

A Appendix

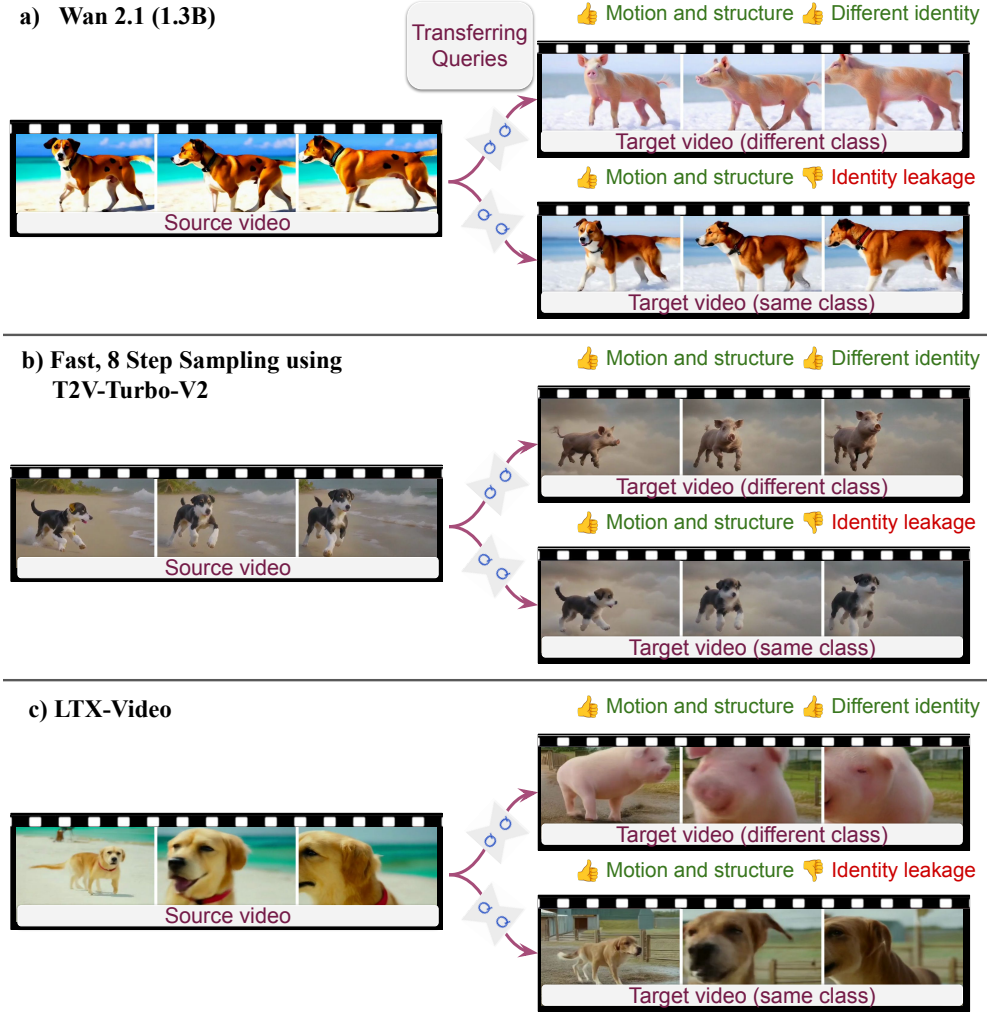


Figure 8: [\(click-to-view-online\)](#) Identity Leakage and Motion Transfer with Other Text-to-Video Models.

A.1 Motion Transfer Implementation Details

- We record and inject only the Q features from the conditional forward pass.
- When recording Q features, we use the same noise seed across different noise levels. During generation, however, we start from a random initialization.
- In VideoCrafter2, to ensure consistency with the video model’s latent space, we follow



Figure 9: **Limitations.** The source subject shape may affects the target object.

[7]: we add minor noise to the source video latent (two steps) and then denoise it back to a clean latent before extracting Q features.

- For quantitative results, we used Q-injection ending at $t = 600$. For qualitative results with VideoCrafter2, we selected videos with Q-injection ending between $t = 800$ and $t = 600$: closer to $t = 800$ for camera motion, and closer to $t = 600$ for non-rigid movement.

