the proposed heterogeneous attention mechanism reduces the number of **floating-point operations** compared to the full DiT model, both at the attention level and at the full Transformer block level. We also compare the resulting speedup against other efficient attention methods. We utilize DeepSpeed library to measure the total floating point operations.

In addition to FLOPs reductions, we also show that our method reduces the actual **latency** of the Transformer. We integrate HetA-DiT with FlexAttention, a built-in PyTorch module for implementing custom attention masks with optimized block-sparse kernels, and profile dense versus heterogeneous attention under the same model and resolution settings to measure the realized wall-clock speedup.

## 5 Results

**VBench.** We compare HetA-DiT on VBench Huang et al. (2024) in Tables 1 and 2, using Wan2.2 and Wan2.1 as the base models, respectively. For clarity, we highlight training-free efficient methods in yellow. Our model performs on par with the full Wan models, which use 50 denoising steps, while achieving a substantial end-to-end sampling speedup. Moreover, HetA-DiT performs on par with the 3-step DMD model, which uses the original DiT with dense attention across all video tokens without any sparsification. This supports our key insight that heterogeneous computation based on token difficulty can preserve generation quality while improving efficiency. We also provide a qualitative comparison with the full 3-step DMD approach in Figure 3 (we show additional qualitative comparisons to this and other methods in the supp. material). Finally, HetA-DiT outperforms strong existing efficient video generation methods. We specifically note gap compared with the caching-based ToCa Zou et al. (2024), which, despite using non-uniform caching, is less suited to low-step generation due to its reliance on redundancy across many denoising steps, as also noted in prior work Zou et al. (2026).

**VBench-2.0.** As shown in Table 3, HetA-DiT maintains strong performance on VBench-2.0, outperforming both the full Wan2.2 model, which requires 50 denoising steps, and the step-distilled DMD model, which uses 2 steps.

**Human Preference Evaluation.** Tab. 4 shows the results of the user study, where we compare our model against DMD with 2 and 3 steps, as well as Attention Surgery, another efficient video generation method. Overall, we observe that our method is preferred almost as often as the full DMD model with 3 denoising steps. Compared to the other two baselines, HetA-DiT achieves significantly higher preference scores.

Table 1: **Comparison with state-of-the-art efficient video diffusion models** on VBench. Unless specified otherwise, all efficient models are implemented using Wan2.2 5B as the base model.

| **Models with 2B–5B parameters** | Tot.↑ | Qual.↑ | Sem.↑ |
|---|---|---|---|
| Open-Sora Plan V1.3 (Lin et al., 2024) | 77.23 | 80.14 | 65.62 |
| Open-Sora V1.2 Zheng et al. (2024) | 79.76 | 81.35 | 73.39 |
| LTX-Video (HaCohen et al., 2024) | 80.00 | 82.30 | 70.79 |
| SnapGenV (Wu et al., 2025b) | 81.14 | 83.47 | 71.84 |
| Hummingbird (Isobe et al., 2025) | 81.35 | 83.73 | 71.84 |
| Mob. Vid. DiT Mob. (Wu et al., 2025a) | 81.45 | 83.12 | 74.76 |
| CogVideoX 2B (Yang et al., 2025b) | 81.55 | 82.48 | 77.81 |
| Neodragon (Karnewar et al., 2026) | 81.61 | 83.68 | 73.36 |
| PyramidalFlow (Jin et al., 2025) | 81.72 | 84.74 | 69.62 |
| CogVideoX 5B (Yang et al., 2025b) | 81.91 | 83.05 | 77.33 |
| CogVideoX1.5 5B (Yang et al., 2025b) | 82.01 | 82.72 | 79.17 |
| Mob. Vid. DiT Serv. (Wu et al., 2025a) | 83.09 | 84.65 | 76.86 |
| Wan2.2 5B (Wan et al., 2025) | 83.28 | 85.03 | 76.28 |
| **Efficient Models** (Res. $704 \times 1280$) | | | |
| STA (Zhang et al., 2025c) (HunyuanVid) | 83.00 | **85.37** | 73.52 |
| DMD2s (2 steps) | 82.48 | 83.05 | 80.22 |
| DMD3s (3 steps) | 82.94 | 83.56 | 80.43 |
| DMD3s-Attn.Surg. (Ghafoorian et al., 2026a) | 82.62 | 83.60 | 78.71 |
| DMD3s-ReHyAt (Ghafoorian & Habibian, 2026) | 82.55 | 82.78 | **81.61** |
| DMD3s-VSA (Zhang et al., 2026b) | 82.24 | 82.71 | 80.38 |
| DMD3s-Toca (Zou et al., 2024) | 80.52 | 81.40 | 76.98 |
| DMD3s-Jenga (Zhang et al., 2026c) | 82.72 | 83.58 | 79.22 |
| DMD3s-HetA-DiT (Ours) | **83.28** | 83.71 | 81.52 |

Table 2: **Comparison with state-of-the-art efficient video diffusion models** on VBench, using Wan2.1 as the base model. (*) denotes reproduced results.

| **Models** (Res. $480 \times 832$) | Total↑ | Quality↑ | Semantic↑ |
|---|---|---|---|
| Wan2.1 1.3B (Wan et al., 2025) | 83.31 | 85.23 | 75.65 |
| Wan2.1 1.3B* (Wan et al., 2025) | 83.10 | 85.10 | 75.12 |
| VSA (Zhang et al., 2026b) | 82.77 | 83.60 | 79.47 |
| DMD3s (3 steps) | **83.61** | **84.82** | **78.78** |
| DMD3s-HetA-DiT (Ours) | 83.10 | 84.23 | 78.55 |

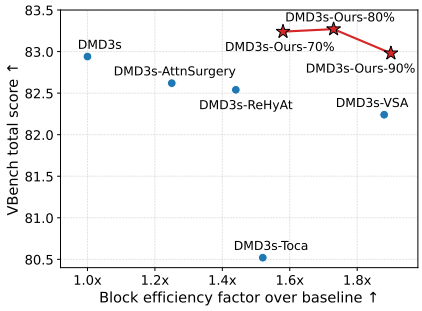


Figure 4: **Quality–efficiency trade-off comparison** of HetA under varying sparsity percentages against state-of-the-art efficient-attention video diffusion models.

