BASS: A BLOCK-WISE ASSESSMENT OF SENSITIVITY AND SELECTIVE SMOOTHING FRAMEWORK FOR QUANTIZATION OF BRANCH-CONDITIONED VIDEO DIFFUSION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video Diffusion Transformers (VDiTs) have substantially advanced high-resolution and controllable video generation, but their considerable computational and memory costs make practical deployment prohibitively expensive. Post-training quantization (PTQ) has proven to be an effective method for reducing memory usage and computational costs, yet existing methods primarily target VDiTs with a single denoising backbone and suffer from accuracy degradation when applied to branch-conditioned architectures with auxiliary control signals. Wan2.2-Animate-14B is a representative example: its auxiliary face and body adapters inject motion and identity conditioning signals into the main DiT, and its conditioning pathways exhibit quantization sensitivities that differ markedly from those of the backbone. Applying a uniform low-bit configuration across these components can therefore corrupt the conditioning signals, leading to motion degradation and identity drift. To address this problem, we propose BASS, a Block-wise Assessment of Sensitivity and Selective Smoothing framework for mixed-precision post-training quantization of branch-conditioned VDiTs. BASS identifies and protects sensitive layers across different branches, particularly those within the adapters, with minimal manual intervention. We validate BASS on advanced branch-conditioned VDiTs, using a mixed-precision policy combining W8A8 and W4A4, and show that it preserves motion fidelity, identity consistency, and overall visual quality under a mixed low-bit quantization policy. Furthermore, we implement efficient GPU kernels to achieve practical memory savings and inference speedups.

1 INTRODUCTION

Diffusion Transformers (DiTs) have become a central architecture for high-fidelity visual generation (Peebles & Xie, 2023). Their extension to video enables joint modeling of spatial content and temporal dynamics, supporting increasingly realistic video generation in models such as CogVideoX (Yang et al., 2025), HunyuanVideo (Kong et al., 2025), Wan (Wan Team, 2025), and Seedance (Seedance, 2026). Beyond text-to-video or image-to-video generation, modern video models increasingly incorporate auxiliary control pathways for pose, facial motion, reference identity, camera trajectory, or other structured signals, following the conditioning paradigm established by ControlNet (Zhang et al., 2023). Wan2.2-Animate-14B, the deployed release of Wan-Animate (Cheng et al., 2025), is representative of this branch-conditioned design: a main DiT backbone performs denoising, while a motion adapter and a face adapter process motion and face conditions and inject additional features into the backbone.

Despite their success in video generation, modern VDiT models require prohibitive computational cost. This bottleneck motivates the study of quantization, particularly *post-training quantization* (PTQ), which reduces weight and activation precision without retraining. PTQ has been extensively studied for image diffusion models (Li et al., 2023; He et al., 2023; Shang et al., 2023; Wu 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.

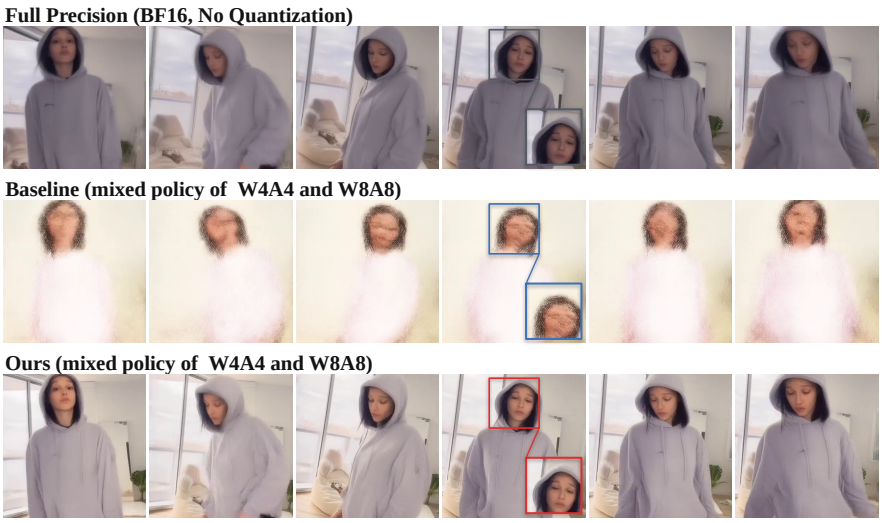


Figure 1: **Qualitative comparison on a TikTok (Jafarian & Park, 2021) dataset example using Wan2.2-Animate-14B (Cheng et al., 2025).** All methods use the same conditioning inputs and sampling configuration, and each row shows the same video frames. The enlarged regions highlight facial details: the mixed-precision baseline introduces severe structural degradation, whereas BASS remains close to the BF16 reference under mixed-precision quantization.

2024) and more recently for video diffusion transformers (Zhao et al., 2025; Feng et al., 2025). However, quantizing branch-conditioned video diffusion models is not equivalent to uniformly quantizing a conventional transformer. First, the main backbone and auxiliary adapters perform different semantic roles. Quantization in the backbone primarily affects denoising features, whereas quantization in the adapter can directly distort motion and identity conditions before they enter the residual stream. Their relative sensitivity should therefore be measured, not assumed. Second, the activation distribution changes over the denoising steps. A layer that is assumed to tolerate low-bit quantization in a late, low-noise stage may be unsafe in an early, high-noise stage. Third, attention projections interact through the softmax and residual pathways. The layer-local reconstruction error used by previous work consequently provides an incomplete estimation of the error that propagates through DiT blocks.

We address these challenges with BASS, a mixed-precision post-training quantization framework. During offline profiling, BASS adopts a three-stage paradigm. In Stage 1, BASS collects representative weight and activation statistics for higher or lower precision quantization from several samples. In Stage 2, BASS partitions the layers according to their structural roles. Each structural group is evaluated separately in every reachable timestep bucket, yielding timestep-conditioned quantization candidates for both the DiT backbone and the adapters. For activation quantization, we design a criterion for nominating candidates for channel-wise activation smoothing in the style of SmoothQuant (Xiao et al., 2023), whose effectiveness is evaluated rather than assumed. BASS then measures the sensitivity of each sampled candidate by quantizing only that group and comparing the output of its enclosing block with the reference output, thereby capturing the effects propagated through DiT modules. In Stage 3, BASS combines the resulting sensitivity estimates with execution-cost savings and employs a top-K risk solver to assign lower-bit quantization to candidates with favorable risk–saving trade-offs while retaining sensitive candidates, including vulnerable adapter layers, at a relatively higher precision. During online inference, BASS applies the offline-optimized precision assignment and smoothing parameters without online sensitivity estimation or policy optimization. Our contributions are summarized as follows:

1. We conduct extensive analysis of complex video diffusion models with auxiliary adapters and identify three major quantization challenges: heterogeneous sensitivity between the DiT backbone and conditioning adapter, step-dependent shifts in activation distributions, and non-local quantization errors propagated through attention and residual interactions.

2. We propose BASS, a mixed-policy post-training quantization framework that adaptively allocates quantization configurations across structural groups, different branches and denoising steps. BASS is tailored to address the three challenges above.
3. We demonstrate the effectiveness of BASS by conducting extensive experiments on several branch-conditioned models and implement efficient GPU kernels to achieve memory savings and acceleration in practical use.