A.1.1 DiT-Based Models

- For both WAN and LTX-Video, we used a 50-step flow-matching denoising scheduler as described by Esser et al. [4]. This schedule shifts the timestep allocation so that more steps are concentrated in the high-noise region. Specifically, we used the FlowMatchEulerDiscreteScheduler from huggingface, with their default $\mu = 3.065$ hyperparam.
- Similar to VideoCrafter2, to transfer the full magnitude of motion we had to inject Q features for a substantial amount of steps. For WAN, we injected Q features for 60% of the denoising schedule; for LTX-Video, we used 40%.
- In WAN, we inject Q features only in layers 20–30. In all other models, Q-injection is applied to all layers.
- Current motion transfer benchmarks consist of short 16- or 24-frame videos, which is significantly shorter than the standard length of WAN 2.1 videos (81 frames). Therefore, for motion transfer in WAN, we repeat each frame twice and pad the last frame one additional time, therefore mapping $16 \rightarrow 33$ and $24 \rightarrow 49$. Accordingly, we double the frame rate of the generated videos to keep the duration the same as in the source video.
- For LTX-Video only, we found that injecting Q features between different initial noise seeds preserved the findings about identity leak but introduced visual artifacts. To mitigate this, we used identical seeds for different prompts and matched the global statistical moments of the Value features, improving compatibility with the injected Query features.

A.2 Background: Self-Attention in T2V models

Our method manipulates the activations of the spatial self-attention in T2V diffusion models. We start by outlining its mechanism and introducing key notations.

Recent T2V diffusion models are based on a latent video diffusion model (LVDM) architecture where a U-Net denoiser is trained to estimate the noise in the noisy latent codes

input. The denoising U-Net is a 3D U-Net architecture consisting of a stack spatio-temporal blocks comprised of convolutional layers, spatial transformers (ST), and temporal transformers (TT). The ST operate independently on each video frame, without awareness of the temporal structure, while the TT operate independently on each temporal patch, without awareness of the spatial structure. In this work, we focus on manipulating the self-attention mechanism of the spatial transformer layers.

B Consistent Video Generation - Supplementary Details

B.1 Notations

Our method manipulates spatial self-attention activations in T2V diffusion models. We denote by $\{Q, K, V, O\}$ the respective Query, Key, Value and Output features of a single self-attention layer (see Appendix A.2 for background). In our method, these features interact across frames, enabling cross-frame attention and consistency. We denote by Q_v the Q features of a layer during a “vanilla”, non-consistent, forward pass in a pretrained network, Q_c the query features from our subject-consistent model, and Q_f as the flow-based query features. For brevity, we omitted the frame index i

B.2 ConsiStory details

ConsiStory [49] operates in three steps. **(1) Subject-Driven localization with extended Self-Attention (SDSA)** – localizes the subject across a set of noisy generated images by aggregating cross-attention maps across layers and timesteps. To ensure subject consistency, SDSA enables each image to attend to patches of the main subject present in *other* image frames. This is done by extending the self-attention mechanism, allowing it to share K, V features of the subject between multiple images. Unfortunately, SDSA alone diminishes *layout* diversity in the generated images. Therefore, **(2) Layout Diversity** – reinforces diversity through two techniques: First, it incorporates Q features from a vanilla, *non-consistent* sampling step. Second, it applies an inference-time dropout to the shared K, V features. Finally, **(3) Refinement Injection** – improves consistency in finer details by injecting the O features between corresponding subject patches.

The pipeline is illustrated in Fig. 10.

B.3 Framewise Subject-Driven Self-Attention

Our first step builds on the Subject-Driven Self-Attention (SDSA) mechanism [49] to incorporate subject features across multiple video shots by extending the self-attention mechanism. We identified two critical challenges when adapting SDSA to video generation: (1) reliably localizing the subject during video denoising, and (2) ensuring motion fluidity is not compromised.

For subject localization, we propose using the estimated clean image \hat{x}_0 for mask generation instead of relying on internal network activations, ensuring reliable masks even in early denoising steps. For motion fluidity, we introduce a framewise attention scheme, where frames with matching temporal indices across shots selectively attend each other. This prevents artifacts and frozen motion.