**Efficiency.** We report the efficiency improvements of our proposed HetA-DiT in Fig. 4. We first examine how the fraction of tokens routed to local self-attention, controlled by the threshold coefficient $c$, affects both performance and speed-up. To this end, we evaluate three variants with average local-compute ratios of $70\%$, $80\%$, and $90\%$. Overall, routing $70\%$ or $80\%$ of tokens to local computation yields comparable performance, with the reduced dense-computation budget leading to larger speed-up gains. Increasing the average masking ratio further to $90\%$ provides additional computational savings, but at the cost of a lower evaluation score. The optimal operating point is therefore the setting with $c = 0.9$, which achieves an overall $1.73\times$ speed-up for the total Transformer block computation and a VBench score of $83.28$. Among other efficient video generation methods, our approach achieves competitive performance-computation trade-off, even at very low dense-attention ratios.

As shown in Table 5, the lower compute burden consequently translates into notable reductions in latencies. While the speed-ups of HetA-DiT are substantial overall, they are particularly larger for high-resolution latents, where the increased input length creates an even more crucial compute bottleneck for self-attention. By reducing the number of tokens processed with dense self-attention to $20\%$ on average, our model becomes $1.63\times$ faster than the full DiT model.

### 5.1 Ablation Studies

Here, we provide a series of ablation experiments to study the individual components of our proposed method. Unless specified otherwise, all models are based on the DMD Wan2.2 model and use

Table 3: **Comparison with state-of-the-art video diffusion models** on VBench2.0

| **Models** (Res. $480 \times 832$) | Tot.↑ | Hum.Fid. ↑ | Creativ. ↑ | Contr. ↑ | Com. Sense ↑ | Phys. ↑ |
|---|---|---|---|---|---|---|
| Wan2.1 1.3B | 56.0 | 80.7 | 48.7 | 34.0 | 63.4 | 53.8 |
| Wan2.2 5B | 58.1 | 79.4 | 57.7 | 36.8 | 64.1 | 51.3 |
| HunyuanVideo | 55.3 | 82.4 | 41.8 | 28.6 | 63.4 | 60.2 |
| CogVideoX-1.5 | 53.4 | 72.1 | 43.7 | 29.6 | 63.2 | 48.2 |
| DMD2s (2 steps) (Wan2.2) | 57.9 | 76.9 | 54.2 | 37.9 | 59.9 | 60.9 |
| DMD3s (3 steps) (Wan2.2) | **60.8** | **80.7** | **58.7** | **36.2** | **64.5** | **63.9** |
| DMD3s-HetA-DiT (Wan2.2) | 58.9 | 79.8 | 55.0 | 35.9 | 62.5 | 61.4 |

Table 4: **Results of the method-blind human visual preference study** over 1,248 pairs of videos.

| Baseline | Human Preference % | | |
|---|---|---|---|
| | HetA-DiT | No preference | Baseline |
| DMD2s (2steps) | 50% | 26% | 24% |
| DMD3s (3steps) | 32% | 32% | 36% |
| DMD3s-Attn. Surg. | 68% | 13% | 19% |

Table 5: **Latency speedup of HetA-DiT** relative to dense attention, including amortized BlockMask construction cost.

| Model | Resolution | Attn. Latency Speedup | Block Latency Speedup | Attn. FLOPs Reduction | Block FLOPs Reduction |
|---|---|---|---|---|---|
| Wan2.1-1.3B | 480×832 | 1.52× | 1.34× | 4.87× | 2.25× |
| Wan2.2-5B | 704×1280 | 1.84× | 1.63× | 4.85× | 1.73× |
| Wan2.2-14B | 480×832 | 1.57× | 1.30× | 4.91× | 1.97× |

3 denoising steps at inference time. For our proposed approach, we set $c = 0.9$, which corresponds on average to $80\%$ sparse computation. For a fair comparison, we use a fixed masking ratio of $80\%$ for the other methods. We provide further ablation studies in the supp. material.

Table 6: **Ablation study of the self-attention routing type.**

| Routing Strategy | Total↑ | Quality↑ | Semantic↑ |
|---|---|---|---|
| Random routing | 82.78 | 83.37 | 80.45 |
| Fixed uniform routing | <u>83.13</u> | **83.76** | <u>80.58</u> |
| Local-only | 82.85 | 83.56 | 79.99 |
| Ours | **83.28** | <u>83.71</u> | **81.52** |

Table 7: **Ablation study of local neighborhood size.**

| Local neighb. | Total↑ | Quality↑ | Semantic↑ | Attn. Speedup |
|---|---|---|---|---|
| 13×13×13 | **83.36** | **83.99** | <u>80.82</u> | 4.76× |
| 11×11×11 | <u>83.28</u> | <u>83.71</u> | **81.52** | <u>4.85</u>× |
| 9×9×9 | 83.22 | 83.89 | 80.54 | **4.91**× |

**Routing Strategy.** Table 6 compares different strategies for routing tokens to dense attention. Our method uses the proposed uncertainty-based routing branch to identify tokens that benefit most from full self-attention. As baselines, random routing selects a random subset of tokens for dense attention independently at each transformer block, while fixed uniform routing selects tokens uniformly from the token volume using a fixed pattern. We also evaluate a local-only variant, which can be viewed as a lower-bound setting: no tokens are routed to dense attention, and all tokens are updated only through local self-attention. The results show that uncertainty-based routing achieves the best total and semantic scores, indicating that the uncertainty signal is effective for identifying tokens that require a larger receptive field. Although naive routing strategies achieve reasonable performance, they are consistently inferior to our uncertainty-based selection, demonstrating the importance of adaptive token routing. Finally, we provide further uncertainty analysis in the supp. material.