2 Related Work

2.1 Diffusion Transformers for Video Generation

Diffusion models have become the dominant paradigm for video generation. Early work such as VDM (Ho et al., 2022) extends denoising diffusion probabilistic models to the temporal domain with a factorized space-time U-Net, and Stable Video Diffusion (Blattmann et al., 2023) shows that curated large-scale data is as decisive as architecture for latent video diffusion. The introduction of transformer backbones, notably DiT (Peebles & Xie, 2023), replaced the U-Net with a vision transformer operating on latent patches. This shift gave rise to Video Diffusion Transformers (VDiTs), adopted by CogVideoX (Yang et al., 2025), HunyuanVideo (Kong et al., 2025), Movie Gen (Polyak et al., 2025), and the Wan family (Wan Team, 2025). Despite their success, video generation requires high computational costs, demanding improvements in efficiency.

A parallel line of work adds auxiliary control branches for controllable generation. Following ControlNet (Zhang et al., 2023) for image diffusion and adapter-based transfer (Hu et al., 2022), human-animation models route pose, identity, and appearance conditions through dedicated modules rather than retraining the base denoiser: Animate Anyone (Hu et al., 2024) uses a reference-appearance branch with pose guidance, UniAnimate (Wang et al., 2024) unifies reference and pose conditioning in a single video diffusion backbone, and Wan-Animate (Cheng et al., 2025) adds adapters that inject skeletal motion and facial expression into the main DiT backbone. This branch-conditioned design offers flexible control, but it also produces architectural heterogeneity: different paths with different parameter counts and semantic roles now share one inference budget.

2.2 Post-Training Quantization for Diffusion Transformers

Post-training quantization quantizes a pretrained model by reducing the numerical precision of weights and activations without training. Diffusion models complicate the standard PTQ recipe because the activation distribution is non-stationary along the denoising steps, and the field has converged on timestep awareness as the response. PTQ4DM (Shang et al., 2023) is among the first to apply PTQ to diffusion models, sampling calibration states across timesteps. Q-Diffusion (Li et al., 2023) introduces shortcut-splitting reconstruction for the concatenated skip connections of latent diffusion U-Nets. A more recent line of work applies PTQ to Diffusion Transformers. PTQ4DiT (Wu et al., 2024) addresses salient channels and timestep-dependent activation. Q-DiT (Chen et al., 2024) combines fine-grained group quantization with automatic granularity allocation for DiTs. MPQ-DM (Feng et al., 2024) performs intra-layer mixed-precision allocation. SVDQuant (Li et al., 2025a) absorbs difficult weight outliers into a low-rank branch to reach 4-bit diffusion inference. For video specifically, ViDiT-Q (Zhao et al., 2025) pairs dynamic per-token activation quantization with DiT-tailored transformations, Q-VDiT (Feng et al., 2025) adds token-aware compensation and distillation for video-generation DiTs, and DVD-Quant (Li et al., 2025b) removes the calibration-data requirement altogether. While these methods successfully improve video diffusion transformers, applying them to more complex models with auxiliary control branches results in degradation of visual performance.

2.3 Video Generation Quality Metrics

Video generation can be evaluated from multiple aspects, so we introduce many metrics accordingly. We adopt VBench (Huang et al., 2024), a benchmark suite designed for video generative models. VBench decomposes video quality into fine-grained dimensions spanning frame-wise quality (aesthetic quality, imaging quality) and temporal consistency (subject consistency, background consistency, temporal flickering, motion smoothness). The multi-dimensional design matters for

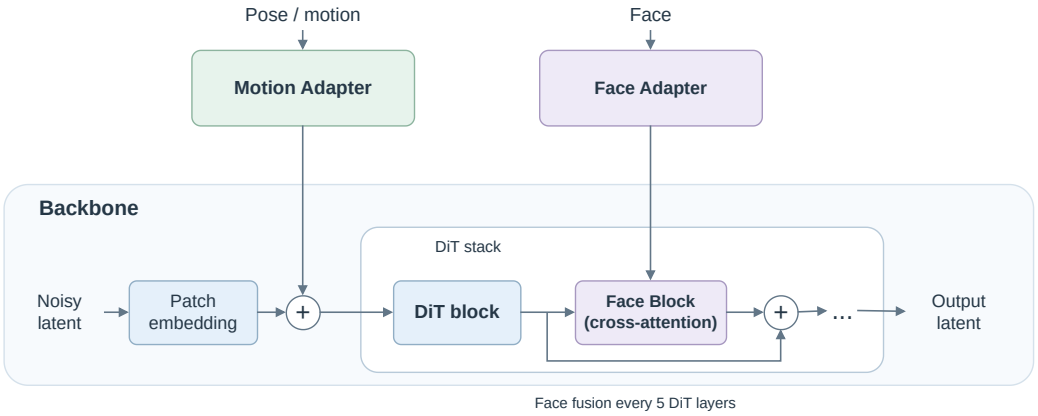


Figure 2: **The overall structure of Wan2.2-Animate-14B (Cheng et al., 2025).**

quantization studies, because different compression strategies influence different quality aspects. Additionally, we report paired reference-based metrics PSNR, SSIM, and LPIPS (Zhang et al., 2018) for frame-level agreement, together with distribution-level FVD (Unterthiner et al., 2019). These metrics assess quantization quality from multiple perspectives.

3 Preliminaries and Analysis

3.1 Branch-Conditioned Video Diffusion Transformers

Figure 2 shows the overall structure of Wan2.2-Animate-14B (Cheng et al., 2025), a typical branch-conditioned video diffusion model. It contains a main backbone and two adapter branches—a body adapter and a face adapter—that inject motion and facial features into the main backbone. For notational simplicity, we treat all auxiliary adapter branches collectively as a single adapter branch, distinct from the main DiT backbone. Let F_θ denote a pretrained video denoiser with a main DiT path D and an auxiliary control path A. At denoising step t, the backbone updates its hidden state using both its own transformation and features produced by the adapter:

$$\mathbf{h}_t^{(i+1)} = D_i\Big(\mathbf{h}_t^{(i)}, \mathbf{c}\Big) + \Gamma_i A_i\Big(\mathbf{h}_t^{(i)}, \mathbf{u}\Big), \tag{1}$$

where $\mathbf{c}$ is the semantic condition, $\mathbf{u}$ contains pose, face, or reference controls, and Γ_i denotes the model-specific injection operator. Since adapter features enter the backbone repeatedly, adapter quantization error influences both control fidelity and every downstream backbone block. Although the adapter part is relatively smaller than the DiT backbone, small size does not imply lower sensitivity to quantization. Figure 1 contains the quantization results of baseline ViDiT-Q (Zhao et al., 2025) under mixed-precision policy of W8A8 and W4A4 on Wan2.2-Animate-14B. Without careful analysis of sensitivity, applying low-bit quantization to adapter branches leads to substantial degradation in visual details. So the relative sensitivity of D and A has to be measured under a carefully designed metric rather than assumed in either direction.