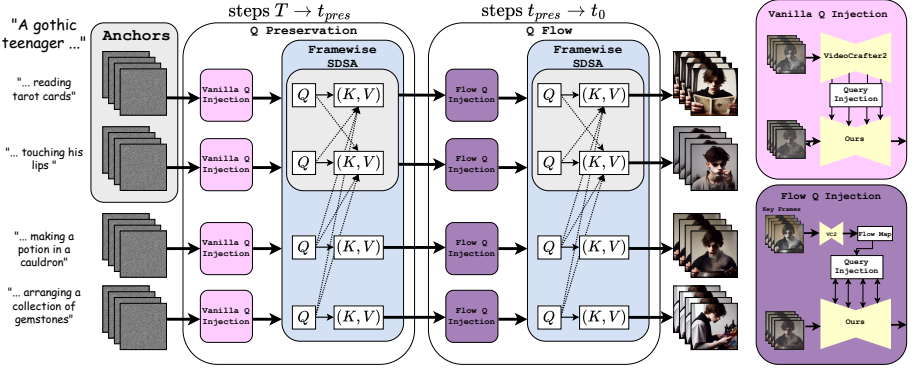


Figure 10: Video Storyboarding Architecture: Our consistent denoising process has two phases: Q Preservation and Q Flow. We first generate and cache video shots using “vanilla” VideoCrafter2. In Q Preservation ($T \rightarrow t_{pres}$), we use Vanilla Q Injection to maintain motion structure by replacing our Q values with vanilla ones. In Q Flow ($t_{pres} \rightarrow t_0$), we use a flow map from vanilla key frames to guide Q feature injection. This phase maintains character identity by allowing the use of Q features from our consistent denoising process, while the flow map ensures that these identity-preserving features are applied in a way that’s consistent with the original motion. Throughout, we employ two complementary techniques: framewise subject-driven self-attention for visual coherence, and refinement feature injection (section B.4) to reinforce character consistency across diverse prompts.

We term this component Framewise-SDSA. Further technical details, including the mask estimation process and the formal definition of Framewise-SDSA, are provided in Appendix B.10..

When generating multiple video shots with consistent subjects, we face a fundamental trade-off between subject consistency and motion quality. Our experiments show that while Framewise-SDSA improves subject consistency, it often results in side-effects, leading to excessive synchronization of motion layout across video shots and introduces motion artifacts (Fig. 7(4th row)). These artifacts arise from the model’s attempt to simultaneously satisfy both the text prompt and the undesired synchronization across shots.

Prior work in ConsiStory (Sec. B.2) demonstrated success in maintaining layout diversity for image generation through SDSA dropout and query injection. However, our experiments show that directly extending this approach to video generation produces poor results, with significant visual artifacts and compromised consistency between shots (Fig. 12). This likely occurs because (1) ConsiStory’s query injection is applied for shorter periods compared to the amount required in video models, and (2) since ConsiStory’s vanilla-network queries are derived from latents that are influenced by consistency-preserving mechanisms in earlier steps, rather than following an independent denoising trajectory.

Our analysis (Fig. 7) reveals that query features encode both motion patterns and subject identity. Injecting only vanilla query features (Q_v) preserves dynamic motion but results in inconsistent subjects across shots (row 3). Conversely, using only subject-consistent query features (Q_c) ensures subject consistency but produces rigid, unnatural, and synchronized movements (row 4). This observation motivates our two-phase approach that leverages both

feature types.

Phase 1: Motion Structure Establishment. In early denoising steps ($t \in [T, t_{\text{pres}}]$), we focus on establishing a robust initial motion structure using a process we call Q Preservation (see Algorithm 2). During this phase, we directly inject vanilla query features (Q_v) from pre-generated video shots. This allows us to retain the motion patterns present in the vanilla videos. Without this initial phase, later denoising steps may deviate from the original motion patterns, leading to degraded motion quality.

Phase 2: Flow-based Consistency Integration. As denoising progresses (beyond t_{pres}), subject consistency becomes increasingly important. To address this, we introduce Q Flow (see Algorithm 2), a technique inspired by TokenFlow [24], where flow-based query features (Q_f) are injected to incorporate subject-consistent information while preserving the original motion. Similar to [24], in this phase, we derive a flow map from vanilla-generated keyframes (Q_v), which provides the motion structure. We then blend subject-consistent query features (Q_c) from nearby frames, as dictated by the flow. This blending process produces Q_f that adhere to the original motion patterns while maintaining subject consistency across frames. Finally, we use a linear schedule ($\alpha = 0.25 + t/1000$) to gradually transition from flow-adjusted to subject-consistent features.

By following this approach, we maintain the natural flow of motion established in Phase 1 and progressively integrate subject-consistent features without sacrificing motion quality. The formal definition of our flow-based query injection process is provided in Appendix B.11.