**Local attention neighborhood.** Table 7 ablates the local neighborhood size used by the local self-attention component of our method. We employ local self-attention with a predefined 3D neighborhood, where increasing the kernel size expands the receptive field but also increases computational cost. As expected, the largest neighborhood, $13\times13\times13$, achieves the highest total and quality scores, but it comes with a lower transformer speedup. Reducing the neighborhood to $9\times9\times9$ improves efficiency but leads to weaker semantic performance. The $11\times11\times11$ setting provides the best overall trade-off: it preserves strong quality, while maintaining the highest speedup. We therefore use a kernel size of $11$ as the default setting.

## 6 Conclusion

In this work, we presented HetA-DiT, a heterogeneous attention framework for efficient video diffusion transformers that allocates computation according to token denoising difficulty. Using a lightweight uncertainty branch, our method routes reliable tokens to cheaper local attention and uncertain tokens to dense global attention, enabling content- and timestep-adaptive sparsification. Experiments on Wan models show that HetA-DiT preserves the quality of dense-attention baselines while substantially reducing attention computation and consistently outperforming existing efficient video generation methods. Ablations further confirm the importance of uncertainty-based routing.

## 7 AI USE STATEMENT

In this work, we have used generative AI tools to provide feedback on research methodology or experiments, summarize or analyze existing literature, and implement methods. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

Hanno Ackermann, Hong Cai, Mohsen Ghafoorian, and Amirhossein Habibian. HLA: Hadamard Linear Attention, 2026. URL https://arxiv.org/abs/2602.12128.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

Haitam Ben Yahia, Denis Korzhenkov, Ioannis Lelekas, Amir Ghodrati, and Amirhossein Habibian. Mobile video diffusion. In *ICCV*, 2025.

Junsong Chen, Yuyang Zhao, Jincheng Yu, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Daquan Zhou, Huan Ling, et al. Sana-video: Efficient video generation with block linear diffusion transformer. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019. URL https://arxiv.org/abs/1904.10509.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *ICLR*, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

Xiangyu Fan, Zesong Qiu, Zhuguanyu Wu, Fanzhou Wang, Zhiqian Lin, Tianxiang Ren, Dahua Lin, Ruihao Gong, and Lei Yang. Phased dmd: Few-step distribution matching distillation via score matching within subintervals, 2026. URL https://arxiv.org/abs/2510.27684.

Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, Youqiang Zhang, and Junshi Huang. Scaling diffusion transformers to 16 billion parameters. *arXiv preprint arXiv:2407.11633*, 2024.

Mohsen Ghafoorian and Amirhossein Habibian. Rehyat: Recurrent hybrid attention for video diffusion transformers. In *CVPR*, 2026.

Mohsen Ghafoorian, Denis Korzhenkov, and Amirhossein Habibian. Attention surgery: An efficient recipe to linearize your video diffusion transformer. In *CVPR*, 2026a.

Mohsen Ghafoorian, Denis Korzhenkov, Adil Karjauv, Ioannis Lelekas, Noor Fathima, Spyridon Stasis, Hanno Ackermann, Boris van Breugel, Markus Nagel, Fatih Porikli, et al. Mobilewan: Closing the quality gap for mobile video diffusion. *arXiv preprint arXiv:2607.06173*, 2026b.

Amirhossein Habibian, Amir Ghodrati, Noor Fathima, Guillaume Sautiere, Risheek Garrepalli, Fatih Porikli, and Jens Petersen. Clockwork diffusion: Efficient generation with model-step distillation. In *CVPR*, 2024.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixe, and Sanja Fidler. Vipe: Video pose engine for 3d geometric perception. In *NVIDIA Research Whitepapers*, 2025a.

Jiancheng Huang, Gengwei Zhang, Zequn Jie, Siyu Jiao, Yinlong Qian, Ling Chen, Yunchao Wei, and Lin Ma. M4v: Multi-modal mamba for text-to-video generation. *arXiv preprint arXiv:2506.10915*, 2025b.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024.

Hunyuan Foundation Model Team. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Takashi Isobe, He Cui, Dong Zhou, Mengmeng Ge, Dong Li, and Emad Barsoum. Amd-hummingbird: Towards an efficient text-to-video model. *arXiv preprint arXiv:2503.18559*, 2025.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong MU, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=66NzcRQuOq.

Kumara Kahatapitiya, Adil Karjauv, Davide Abati, Fatih Porikli, Yuki M Asano, and Amirhossein Habibian. Object-centric diffusion for efficient video editing. In *ECCV*, 2024.

Animesh Karnewar, Denis Korzhenkov, Ioannis Lelekas, Noor Fathima, Adil Karjauv, Mohsen Ghafoorian, and Amirhossein Habibian. Neodragon: Mobile video generation using diffusion transformer. In *ICLR*, 2026.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: fast autoregressive transformers with linear attention. In *ICML*, 2020.

Bin Lin, Yunyang Ge, Xinhua Cheng, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024. URL https://arxiv.org/abs/2412.00131.

Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. In *ICML*, 2025. URL https://proceedings.mlr.press/v267/lin25m.html.

Haosong Liu, Yuge Cheng, Wenxuan Miao, Zihan Liu, Aiyue Chen, Jing Lin, Yiwu Yao, Chen Chen, Jingwen Leng, Minyi Guo, and Yu Feng. Astraea: A token-wise acceleration framework for video diffusion transformers. In *ICLR*, 2026. URL https://openreview.net/forum?id=e8P4Oo8S6U.

Weili Nie, Julius Berner, Nanye Ma, Chao Liu, Saining Xie, and Arash Vahdat. Transition matching distillation for fast video generation, 2026.

Elia Peruzzo, Adil Karjauv, Nicu Sebe, Amir Ghodrati, and Amir Habibian. Adaptor: Adaptive token reduction for video diffusion transformers. In *CVPR*, 2025.

Dor Shmilovich, Tony Wu, Aviad Dahan, and Yuval Domb. Liteattention: A temporal sparse attention for diffusion transformers, 2025.

Team Wan et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020. URL https://arxiv.org/abs/2006.04768.