3.2 Quantization Error

For a tensor $\mathbf{x}$, q-bit quantization is

$$Q_q(\mathbf{x}; s) = s \cdot \mathrm{clip}\big(\mathrm{round}(\mathbf{x}/s), -2^{q-1}, 2^{q-1} - 1\big), \tag{2}$$

where $\mathrm{clip}(z, a, b) = \min\{\max\{z, a\}, b\}$ restricts z to $[a, b]$, and round denotes rounding to the nearest integer. Let $q_- = -2^{q-1}$, $q_+ = 2^{q-1} - 1$, and $c_s(x_i) = \mathrm{clip}(x_i, sq_-, sq_+)$ be the value clipped to the quantization range. The error can be written as:

$$x_i - Q_q(x_i; s) = \underbrace{\big(x_i - c_s(x_i)\big)}_{\text{clipping}} + \underbrace{\left(c_s(x_i) - s\,\mathrm{round}\left(\frac{c_s(x_i)}{s}\right)\right)}_{\text{rounding}}. \tag{3}$$

Setting a proper scale $s = (\max_i |x_i|)/(2^{q-1} - 1)$ avoids the clipping error. So the rounding error reflecting the variation of data within a group is the major source of quantization error. The practical problems are how to choose proper groups and how to reduce the variation of a group. For quantization of weights, considering hardware implementation, relying solely on offline profiling without grouping introduces extra computation costs for the casting between data which should be calculated together. Activations, on the other hand, vary widely during inference and are harder to quantize. SmoothQuant (Xiao et al., 2023) uses smoothing to balance the difficulty of weight and activation quantization. However, ViDiT-Q (Zhao et al., 2025) points out the effectiveness of smoothing is sensitive to the choice of factors, and that using the same factors across denoising steps can adversely affect weight quantization. Therefore, it is necessary to evaluate the risks of applying smoothing.

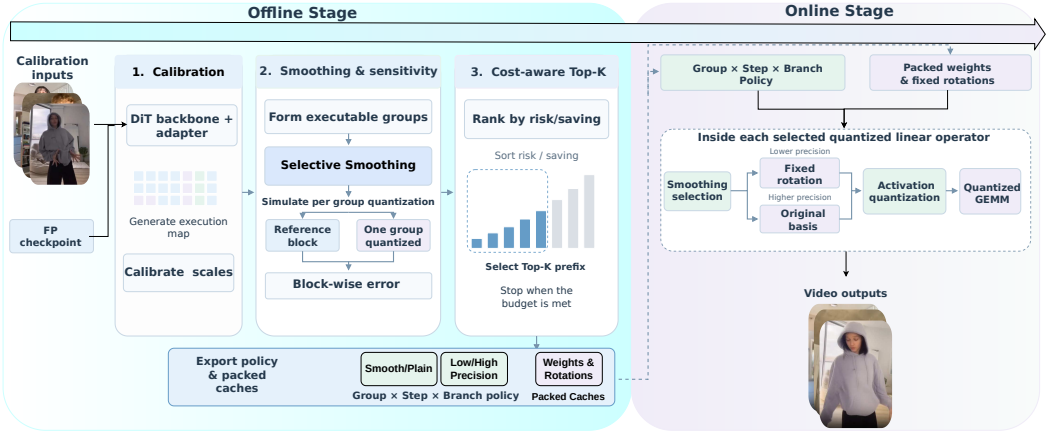


Figure 3: **The overall framework of BASS.** Offline stage (left) is composed of three parts. First, we construct a realistic execution map and calibrate channel statistics. Second, we conduct careful analysis on whether to use smoothing and the sensitivity of our defined groups. Finally, we select groups for low precision quantization by a cost-aware top-K mixed-precision policy. The exported policy and packed weight caches then guide online stage (right), where each quantized linear operator follows the offline policy and only conducts dynamic per-token activation quantization without online sensitivity estimation or policy search.

4 METHOD

Motivated by the aforementioned analysis, we present BASS. Specifically, BASS adopts a three-stage paradigm during offline profiling and conducts online inference with negligible additional hardware costs. **Stage 1** (Section 4.1) constructs a realistic execution map and calibrates channel statistics. **Stage 2** (Section 4.2) places backbone and adapter groups in a unified candidate space and measures each candidate by quantizing it at a lower precision and calculating the error at its enclosing block output. **Stage 3** (Section 4.3) combines these sensitivity estimates with execution-weighted savings, and uses a top-K risk solver to obtain a realizable policy. During online inference (Section 4.4), BASS applies the precision assignment optimized offline and smoothing selection. An overview of our proposed BASS framework is shown in Figure 3.

4.1 STAGE 1: EXECUTION-AWARE CALIBRATION AND STATISTICS PREPARATION

Section 3.2 identifies activation variation across denoising steps as a central difficulty. Rather than representing this variation with a finite table of activation scales, BASS uses dynamic per-token scaling in both offline profiling and online inference. Offline calibration therefore focuses on the execution space and statistics required by the later stages.

Execution-map construction. We first trace inference under a reference precision with sampled input. The trace records each quantizable linear layer, its model execution path, and every reached

context $c = (u, b)$. Here, u indexes a bucket of adjacent denoising steps and b identifies the active model branch (DiT backbone or auxiliary adapter branch). This map defines the candidates used by the solver described in Section 4.3.

Rotation-assisted preparation. Previous studies by Ashkboos et al. (2024), Liu et al. (2025) and Zhao et al. (2025) use rotation-based methods to mitigate extreme imbalances across activation channels. BASS adopts this approach during preparation. For each lower-precision candidate layer ℓ, we generate a deterministic random sign diagonal matrix $\mathbf{D}_\ell$ and a normalized block-Hadamard matrix $\mathbf{H}_\ell$, and define

$$\mathbf{Q}_\ell = \mathbf{D}_\ell \mathbf{H}_\ell, \qquad \widetilde{\mathbf{x}}_\ell = \mathbf{x}_\ell \mathbf{Q}_\ell, \qquad \widetilde{\mathbf{W}}_\ell = \mathbf{W}_\ell \mathbf{Q}_\ell. \tag{4}$$

Here, ℓ indexes a linear layer, $\mathbf{x}_\ell$ is its row-wise input, $\mathbf{W}_\ell$ is its weight, and $\mathbf{Q}_\ell$ is a fixed orthogonal transformation over the input-channel dimension. Since $\mathbf{Q}_\ell \mathbf{Q}_\ell^\mathsf{T} = \mathbf{I}$, $\widetilde{\mathbf{x}}_\ell \widetilde{\mathbf{W}}_\ell^\mathsf{T} = \mathbf{x}_\ell \mathbf{W}_\ell^\mathsf{T}$ before quantization. Rotation-based methods help mitigate the variation of data during inference, making the analysis of sensitivity and smoothing effectiveness reliable.

4.2 Stage 2: evaluation of selective smoothing and Structured Sensitivity

Section 3.1 shows that the DiT backbone and the auxiliary adapter branch have different semantic roles, while parameter count does not determine their sensitivity. Section 3.2 further shows that smoothing can reduce activation error while increasing weight error. We therefore evaluate both branches using the same block-output criterion. We also assess the error introduced by applying smoothing to weights and activations.

Before measuring sensitivity, we first define structural groups following kernel operator boundaries. As analyzed in Section 3.2, the choice of group influences hardware execution costs. Query and key projections that participate in one attention computation form a joint group; value, attention-output, and feed-forward projections form separate groups. The same construction is applied to every reached layer in the backbone and adapter.