B.4 Refinement Feature Injection for Enhanced Consistency

Despite improved motion preservation and subject consistency, fine details in subject appearance can still vary across frames. We address this by adapting the refinement feature injection technique.

However, naively applying refinement feature injection solely to the conditional denoising step, as in ConsiStory, introduces unnatural motion artifacts. This is likely due to the conditional step uses a correspondence map to inject features from different frames, while the unconditional step does not, resulting in inconsistent feature injection. To mitigate this, we extend refinement feature injection to the unconditional denoising step, using the same DIFT correspondence map. We also utilize the entire frame set of each anchor video for refinement injection. This synchronized approach improves overall consistency and reduces motion artifacts. For qualitative results, see Fig. 12.

B.5 Consistent Video Generation - Comparisons to Baselines

We compare *Video Storyboarding* with strong baselines, starting with a qualitative comparison that shows improved subject-consistency and better motion-alignment. We then conduct an ablation study to examine how self-attention query (Q) tokens affect motion and identity, highlighting the contributions of the components in our method. Finally, quantitative evaluation follows, including a large-scale user study, which demonstrates that users typically favor our results.

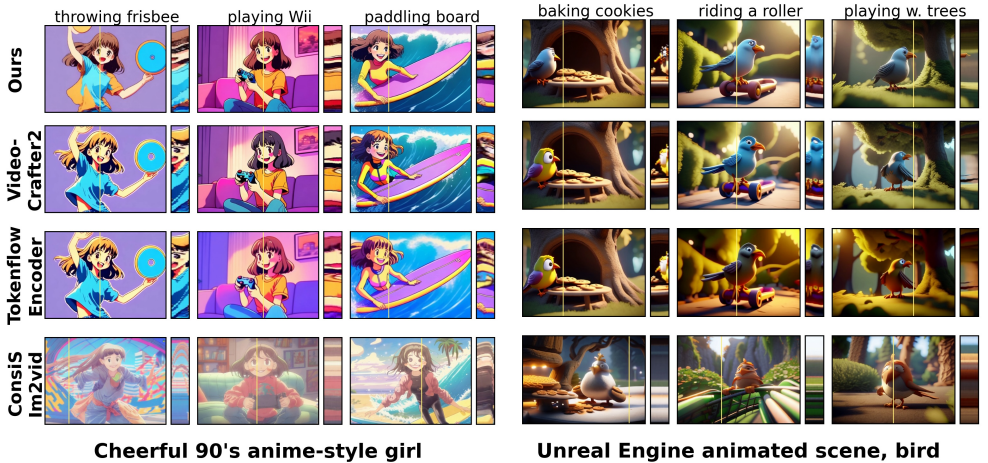


Figure 11: **Qualitative Comparisons.** The first frame of each video shot is displayed along with a spatiotemporal y - t slice to visualize motion. *Ours* (top row) shows improved character consistency across shots while maintaining natural motion. *VideoCrafter2* (row 2) is the vanilla model, showing diverse motion but inconsistent characters. *Tokenflow-Encoder* (row 3) preserves original motion but struggles with character consistency and introduces coloring artifacts. *ConsiS Im2Vid* (bottom row) fails to maintain consistency and exhibits limited motion adherence to prompts. See more examples in Fig. 16.

B.6 Evaluation baselines

We compare our method to several baselines: (1) **VideoCrafter2:** A baseline “vanilla” text-to-video model, without adaptations. VideoCrafter2 is a public SoTA video model [25]. (2) **Tokenflow-Encoder:** A combination of TokenFlow [20] with IP-Adapter, a Personalization-Based Encoder [69]. We personalize TokenFlow by conditioning the IP-Adapter on the first frame of one video generated by the vanilla model. For IP-Adapter we use a high-scale hyper-parameter to push the model toward stronger consistency. (3) **ConsiS Im2Vid:** A combination of SoTA *image*-consistency approach [49], with a subsequent Image-to-Video variant of VideoCrafter. First, we generate a set of consistent *reference* images. Then, we use them as inputs to an Image-to-Video model. We chose VideoCrafter, as it is a public image-to-video model that has an overall quality equivalent to that of the text-to-video VideoCrafter2 model according to the VBench benchmark [25]. (4) **VSTAR:** A method for generating a long video with dynamic evolution [61]. We directly provide the multiple prompts and sample 16 frames per prompt, then splitting the result into individual shots. (5) **Turbo-V2:** A recent state-of-the-art text-to-video model [80] that we use to demonstrate our method’s adaptability to other architectures.