Yushu Wu, Yanyu Li, Anil Kag, Ivan Skorokhodov, Willi Menapace, Ke Ma, Arpit Sahni, Ju Hu, Aliaksandr Siarohin, Dhritiman Sagar, Yanzhi Wang, and Sergey Tulyakov. Taming diffusion transformer for real-time mobile video generation. *arXiv preprint arXiv:2507.13343*, 2025a. URL https://arxiv.org/abs/2507.13343.

Yushu Wu, Zhixing Zhang, Yanyu Li, Yanwu Xu, Anil Kag, Yang Sui, Huseyin Coskun, Ke Ma, Aleksei Lebedev, Ju Hu, et al. Snapgen-v: Generating a five-second video within five seconds on a mobile device. In *CVPR*, 2025b.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *AAAI*, 2021.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Zhang Yuxuan, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025b. URL https://openreview.net/forum?id=LQzN6TRFg9.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved Distribution Matching Distillation for Fast Image Synthesis. In *NeurIPS*, 2024a. URL https://openreview.net/forum?id=tQukGCDaNT.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step Diffusion with Distribution Matching Distillation. In *CVPR*, 2024b.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *NeurIPS*, 2020.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattn: Accurate sparse attention accelerating any model inference. In *International Conference on Machine Learning (ICML)*, 2025a.

Jintao Zhang, Haoxu Wang, Kai Jiang, Shuo Yang, Kaiwen Zheng, Haocheng Xi, Ziteng Wang, Hongzhou Zhu, Min Zhao, Ion Stoica, Joseph E. Gonzalez, Jianfei Chen, and Jun Zhu. SLA: Beyond sparsity in diffusion transformers via fine-tunable sparse–linear attention. In *ICLR*, 2026a. URL https://openreview.net/forum?id=eD8IPvNoZB.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric P Xing, and Hao Zhang. Faster video diffusion with trainable sparse attention. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast video generation with sliding tile attention. In *ICML*, 2025c. URL https://openreview.net/forum?id=U74MOXPEJd.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric Xing, and Hao Zhang. Faster video diffusion with trainable sparse attention. *Advances in Neural Information Processing Systems*, 38:152509–152534, 2026b.

Yuechen Zhang, Jinbo Xing, Shaoteng Liu, Bohao PENG, Xin Tao, Pengfei Wan, Eric Lo, Jiaya Jia, et al. Training-free efficient video generation via dynamic token carving. *Advances in Neural Information Processing Systems*, 38:81913–81946, 2026c.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, et al. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

Kaiwen Zheng, Yuji Wang, Qianli Ma, Huayu Chen, Jintao Zhang, Yogesh Balaji, Jianfei Chen, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Large scale diffusion distillation via score-regularized continuous-time consistency. In *ICLR*, 2026. URL https://openreview.net/forum?id=2uNlM353RI.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. *arXiv preprint arXiv:2410.05317*, 2024.

Chang Zou, Changlin Li, Songtao Liu, Zhao Zhong, Kailin Huang, and Linfeng Zhang. Disca: Accelerating video diffusion transformers with distillation-compatible learnable feature caching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2602.05449. to appear.

# A Appendix

# B Training Details

## B.1 Training of HetA-DiT.

We provide additional details on the two training stages of HetA-DiT. The model is trained using the standard DMD pipeline, with the objective functions for each stage described in the main paper. The first training stage consists of 15k fine-tuning iterations using a per-GPU batch size of 2 across 4 GPUs. The student learning rate is set to $1 \times 10^{-5}$, while the critic learning rate is set to $5 \times 10^{-6}$. We employ a constant learning rate schedule with a 10-step warm-up phase. Optimization is performed using AdamW with a weight decay of 0.01, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. Training is conducted in mixed precision using bfloat16. For timestep sampling, we draw timesteps from a shifted uniform distribution with a shift parameter of $s = 5$. Finally, the classifier guidance scale is set to 5. The second training stage consists of 6k fine-tuning iterations using a per-GPU batch size of 1 across 4 GPUs. Timesteps are again sampled from a shifted uniform distribution; however, the shift parameter is set to $s = 1$. All other hyperparameters remain unchanged.

## B.2 Training of baselines.

For Attention Surgery, we replaced 20 transformer blocks with hybrid blocks with the hybrid rate of 8 (20×R8 configuration). Following the recipe of Ghafoorian et al. (2026a), the training comprised three stages: block-wise distillation, flow-matching finetuning, and DMD2 distillation. The same 3-stage pipeline was applied to ReHyAt but in that case all 30 blocks of Wan2.2 video transformer have been replaced. Every chunk consisted of 3 latent frames, and chunk overlap was set to a single frame. Both Attention Surgery and ReHyAt used polynomials of degree 2 in their linear attention branch.

In our VSA training, we mostly followed the recipe provided by FastVideo library and set the sparsity rate to 0.8 while spatiotemporal tile size was $4 \times 4 \times 4$. In contrast, Jenga is a non-trainable block-sparse attention method. We applied it on top of our baseline DMD checkpoint with sparsity ratio of 0.8, threshold probability of 0.9, and tile size equal to 64.

Similar to Jenga, ToCa is a training-free caching method. To make ToCa more suited for the step-distilled regime, we run it only on 1 out of 3 steps: the first and last denoising steps are evaluated fully, while ToCa caching is applied only on the middle step. We set the token refresh ratio to $0.6$, meaning that on the cached step $60\%$ of FFN tokens are recomputed and the remaining tokens reuse cached activations; with the ToCa text-to-video module scheduler, this corresponds to an effective FFN refresh ratio of approximately $89.7\%$ and an FFN cache ratio of approximately $10.3\%$.

# C Additional Results

**Per-dimension Evaluation Score.** We provide per-dimensions comparisons our proposed HetA-DiT on the VBench and VBench-2.0 to the other state-of-the-art efficient models in Figure 5 and Figure 6.