Selective smoothing. Motivated by activation-aware second-order reconstruction (Frantar et al., 2023; Dong et al., 2020) and SmoothQuant's difficulty-transfer principle (Xiao et al., 2023), we apply smoothing to weights and activations to reduce the difficulty of quantizing activations:

$$s_{\ell,j} = \frac{(\max_r |X_{\ell,rj}|)^\alpha}{(\max_i |W_{\ell,ij}|)^{1-\alpha}}, \qquad \mathbf{Y}_\ell = \mathbf{X}_\ell \mathbf{W}_\ell^\mathsf{T} = \big(\mathbf{X}_\ell \mathbf{S}_\ell^{-1}\big)\big(\mathbf{W}_\ell \mathbf{S}_\ell\big)^\mathsf{T}. \tag{5}$$

Eq. (5) defines the smoothing method for input channel j of layer ℓ. Here, $X_{\ell,rj}$ is channel j of calibration token row r at layer ℓ, $W_{\ell,ij}$ connects input channel j to output channel i, and $\alpha \in [0, 1]$ controls difficulty transfer. As Section 3.2 points out, the effectiveness of smoothing needs to be estimated as it harms the distribution of weights. Thus, we design an offline evaluation of error introduced by smoothing. For state $v \in \{\mathrm{plain}, \mathrm{smooth}\}$, let $\bar{\mathbf{x}}^{(v)}$ and $\bar{\mathbf{W}}^{(v)}$ denote activations and weights after the selective smoothing and the fixed rotation used in Section 4.1. At the target lower precision q, define $V_j^{(v)} = \mathbb{E}_r[(\bar{x}_{r,j}^{(v)})^2]$ and $A_j^{(v)} = \mathbb{E}_r[(\mathrm{F}_q(\bar{x}_{r,j}^{(v)}) - \bar{x}_{r,j}^{(v)})^2]$, and let $\Delta\mathbf{W}^{(v)} = \mathrm{F}_q(\bar{\mathbf{W}}^{(v)}) - \bar{\mathbf{W}}^{(v)}$. We estimate normalized propagated distortion as

$$D_\ell^{(v)} = \frac{\sum_j V_j^{(v)} \|\Delta\mathbf{W}_{:,j}^{(v)}\|_2^2 + \sum_j A_j^{(v)} \|\bar{\mathbf{W}}_{:,j}^{(v)}\|_2^2}{\sum_j V_j^{(v)} \|\bar{\mathbf{W}}_{:,j}^{(v)}\|_2^2 + \epsilon_s}, \qquad G_\ell = \frac{D_\ell^{(\mathrm{plain})} - D_\ell^{(\mathrm{smooth})}}{D_\ell^{(\mathrm{plain})} + \epsilon_s}. \tag{6}$$

Here, r indexes sampled token rows, j input channels, and F_q the deployment-matched fake quantizer. The small constant $\epsilon_s > 0$ prevents division by zero. The numerator terms of D_ℓ approximate propagated weight and activation error using smoothing. $D_\ell^{(\mathrm{plain})}$ tests whether quantization causes material distortion, while G_ℓ measures the relative net benefit of smoothing after weight-error transfer. We select and apply preferred smoothing only when $G_\ell \geq \tau_G$.

Block-wise sensitivity estimation. Starting from the higher-precision reference with dynamic activation quantization, BASS evaluates calibration samples by lowering the precision of one candidate group at a time and measuring the normalized mean squared error (NMSE) of the corresponding

transformer block output.

$$e_{s,i}^{(v)} = \frac{\|\mathbf{Z}_{s,i}^{(v)} - \mathbf{Z}_{s,i}^{\mathrm{ref}}\|_F^2}{\|\mathbf{Z}_{s,i}^{\mathrm{ref}}\|_F^2 + \epsilon_{err}}. \tag{7}$$

Here, s indexes a calibration sample, $i = (g, c)$ indexes a reached structural group in a timestep–branch context, and v a smoothing state; $\mathbf{Z}_{s,i}^{(v)}$ and $\mathbf{Z}_{s,i}^{\mathrm{ref}}$ are the candidate and reference block outputs. For each group, we identify its enclosing transformer block based on the execution map constructed in Section 4.1. The block-level output error captures softmax, projection, and residual interactions omitted by layer-local error, serving as a better metric of the candidate group's sensitivity to quantization.

We further aggregate samples conservatively as

$$R_i^{(v)} = \mu_i^{(v)} + \lambda_\sigma \sigma_i^{(v)}, \tag{8}$$

where $\mu_i^{(v)}$ and $\sigma_i^{(v)}$ are the sample mean and standard deviation of Eq. (7), and λ_σ controls the instability penalty. $R_i^{(v)}$ is the block-wise risk of applying low-bit quantization to a group, and we further use it for selecting a preferred policy in Section 4.3.

4.3 Stage 3: Cost-Aware Top-K Policy Optimization

Section 4.2 estimates the sensitivity of candidate structural groups in both DiT backbone and adapter branches, so Stage 3 needs to make a decision on where to use low-bit quantization. We treat every reached group–context pair $i = (g, c)$ as one deployable candidate. We define the risk score as

$$\rho_i = \frac{R_i}{\Delta C_i}. \tag{9}$$

Here, i indexes a reached structural group in a timestep–branch context, R_i is its conservative block-wise sensitivity from Section 4.2, and $\Delta C_i > 0$ is the reduction of bytes when group g switches from reference precision to lower precision in context c. Thus, a smaller ρ_i indicates a lower risk per unit of cost savings when choosing the group for lower-bit quantization. BASS evaluates groups from different branches under the same criterion, with minimal reliance on handcrafted rules favoring either the backbone or the adapter. Let $\rho_{(1)} \le \rho_{(2)} \le \cdots \le \rho_{(N)}$ denote the candidates sorted from the most to the least favorable trade-off. Given calibrated reference byte cost C_0 and target normalized cost τ, the required saving is $B = (1 - \tau)C_0$. We select the shortest ranked prefix that reaches this saving:

$$K^\star = \min\left\{K : \sum_{k=1}^{K} \Delta C_{(k)} \ge B\right\}, \qquad \mathcal{L} = \{(1), \ldots, (K^\star)\}. \tag{10}$$

Here, $K^\star$ is the budget-determined number of lower-precision candidates, $\Delta C_{(k)}$ is the saving of the candidate at rank k, and $\mathcal{L}$ is the final lower-precision set. All remaining candidates use the higher precision. The reference cost covers the quantizable block and adapter operators. After selecting preferred candidates, BASS generates an offline policy for use during online inference.

4.4 Online Inference: Policy-Guided Quantized Denoising

At each denoising step, the runtime maps the scheduler timestep and active branch to $c = (u, b)$. A constant-time lookup selects the cached precision and plain/smooth state for each reached group. Thus, the sensitivity discussed in Section 3.1 is handled by the offline policy; no further search or calculation of sensitivity is performed online.

For an input row $\mathbf{x}_{\ell,r}$, an optional smoothing multiplier is first applied based on the policy generated in Stage 3 (Section 4.3). A lower-precision path then applies the layer's fixed rotation, whereas the higher-precision path remains in the original coordinates. In either case, the current activation scale is

$$a_{\ell,r}^{(p)} = \max\left(\frac{\max_j |\bar{x}_{\ell,r,j}|}{q_p^+}, \epsilon_{scale}\right), \qquad \widehat{\mathbf{x}}_{\ell,r}^{(p)} = \mathrm{clip}\Big(\mathrm{round}(\bar{\mathbf{x}}_{\ell,r}/a_{\ell,r}^{(p)}), q_p^-, q_p^+\Big). \tag{11}$$

Here, r indexes a token row, j an input channel, p the selected precision, $\bar{\mathbf{x}}$ the input after the selected transformation, and $[q_p^-, q_p^+]$ its integer range. This per-token absmax scale is computed from the current inference activation.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Model and datasets. We evaluate BASS on Wan2.2-Animate-14B (Cheng et al., 2025), a complex diffusion transformer with auxiliary body and face adapters. We generate videos at a resolution of 1280×720 and save them at 24 frames per second. We choose W4A4 as candidate precision and W8A8 as reference in Section 4.2 and Section 4.3 for our experimented models. We use two real-world evaluation datasets. **Fashion UBC** (Zablotskaia et al., 2019): Diverse human motion sequences covering walking, turning, posing, and gesturing. **TikTok** (Jafarian & Park, 2021): In-the-wild video clips with diverse subjects, backgrounds, and motion patterns, representing challenging real-world scenarios. For both datasets, we use a fixed cross-identity reference pairing to generate proper data for animate generation. Each example provides the same reference image, pose sequence, facial condition, and text prompt to every method.