B.7 Qualitative Results

To visually assess both multi-shot consistency and motion quality in videos, we present two elements per video shot: the initial frame for comparing consistency between shots, and a spatiotemporal slice of the space-time volume, termed “ y - t slice” [12], to visualize motion

quality. The selected column for the y - t slice is marked by a yellow line. Typically, we choose the column with the maximum variance in the vanilla-generated video shot. Occasionally, we manually select the y - t column to highlight specific motion characteristics. For ConsiS Im2Vid, the max-variance column is chosen independently, as it does not directly correspond to the vanilla model. The videos for the consistent multi-shot generation results are available online at https://research.nvidia.com/labs/par/video_storyboarding/.

In Fig. 11 and Fig. 16, we showcase qualitative comparisons between our approach, the vanilla model, and the baselines. Our method demonstrates the ability to alter subject identities consistently across shots, while guiding them towards a unified appearance. This consistency is evident when comparing image frames from different shots. Additionally, an examination of the y - t motion slices reveals that our approach successfully adheres to the motion guided by the vanilla model.

The Tokenflow-Encoder baseline preserves the original motion from vanilla models while primarily affecting the color palette and color style of objects and scenes in videos. However, its impact on the identity of the subject is less pronounced than our approach. Additionally, the combination with a high-scaled IP-Adapter often degrades video quality, causing blurring and color artifacts. See the bird example in Fig. 11 (3rd row) and the boy in Fig. 16 (3rd row).

The ConsiS Im2Vid baseline maintains consistency in its *reference* images. However, the subsequent image-to-video model introduces certain limitations. It lacks awareness of the consistency requirement and the capability to maintain it, causing the subject identity to vary between video shots. Although consistency is maintained within each shot, overall consistency with the reference image is compromised, as seen in the bird example in Figure 1 (4th row). Additionally, the image-to-video model fails to account for the action specified in the text prompt. This results in either minimal motion or movement that does not align with the prompt, as the model relies solely on the conditioning image and cannot effectively utilize the textual information. See the limited motion in the y - t slices in Fig. 11 (4th row) and the corresponding videos in the supplemental material.

VSTAR (Fig. 16, Appendix) produces large motion dynamics, but struggles with prompt control, often resulting in entire videos misaligning with text descriptions. As it maintains consistency through continuous video generation, it better suits scene transitions than independent shots.

When applied to Turbo-V2 (Fig. 15), our method enables subject consistency while leveraging Turbo-V2’s enhanced motion capabilities.

Adapting ConsiStory for Video Generation. Next, we demonstrate the challenges of adapting the image-based ConsiStory algorithm [49] to video generation. Fig. 12 (3rd row “ConsiS”) shows a naive implementation of ConsiStory with subject-driven extended attention coupled across all frames in each video shot, using subject mask dropout and omitting feature injections to the unconditioned diffusion pass. At each step, it also employs queries influenced by the consistency-preserving mechanism of previous steps, rather than queries from an independent vanilla denoising process. This results in impaired identity consistency, strong motion artifacts, and unnatural motion flow of different body parts for both the rabbit and monster examples. Adding feature injection to the unconditional feature denoising (4th row “ConsiS +Uncond”) resolves motion artifacts but largely reduces motion magnitude (*e.g.* body postures are mostly frozen), and compromises identity. Next, coupling each frame in a shot with a single frame in an anchor video and avoiding SDSA dropout (5th row “Q ConsiS”) allows for subtle natural motion, although it remains partially synchronized. It also improves identity preservation to some degree. Unlike ConsiStory, SDSA dropout in