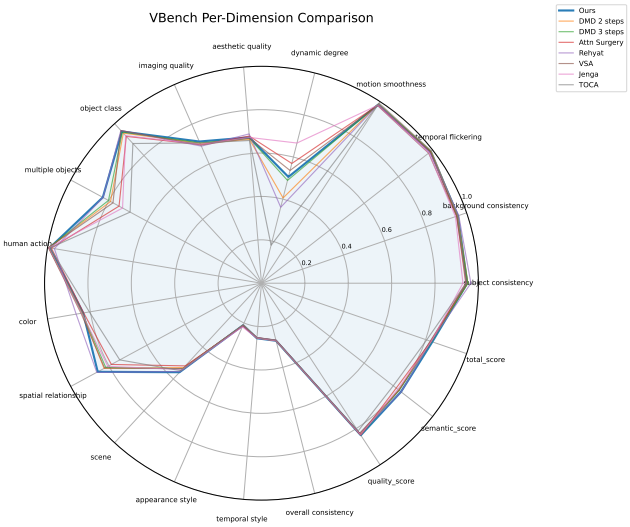


Figure 5: Radar plot comparing HetA-DiT on VBench (Huang et al., 2024) to the state-of-the-art efficient video generation models.

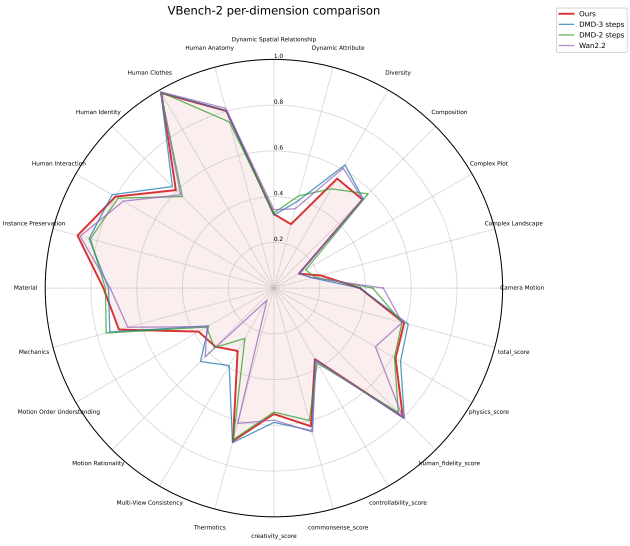


Figure 6: Radar plot comparing HetA-DiT on VBench-2.0 (Zheng et al., 2025) to the state-of-the-art efficient video generation models.

**Random Token Percentage.** Table 8 studies the effect of injecting random token selection into our uncertainty-based routing. Purely random selection performs worse than uncertainty-based routing, confirming that uncertainty provides a stronger signal for identifying tokens that require full attention. However, adding a small amount of randomness improves performance. Random selection periodically exposes tokens that are not selected by the uncertainty criterion to dense attention, thereby increasing their receptive field. For a single denoising step, with random swap ratio $r$ and $B$ transformer blocks, the probability that a token receives dense attention at least once through random selection is $1-(1-r)^B$. For $r=0.07$ and $B=30$, this probability is $1-(1-0.07)^{30}\approx 0.89$. It further increases across denoising steps, improving token coverage over the full generation process. In contrast, fully random selection weakens performance, as shown by the degradation at $20\%$. We therefore use $7\%$ random tokens as the default setting, which provides the best balance.

Table 8: **Ablation study of random token percentage**. Adding a random swapping component to uncertainty-based routing increases dense-attention coverage, leading to improved results.

| Random % | Total↑ | Quality↑ | Semantic↑ |
|---|---|---|---|
| 20% (all) | 82.78 | 83.37 | 80.45 |
| 7% | **83.27** | **83.71** | **81.52** |
| 3% | <u>83.18</u> | <u>83.63</u> | <u>81.39</u> |
| 0% | 83.00 | 83.48 | 81.25 |

## D Additional Qualitative Comparisons

We provide additional qualitative comparisons of our methods to other efficient video generation baselines in Figure 7, Figure 8, Figure 9, Figure 10 and Figure 11.

## E Uncertainty Branch

We provide intuition behind the proposed uncertainty analysis. More specifically, the uncertainty prediction can be motivated through a simple SNR-based interpretation. Suppose that, for a given token, the model prediction can be viewed as a Gaussian estimate of the clean latent:

$$x_0 \approx \hat{x}_0 + u\eta, \qquad \eta \sim \mathcal{N}(0,1). \tag{15}$$

Equivalently, $\hat{x}_0 \approx x_0 - u\eta$. Re-noising this prediction to timestep $\tau$ gives

$$\begin{aligned} \hat{x}_\tau &= \alpha_\tau \hat{x}_0 + \sigma_\tau \epsilon \\ &= \alpha_\tau x_0 + \sigma_\tau \epsilon - \alpha_\tau u\eta. \end{aligned} \tag{16}$$

Assuming that $\epsilon$ and $\eta$ are independent standard Gaussian variables, the effective noise variance becomes

$$\sigma_\tau^2 + \alpha_\tau^2 u^2. \tag{17}$$

Thus, the effective signal-to-noise ratio is

$$\mathrm{SNR}_{\mathrm{eff}}(\tau, u) = \frac{\alpha_\tau^2}{\sigma_\tau^2 + \alpha_\tau^2 u^2}. \tag{18}$$

We match this quantity to the SNR of an artificial timestep $\tau'$:

$$\frac{\alpha_\tau^2}{\sigma_\tau^2 + \alpha_\tau^2 u^2} = \frac{\alpha_{\tau'}^2}{\sigma_{\tau'}^2}, \qquad \tau' \geq \tau. \tag{19}$$

Under this interpretation, a larger uncertainty $u$ makes the token behave as if it were at a noisier timestep $\tau'$. Therefore, high-uncertainty tokens should receive more global computation. This is similar in spirit to SNR matching ideas used in pyramidal or multi-scale diffusion methods.

We further show some qualitative examples of the predicted uncertainty maps in Figure 12, Figure 13, Figure 14, Figure 15 and Figure 16.