Metrics. As mentioned in Section 2.3, we adopt VBench (Huang et al., 2024), a benchmark suite designed for video generative models to evaluate visual performance from different aspects: subject consistency, background consistency, aesthetic quality, imaging quality, temporal flickering, and motion smoothness. VBench Average is their unweighted arithmetic mean. Additionally, we report paired reference-based metrics—PSNR, SSIM, and LPIPS (Zhang et al., 2018) for frame-level agreement, and distribution-level FVD (Unterthiner et al., 2019).

Baselines. We compare BASS with SmoothQuant (Xiao et al., 2023), methods tailored for diffusion transformer quantization including ViDiT-Q (Zhao et al., 2025) and Q-VDiT (Feng et al., 2025), and Naive W4A4 quantization. We adapt the baseline methods to the target architectures and quantization settings; implementation details are provided in Appendix A.3.

Hardware. We conduct our experiments on a single NVIDIA RTX 5090 GPU using CUDA 12.8.

Table 1: **Quality comparison on TikTok and Fashion UBC.** VBench Average is the unweighted mean of the six reported VBench dimensions. W4A4-MP denotes a mixed-precision quantization policy combining W8A8 and W4A4.

Dataset	Method	Subject Consist ↑	Background Consist ↑	Aesthetic Quality ↑	Imaging Quality ↑	Temporal Flickering ↑	Motion Smoothness ↑	VBench Average ↑	LPIPS ↓	FVD ↓	PSNR ↑	SSIM ↑
TikTok	BF16 (No Quant.)	0.9199	0.9535	0.4170	0.4976	0.9900	0.9930	0.7951	–	–	–	–
	Naive W4A4	0.9018	0.9504	0.4118	0.4337	0.9755	0.9841	0.7762	0.95	1094.00	7.82	0.17
	SmoothQuant (W4A4)	0.8120	0.9239	0.3779	0.4374	0.9611	0.9697	0.7470	0.99	1581.64	5.67	0.17
	ViDiT-Q (W4A4-MP)	0.9099	0.9460	0.4085	0.3272	0.9896	0.9902	0.7619	0.22	733.01	18.36	0.77
	Q-VDiT (W4A4-MP)	**0.9283**	0.9500	0.3591	0.2878	0.9767	0.9799	0.7470	0.69	1720.87	19.50	0.30
	BASS (Ours)	0.9101	**0.9509**	**0.4132**	**0.4530**	**0.9896**	**0.9929**	**0.7850**	**0.06**	**87.68**	**26.43**	**0.94**
Fashion UBC	BF16 (No Quant.)	0.9930	0.9823	0.4718	0.5939	0.9989	0.9965	0.8394	–	–	–	–
	Naive W4A4	0.9004	0.8886	0.4125	0.5242	0.9798	0.9871	0.7821	0.95	1460.58	6.35	0.25
	SmoothQuant (W4A4)	0.9145	0.9600	0.4444	0.3590	0.9754	0.9791	0.7721	0.73	1175.89	5.44	0.3782
	ViDiT-Q (W4A4-MP)	**0.9997**	**0.9811**	0.4337	0.5015	0.9984	0.9954	0.8183	0.04	120.44	27.46	0.94
	Q-VDiT (W4A4-MP)	0.9480	0.9413	0.4017	0.3324	0.9810	0.9835	0.7647	0.61	1710.30	19.31	0.45
	BASS (Ours)	0.9891	0.9759	**0.4711**	**0.5838**	**0.9987**	**0.9962**	**0.8358**	**0.01**	**17.04**	**32.82**	**0.97**

5.2 MAIN RESULTS

Table 1 presents our main results. BASS achieves a mixed-precision of W8A8 and W4A4. Overall, on the Fashion UBC and TikTok datasets, our proposed BASS achieves the highest VBench Average among the evaluated quantized methods. Most post-training quantization methods target video generation models with a single denoising backbone and lose accuracy on complex branch-conditioned architectures such as Wan2.2-Animate, resulting in degradation in video quality. Moreover, when we compare frames generated by all methods with those generated by full-precision inference, BASS also shows better performance in terms of LPIPS, FVD, PSNR, and SSIM metrics, demonstrating better reconstruction of frames with low-bit quantization. We also evaluate methods on more models in Appendix B.

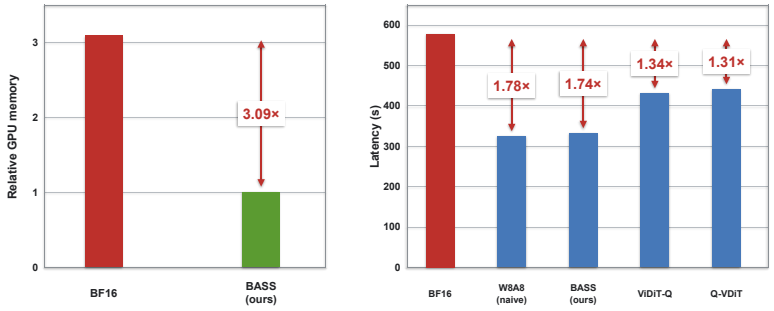


Figure 4: **The illustration of BASS's hardware resource savings.**

5.3 Hardware Efficiency

We present the memory reduction and inference speedups of BASS in Figure 4.

Memory reduction: BASS can reduce memory since offline weight quantization reduces allocated memory for storing model weights. During inference, BASS can effectively reduce peak memory allocated by 3.09x, slightly less than the theoretical value.

Inference speedup: As shown in Figure 4, using BASS with implemented kernels in Appendix A.2 achieves a 1.74x speedup during a 10-step inference. We also compare our method with ViDiT-Q (Zhao et al., 2025), Q-VDiT (Feng et al., 2025) and Naive W8A8. BASS shows an overall larger speedup compared with methods except Naive W8A8 which lacks dynamic adjustment during inference.

5.4 Ablation Studies

We conduct ablation studies to evaluate the contributions of selective smoothing and sensitivity-guided precision allocation, as shown in Table 2. **Effectiveness of smoothing selection:** We conduct experiments comparing our method with variants using no smoothing, random smoothing, or full smoothing. Replacing our selective smoothing module leads to degradation in frame-level quality, demonstrating the effectiveness of selective smoothing. **Effectiveness of sensitivity analysis:** We compare BASS using a randomly selected policy without sensitivity analysis, sharing the same ratio of W4A4 quantization. The results also show that BASS produces outputs closer to the full-precision reference.

Table 2: **Ablation studies under a fixed quantization budget.** Metrics are calculated against the full-precision inference.

Dataset	Variant	LPIPS ↓	FVD ↓	PSNR ↑	SSIM ↑
TikTok	Random Selection	0.0773	117.84	24.29	0.91
	No Smoothing	0.3370	331.78	19.64	0.32
	Random Smoothing	1.0100	435.81	7.57	0.38
	Full Smoothing	0.0827	138.49	25.43	0.92
	BASS (Ours)	**0.0609**	**87.68**	**26.43**	**0.94**

6 Conclusion and Limitations

We design BASS, a block-wise assessment of sensitivity and selective smoothing framework for mixed-precision post-training quantization of branch-conditioned VDiTs. It achieves a mixed policy quantization of W8A8 and W4A4 with little degradation in performance. We further implement CUDA kernels to achieve memory savings and inference speedups. Nevertheless, our evaluation is limited to certain branch-conditioned architectures, and the applicability of BASS to a broader range of models remains to be validated.