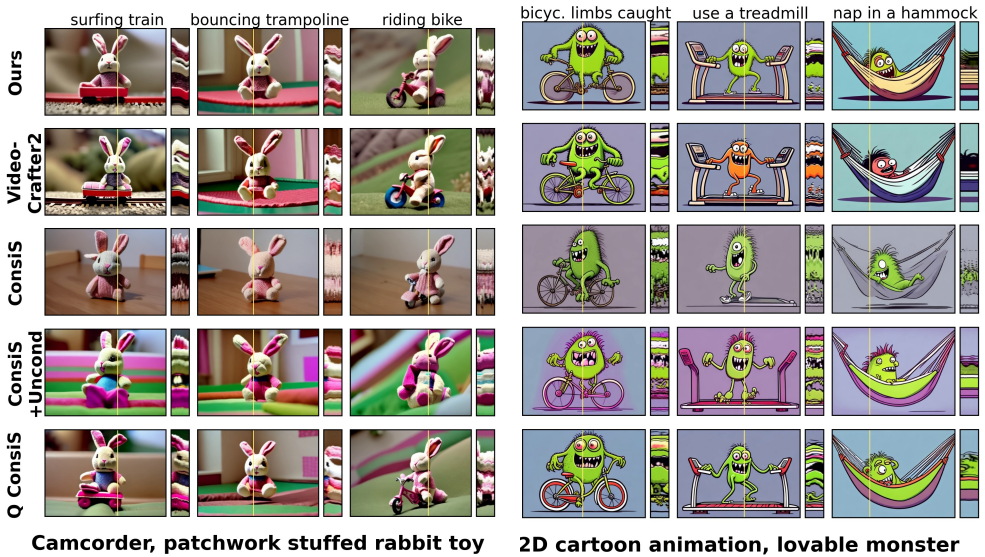


Figure 12: **Ablation Study on ConsiStory Components for Video Generation.** “Ours” (top row) demonstrates improved motion richness and identity preservation. VideoCrafter2 (second row) shows diverse motion but inconsistent characters. “ConsiS” (third row), a naive ConsiStory implementation, shows impaired identity and motion artifacts. “ConsiS + Uncond” (fourth row) adds feature injection to unconditional denoising, resolving motion artifacts but reducing motion magnitude and compromising identity. “Q ConsiS” (fifth row) couples each frame with a single frame in an anchor video, allowing some natural motion, although partially synchronized, with improved identity. Our method achieves the best balance of motion quality and identity.

videos hurts identity without significantly improving motion. Finally, our method (1st row - “Ours”) employs a novel Q intervention mechanism. It achieves richer motion with better identity and adherence to the original motion of the vanilla model.

B.8 Quantitative evaluation

We conducted a quantitative analysis using automated metrics and a user study, based on a benchmark dataset that we created to assess set-consistency in video generation.

Benchmark Dataset: We constructed a benchmark dataset of 30 video sets, each containing 5 video-shots with shared subjects but varying prompts. See further details in Appendix B.12.

Evaluation Protocol: To avoid overfitting, we conducted all development and parameter tuning on a separate collection of 16 distinct subject-prompt sets. The test set was used exclusively for final evaluations, without any component development or hyperparameter tuning.

Evaluation Metrics: Our evaluation approach builds on previous work in image consistency and personalization [18, 14, 49], focusing on multi-shot set-consistency and motion dynamics. For **set-consistency**, we measure average pairwise DINO feature similarity

[8, 25] across all frames in a set, excluding pairs within the same video shot. We isolate the subject by masking out the background [12] before extracting each frame’s features, using ClipSEG [46] with a dynamic threshold determined by “Otsu’s method” [38]. For **motion dynamics**, we evaluate all 150 generated videos using VBench’s “Dynamic Degree” metric [25], which classifies the significance of video motion by measuring RAFT-based optical flow intensity. We focused on motion dynamics over text prompt alignment due to two challenges: actions are often visible even in videos with minimal motion, making it difficult for temporal CLIP-like models [53] to distinguish between our method and baselines; also, sharing seeds across baselines lead to similar visual structures, with main differences in motion quality. We include text-similarity metrics in Table 3 (Appendix), measuring temporal CLIP similarity between each video shot and its prompt.

Results: Fig. 13 show our approach enhances multi-shot set consistency, while sacrificing motion magnitude compared to vanilla VideoCrafter2. Tokenflow-Encoder baseline shows consistency improvement and slight motion decrease. ConsiS-Im2Vid baseline’s performance aligns with qualitative analysis, showing low motion scores. A comparison of all baselines, including VSTAR and Turbo-V2, is presented in Table 3 (Appendix). VSTAR struggles with prompt control (19.8 vs 27.7 for ours), while achieving the highest consistency and motion dynamics. When combined with Turbo-V2, our method improves multi-shot consistency while maintaining high motion quality: The dynamic degree improves three-fold, from 20 to 62, while keeping the same level of text alignment.