Figure 7: Qualitative comparison of HetA-DiT to the state-of-the-art efficient video generation models. **Prompt**: *A 3D model of a 1800s victorian house.*

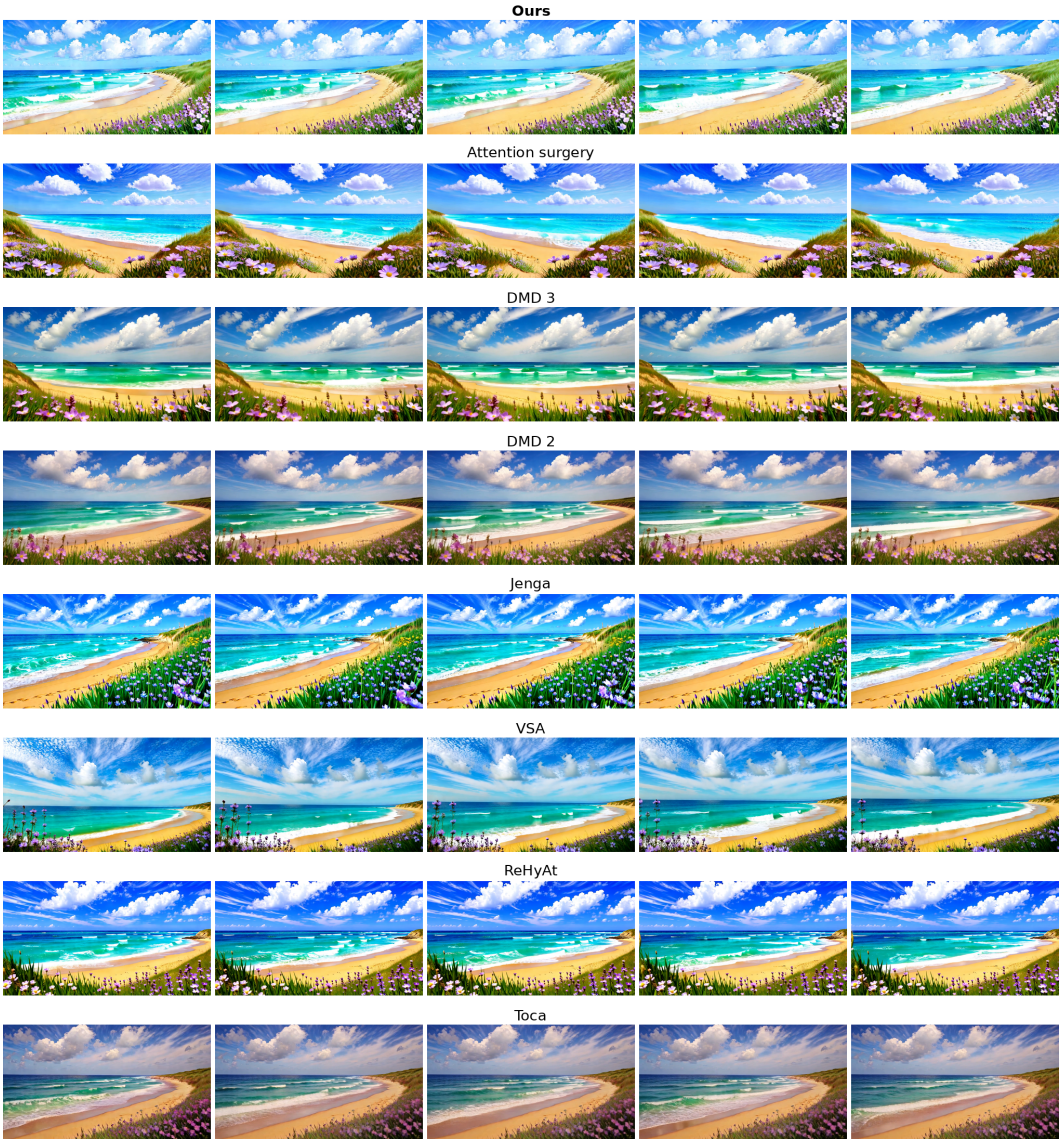


Figure 8: Qualitative comparison of HetA-DiT to the state-of-the-art efficient video generation models. **Prompt**: *A beautiful coastal beach in spring, waves lapping on sand, Van Gogh style.*