Reproducibility Statement

All findings presented in this paper are reproducible. Detailed information about our experiments is available in Section 5.1 and Appendix A.1. We will also release our code.

Ethics Statement

The research conducted in the paper conforms, in every respect, with the ICLR Code of Ethics.

Use of Large Language Models

In this work, we used generative AI tools to assist with translation and discovery of related work. Additionally, we used generative AI tools for editing some parts of the research paper to improve readability and editing software code. We have reviewed all AI-assisted work. We checked and modified LLM-generated translation and LLM-generated code was tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated llms. *arXiv preprint arXiv:2404.00456*, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Lei Chen, Yuan Meng, Chen Tang, Xinzhu Ma, Jingyan Jiang, Xin Wang, Zhi Wang, and Wenwu Zhu. Q-DiT: Accurate post-training quantization for diffusion transformers. *arXiv preprint arXiv:2406.17343*, 2024.

Gang Cheng, Xin Gao, Li Hu, et al. Wan-Animate: Unified character animation and replacement with holistic replication. *arXiv preprint arXiv:2509.14055*, 2025.

Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. HAWQ-V2: Hessian aware trace-weighted quantization of neural networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Weilun Feng, Haotong Qin, Chuanguang Yang, Zhulin An, Libo Huang, Boyu Diao, Fei Wang, Renshuai Tao, Yongjun Xu, and Michele Magno. MPQ-DM: Mixed precision quantization for extremely low bit diffusion models. *arXiv preprint arXiv:2412.11549*, 2024.

Weilun Feng, Chuanguang Yang, Haotong Qin, Xiangqi Li, Yu Wang, Zhulin An, Libo Huang, Boyu Diao, Zixiang Zhao, Yongjun Xu, and Michele Magno. Q-VDiT: Towards accurate quantization and distillation of video-generation diffusion transformers. *arXiv preprint arXiv:2505.22167*, 2025.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2023.

Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. PTQD: Accurate post-training quantization for diffusion models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. Animate Anyone: Consistent and controllable image-to-video synthesis for character animation. *arXiv preprint arXiv:2311.17117*, 2024.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 21807–21818, 2024.

Yasamin Jafarian and Hyun Soo Park. Learning high fidelity depths of dressed humans by watching social media dance videos. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 12753–12762, 2021.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. VACE: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598*, 2025.

Weijie Kong, Qi Tian, Zijian Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2025.

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. SVDQuant: Absorbing outliers by low-rank component for 4-bit diffusion models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025a.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-Diffusion: Quantizing diffusion models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 17489–17499, 2023.

Zhiteng Li, Hanxuan Li, Junyi Wu, Kai Liu, Haotong Qin, Linghe Kong, Guihai Chen, Yulun Zhang, and Xiaokang Yang. DVD-Quant: Data-free video diffusion transformers quantization. *arXiv preprint arXiv:2505.18663*, 2025b.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. *arXiv preprint arXiv:2405.16406*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 4172–4182, 2023.

Adam Polyak, Amit Zohar, Andrew Brown, et al. Movie Gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2025.

Team Seedance. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 1972–1981, 2023.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2019.

Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Xiang Wang, Shiwei Zhang, Changxin Gao, Jiayu Wang, Xiaoqiang Zhou, Yingya Zhang, Luxin Yan, and Nong Sang. UniAnimate: Taming unified video diffusion models for consistent human image animation. *arXiv preprint arXiv:2406.01188*, 2024.

Junyi Wu, Haoxuan Wang, Yuzhang Shang, Mubarak Shah, and Yan Yan. PTQ4DiT: Post-training quantization for diffusion transformers. *arXiv preprint arXiv:2405.16005*, 2024.

Guangxuan Xiao, Ji Lin, Mickaël Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38087–38099, 2023.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2025.

Polina Zablotskaia, Aliaksandr Siarohin, Bo Zhao, and Leonid Sigal. DwNet: Dense warp-based network for pose-guided human video generation. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, 2019.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 3813–3824, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 586–595, 2018.

Tianchen Zhao, Tongcheng Fang, Haofeng Huang, Rui Wan, Widyadewi Soedarmadji, Enshu Liu, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. ViDiT-Q: Efficient and accurate quantization of diffusion transformers for image and video generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

A Additional Experimental Details

A.1 Detailed Experimental Settings

For each model, BASS uses five samples for offline calibration. Each calibration run requests 81 frames at 1280×720 resolution and uses seed 0 and a classifier-free guidance (CFG) scale of 1.0, matching the generation configuration. For a 20-step denoising procedure, we set 4 timestep buckets, with 5 denoising steps in each bucket. For smoothing in Eq. (5), we set the migration exponent to $\alpha = 0.5$. For selective smoothing evaluation, we sample at most 512 token rows per layer and set a threshold of $\tau_G = 0$ in Eq. (6), so smoothing is selected when its estimated distortion does not exceed that of the plain configuration. We use $\epsilon_s = 10^{-12}$ to stabilize denominators, and the same for the block-level NMSE denominator in Eq. (7). Weight and activation quantization scales have a minimum positive value of $\epsilon_{scale} = 10^{-8}$. The fixed randomized rotation in Eq. (4) uses base seed 2025. For precision allocation, the risk-aggregation coefficient in Eq. (8) is $\lambda_\sigma = 1$. The Top-K objective in Eq. (10) uses a target cost ratio of $\tau = 0.85$. We also conduct an ablation study of the ratio τ in Appendix B.2.

A.2 Kernel Implementation

W8A8 execution. The optimized W8A8 path uses a CUTLASS Tensor Core GEMM with INT8 inputs and INT32 accumulation. Its epilogue multiplies the accumulator by the token scale and output-channel scale, then writes BF16 output directly. This removes the separate full-size INT32 output and rescaling pass of the unfused path. The implementation uses a $128 \times 256 \times 64$ threadblock tile and a three-stage main loop.

W4A4 execution on Blackwell. The fast path stores two signed four-bit weight codes per byte in a prepacked layout. At runtime, a CUDA kernel expands these codes to an E4M3 FP8 operand on the GPU. Activations are dynamically rounded and clipped to four-bit integer codes and written directly into an FP8 carrier, avoiding an activation packing–unpacking round trip. These small integer codes are exactly representable in E4M3: the conversion does not introduce an additional quantization step. GEMM is dispatched through the FP8 Tensor Core path with FP32 output and fast accumulation disabled, followed by token-wise and channel-wise rescaling. The runtime falls back to an INT8 carrier when the FP8 path is unavailable.

A.3 Implementation Details for Baseline Methods

We adapt the baseline methods to our experimental setting. **Naive W4A4** applies uniform precision without smoothing, rotation, or sensitivity-based allocation. **SmoothQuant** (Xiao et al., 2023) uses five held-out videos per dataset to collect channel-wise absolute maxima, with fixed $\alpha = 0.5$, per-channel W4 weights, and dynamic per-token activations. The **ViDiT-Q** adaptation (Zhao et al., 2025) uses five calibration videos and four timestep bins to determine a fixed W4A4/W8A8 policy; candidates are ranked by generation-metric sensitivity. Adapters follow the same rules, and activation scales remain dynamic. **Q-VDiT** (Feng et al., 2025) implements the original Token-aware Quantization Estimator and Temporal Maintenance Distillation.

B Additional Cross-Model and ablation experiments

B.1 Wan2.1-VACE-1.3B

Wan2.1-VACE-1.3B (Jiang et al., 2025) is a video creation and editing model that supports reference-to-video generation, video-to-video editing, and masked video editing (Jiang et al., 2025). Its Context Adapter processes visual conditions and injects additive features into the main DiT blocks. This provides a second branch-conditioned architecture with a different model scale and conditioning mechanism from Wan2.2-Animate-14B. We apply BASS to both the backbone and the VACE branch, generating new calibration statistics and a new policy. This experiment tests whether sensitivity-based allocation and selective smoothing remain useful across different branch-conditioned models.

Table 3 reports the experimental results of baselines and BASS. On both datasets, BASS achieves the highest VBench Average score, close to the BF16 reference and the best LPIPS, PSNR, FVD and SSIM among the reported quantized configurations. These results support cross-model applicability of BASS.

Table 3: **Wan2.1-VACE-1.3B results on matched Fashion UBC (Zablotskaia et al., 2019) and TikTok (Jafarian & Park, 2021)**. VBench Average is the unweighted average of the six VBench dimensions; MP denotes mixed precision.

Dataset	Method	Subject Consist ↑	Background Consist ↑	Aesthetic Quality ↑	Imaging Quality ↑	Temporal Flickering ↑	Motion Smoothness ↑	VBench Average ↑	LPIPS ↓	FVD ↓	PSNR ↑	SSIM ↑
Fashion UBC	BF16 (No Quant.)	0.9965	0.9884	0.5604	0.5728	0.9988	0.9969	0.8523	–	–	–	–
	Naive W4A4	0.9129	0.9531	0.3601	0.5093	0.9713	0.9784	0.7808	1.0405	1464.96	9.69	0.32
	SmoothQuant (W4A4)	0.8742	0.9615	0.4205	0.3967	0.9880	0.9909	0.7720	0.8631	943.74	13.38	0.48
	ViDiT-Q (W4A4-MP)	**0.9959**	0.9866	0.5521	0.5432	0.9986	0.9967	0.8455	0.1262	75.15	24.79	0.89
	Q-VDiT (W4A4-MP)	0.9953	0.9855	0.5530	0.5778	0.9986	0.9967	0.8512	0.0869	50.76	28.40	0.93
	BASS (Ours)	0.9953	**0.9873**	**0.5542**	**0.5788**	**0.9986**	**0.9968**	**0.8518**	**0.0825**	**48.03**	**29.06**	**0.94**
TikTok	BF16 (No Quant.)	0.9352	0.9202	0.5154	0.5564	0.9915	0.9944	0.8188	–	–	–	–
	Naive W4A4	0.9063	0.9551	0.3685	0.5287	0.9769	0.9816	0.7862	0.7357	2168.95	16.25	0.61
	SmoothQuant (W4A4)	0.9062	**0.9691**	0.4137	0.4239	0.9878	0.9912	0.7820	0.6890	1783.56	17.88	0.70
	ViDiT-Q (W4A4-MP)	0.9294	0.9202	0.5057	0.5191	0.9912	0.9938	0.8099	0.3414	387.00	23.25	0.84
	Q-VDiT (W4A4-MP)	0.9236	0.9332	0.5031	0.4341	**0.9931**	0.9946	0.7969	0.3459	345.87	26.81	0.87
	BASS (Ours)	**0.9308**	0.9292	**0.5080**	**0.5353**	0.9924	**0.9946**	**0.8151**	**0.1956**	**221.82**	**30.45**	**0.90**

B.2 Ablation on the Cost Budget τ

We evaluate the effect of the cost budget τ in Eq. (10) on Wan2.2-Animate-14B using the TikTok dataset. Smaller τ requires more aggressive low-precision allocation. All three settings share the calibration samples and block-wise sensitivity measurements.

Table 4: **Budget ablation on Wan2.2-Animate-14B (TikTok dataset).** VBench Average is the unweighted mean of the six VBench dimensions. LPIPS, PSNR, and SSIM use matched BF16 generations as references; FVD compares the generated-video feature distributions against BF16.

τ	VBench Average ↑	LPIPS ↓	FVD ↓	PSNR ↑	SSIM ↑
0.65	0.7453	0.68	1693.72	14.42	0.19
0.75	0.7772	0.08	205.84	24.98	0.93
0.85	**0.7850**	**0.06**	**87.68**	**26.43**	**0.94**

Table 4 shows progressively larger deviations from BF16 as the budget decreases. The 0.75 setting achieves greater compression at the cost of lower fidelity compared with 0.85, whereas 0.65 incurs severe degradation in video quality. We choose $\tau = 0.85$ according to the results of the ablation study.

C Additional Experimental Results

We provide three qualitative comparisons using different inputs from the TikTok (Jafarian & Park, 2021) dataset. Within each figure, all methods share the input and displayed frame indices and frames are extracted from video generation. These examples show appearance differences of BASS and baselines.

C.1 Wan2.2-Animate-14B: Method Comparison

Figure 5 shows the visual performance of one sample on Wan2.2-Animate-14B (Cheng et al., 2025). Naive W4A4 produces structured noise, while the ViDiT-Q and Q-VDiT adaptations show substantial loss of facial and clothing detail in this example. SmoothQuant preserves a recognizable person but changes background appearance and introduces visible texture artifacts. BASS preserves the human and background details more closely to full-precision inference.