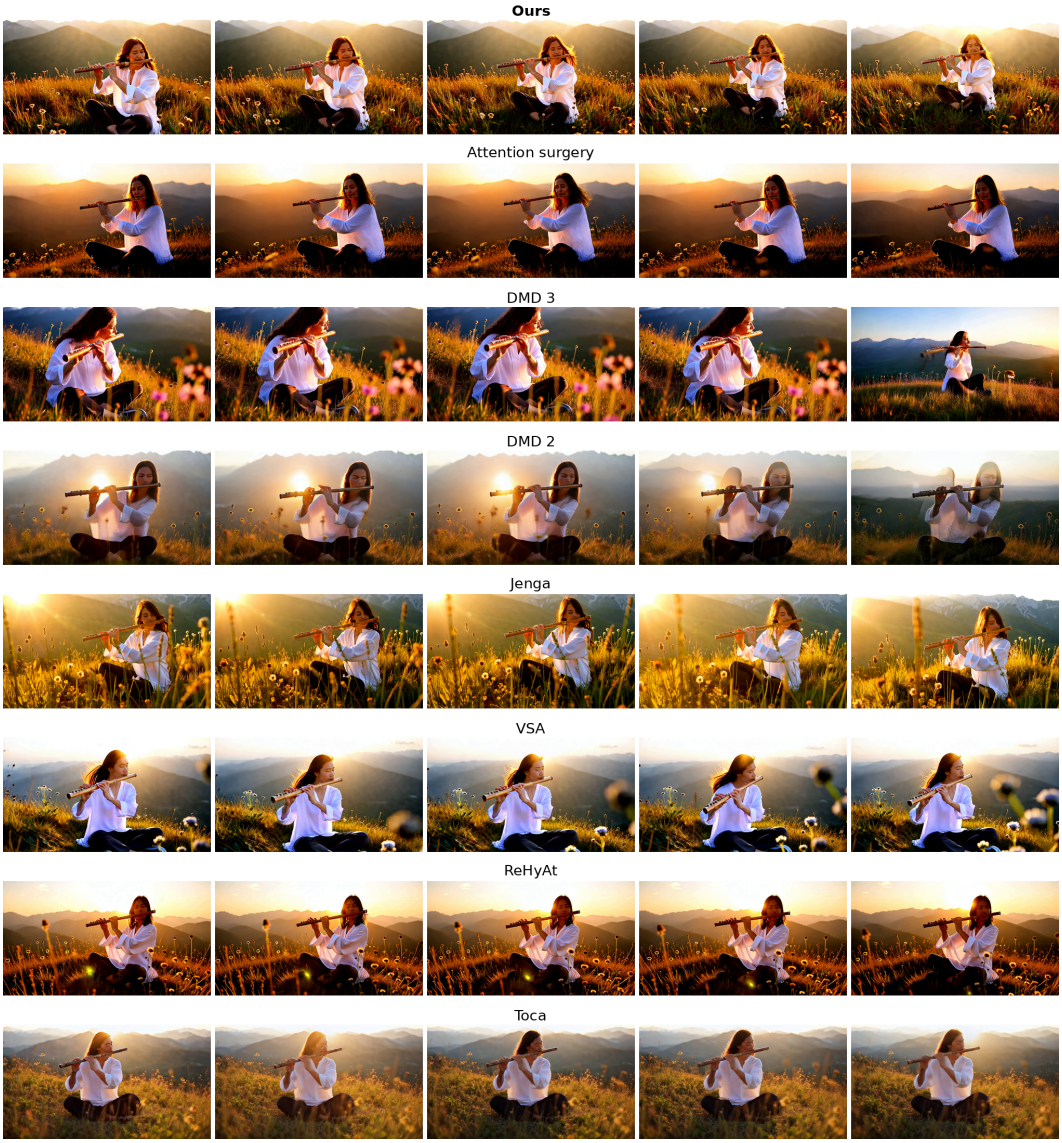


Figure 9: Qualitative comparison of HetA-DiT to the state-of-the-art efficient video generation models. **Prompt**: *A person is playing flute.*

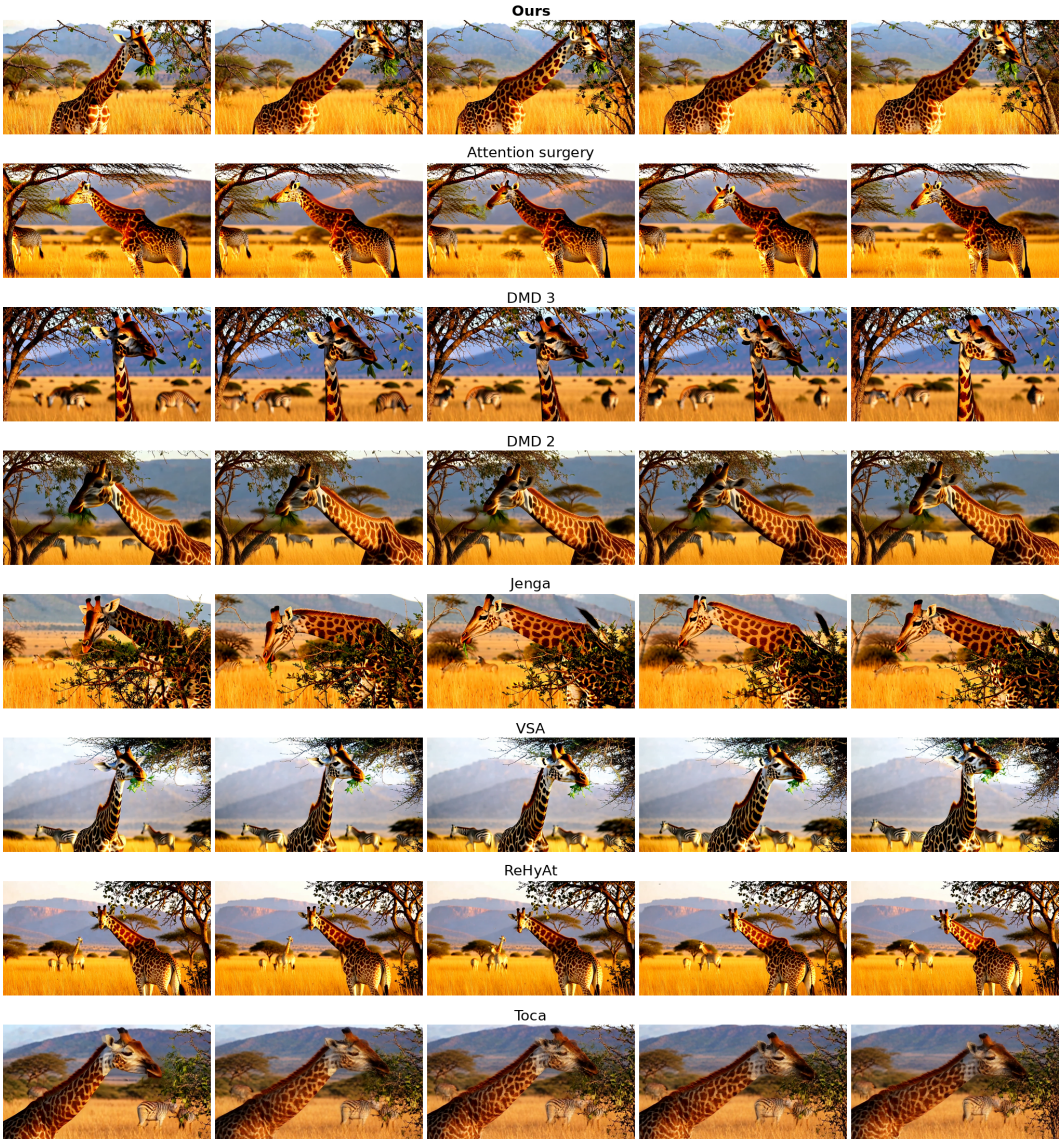


Figure 10: Qualitative comparison of HetA-DiT to the state-of-the-art efficient video generation models. **Prompt**: *a giraffe*.