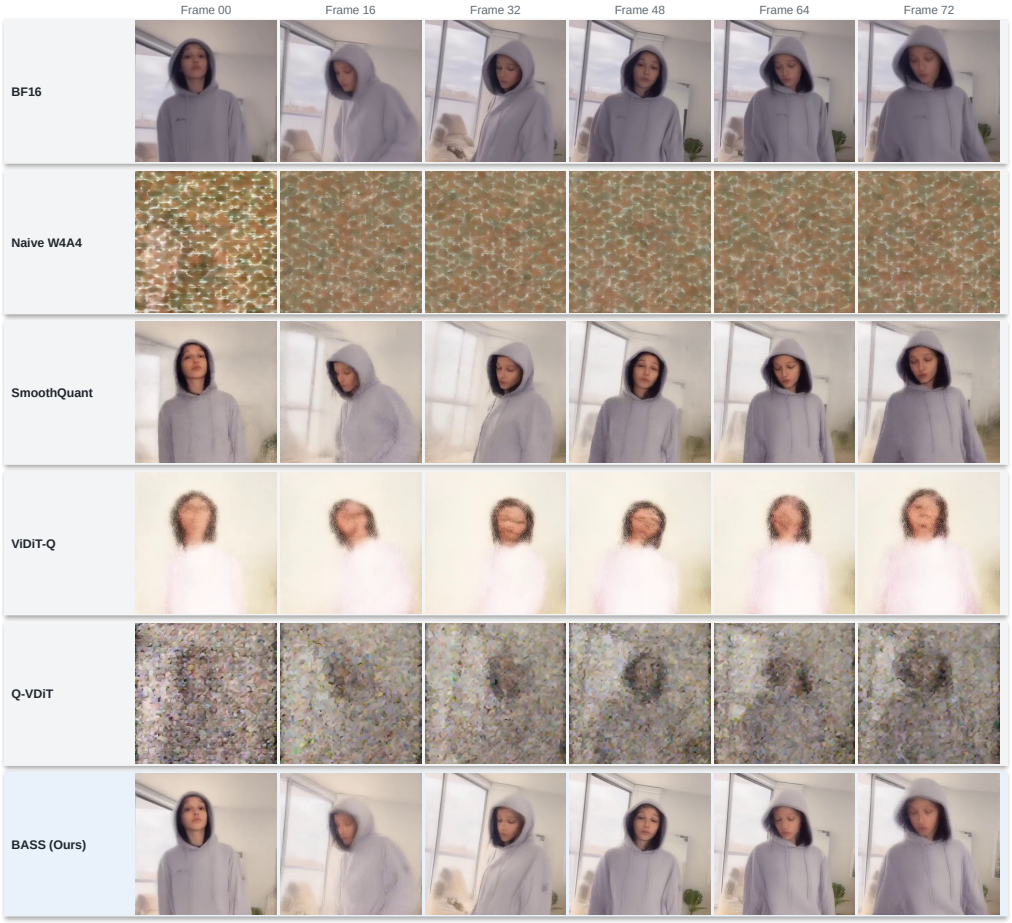


Figure 5: **Wan2.2-Animate-14B comparison on a sample from TikTok.** Columns show frames 0, 16, 32, 48, 64, and 72. SmoothQuant uses W4A4; ViDiT-Q, Q-VDiT, and BASS use their respective mixed-precision policies.

C.2 WAN2.2-ANIMATE-14B: ABLATION COMPARISON

Figure 6 shows the visual performance of ablation study in Section 5.4. The no-smoothing variant collapses into noise, whereas random smoothing recovers the subject but leaves pronounced texture artifacts. Random selection and full smoothing preserve recognizable content but exhibit local appearance differences and blurring. BASS retains the subject, clothing, and indoor scene without the severe corruption.

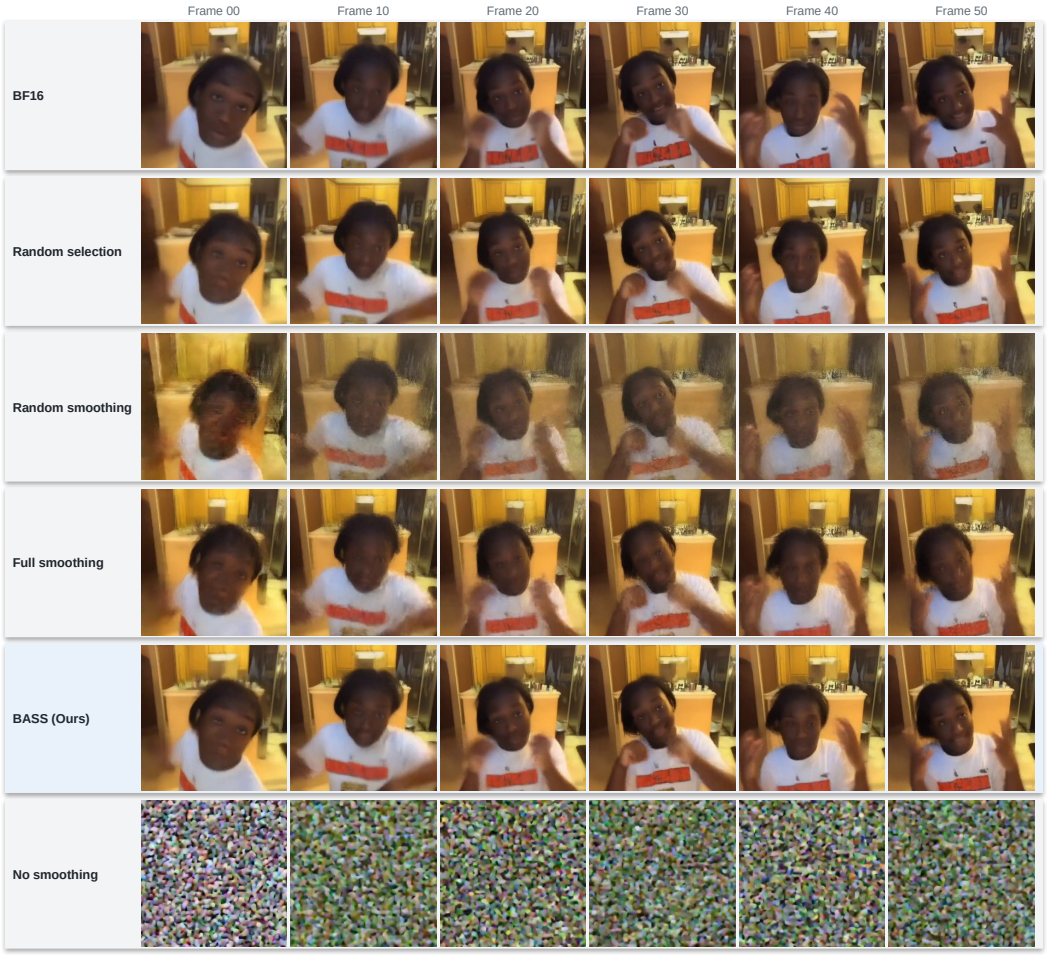


Figure 6: **Ablation comparison on a TikTok sample with frame indices 0, 10, 20, 30, 40, and 50.**

C.3 WAN2.1-VACE-1.3B: METHOD COMPARISON

Figure 7 shows the visual performance of one sample on Wan2.1-VACE-1.3B (Jiang et al., 2025). SmoothQuant W4A4 fails to recover a recognizable scene in this example. ViDiT-Q, Q-VDiT, and BASS retain the person and outdoor composition, with differences in lighting, texture, and fine detail. BASS preserves the visible pose sequence and background structure, while ViDiT-Q misses background details and Q-VDiT introduces more blurring.

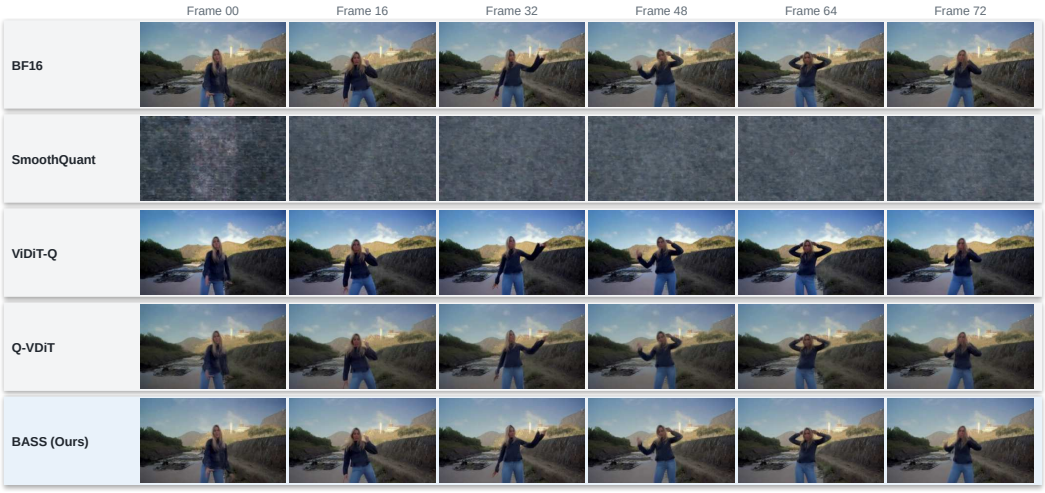


Figure 7: **Wan2.1-VACE-1.3B comparison on a sample from TikTok.** The columns show full frames at indices 0, 16, 32, 48, 64, and 72. The five rows show BF16, SmoothQuant W4A4, the ViDiT-Q and Q-VDiT adaptations, and BASS.