Figure 11: Qualitative comparison of HetA-DiT to the state-of-the-art efficient video generation models. **Prompt**: *A jellyfish floating through the ocean, with bioluminescent tentacles*.

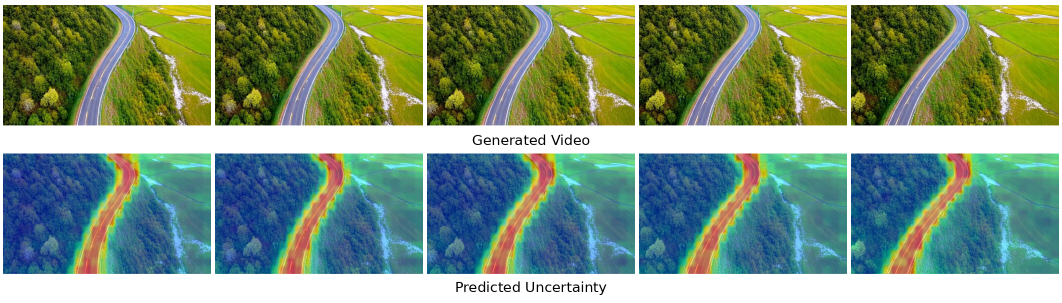


Figure 12: Example of the predicted uncertainty map at diffusion timestep $\sigma = 0.6$.

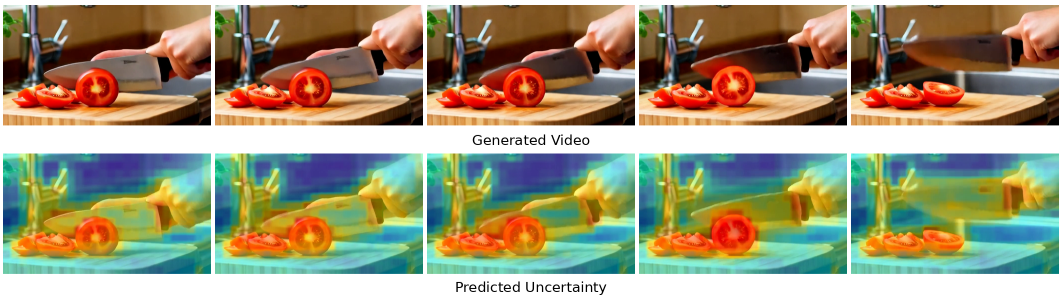


Figure 13: Example of the predicted uncertainty map at diffusion timestep $\sigma = 0.6$.

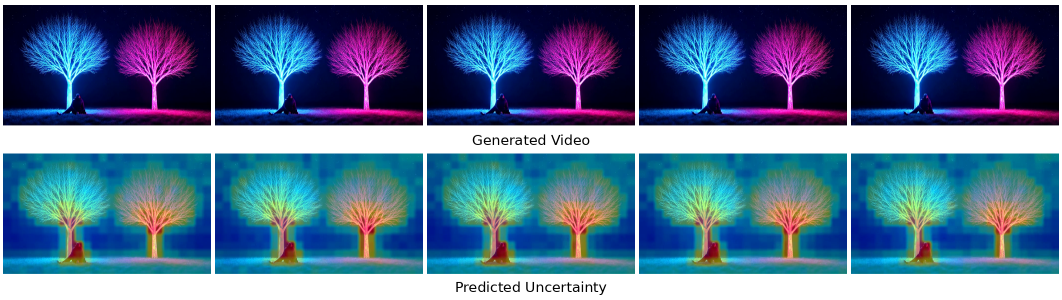


Figure 14: Example of the predicted uncertainty map at diffusion timestep $\sigma = 0.6$.

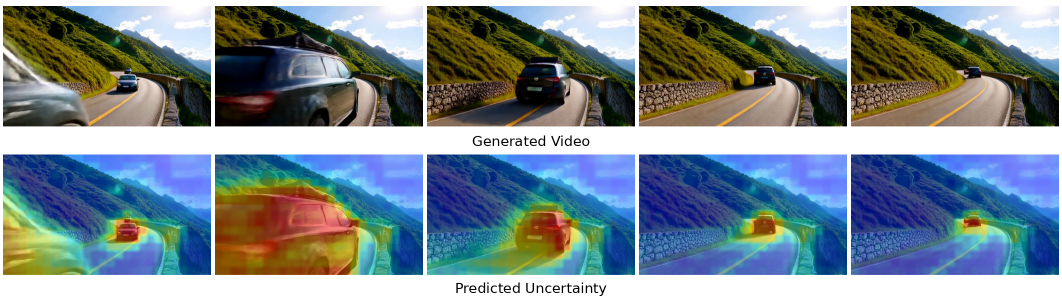


Figure 15: Example of the predicted uncertainty map at diffusion timestep $\sigma = 0.6$.

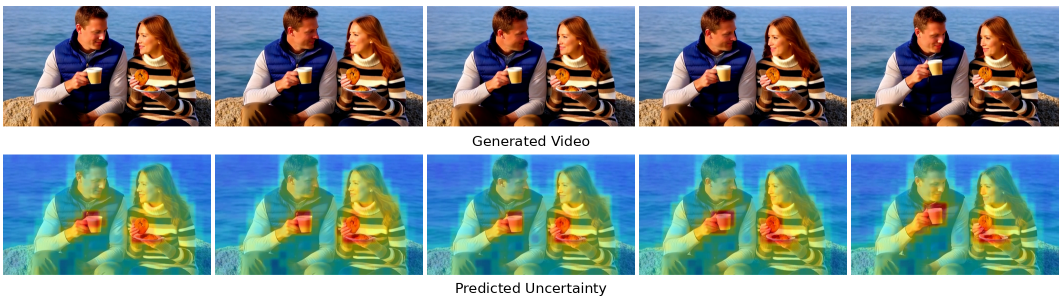


Figure 16: Example of the predicted uncertainty map at diffusion timestep $\sigma = 0.6$.