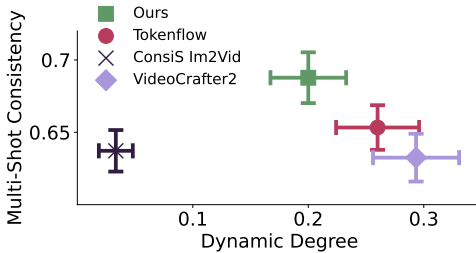


Figure 13: Quantitative Evaluation of Set Consistency and Motion Dynamics: Our approach achieves highest set consistency score while maintaining competitive motion dynamics. Error bars indicate standard error of the mean.

These quantitative results offer insights into trade-offs between our approach and baselines, but cannot fully capture user-perceived quality or alignment of generated motions with text prompts. Therefore, we conducted a comprehensive user preference study using two and three-alternative forced-choice format, focusing on two key aspects: set-consistency and text-motion alignment. For set-consistency, users selected the better set from two sets of 5 videos each depicting the subject. For text-motion alignment, users chose the video best matching the action described in the prompt from a pair of videos. To distinguish between degraded motions and those largely unchanged, users could also indicate if motion quality was equivalent in both videos. We used the same test benchmark as the automated metric study, collecting 5 repetitions per question for set-consistency and 3 repetitions for text-motion alignment, totaling 1800 responses.

The user-study results in Fig. 14, reveal that *Video Storyboarding* outperforms the baselines in set consistency. For motion quality, 55% of users rated the generated motions as similar or superior to those of the vanilla model. The ConsiS-Img2Vid baseline’s motion quality was consistent with our earlier findings, showing lower motion quality. However, it achieved the highest set consistency among the baselines, winning in 34% of the generated sets compared to our approach.

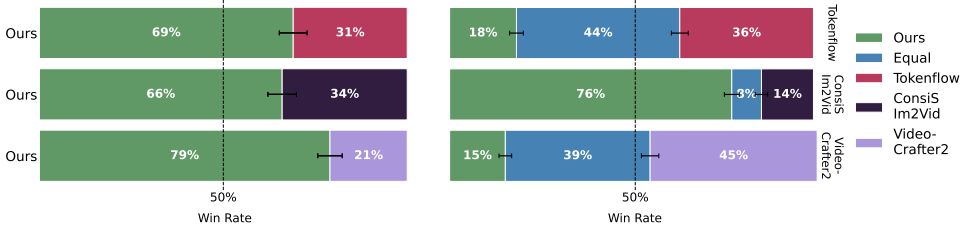


Figure 14: **User Study:** (left) We measure user preferences for set consistency and (right) how well the generated motion matches the text prompt. Our approach achieves the superior set consistency score while maintaining competitive text-motion alignment. Notably, 55% of our generated motions were judged to be of similar or better quality compared to the vanilla model. Error bars are S.E.M.

B.9 Additional Results

Fig. 15 illustrates the adaptability of our method when applied to the state-of-the-art T2V-Turbo-V2 model [80]. The results show enhanced motion quality while maintaining subject consistency, demonstrating that our approach can effectively improve even the most recent video generation models.

Fig. 16, provides additional qualitative comparisons to Fig. 11, and also includes qualitative comparison with VSTAR baseline [41].

In Table 3 we present a comprehensive quantitative comparison across different models using three key metrics. Our method, when combined with both VideoCrafter2 and Turbo-V2, shows improved Multi-Shot Consistency scores (68.8 and 67.3 respectively) compared to their baseline versions (63.2 and 63.3), while maintaining comparable Text Similarity and Dynamic Degree measurements. This indicates that our approach successfully enhances subject consistency without significantly compromising other important aspects of video generation. In the reported metrics, we also include a “Subject-Consistency” metric, introduced by VBench [45]. This metric measures the similarity between frames within the same video shot using DINO (see Table 1 in the Appendix).

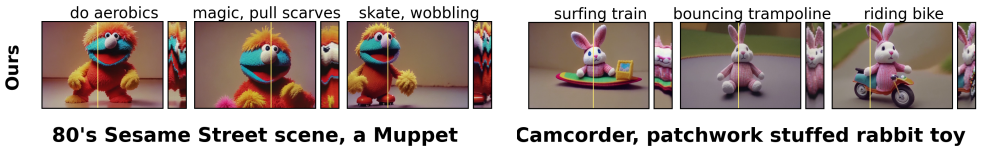


Figure 15: **T2V-Turbo-V2:** *Video Storyboarding* can be applied to T2V-Turbo-V2 [80], a recent state-of-the-art video model, that exhibits significantly better motion.

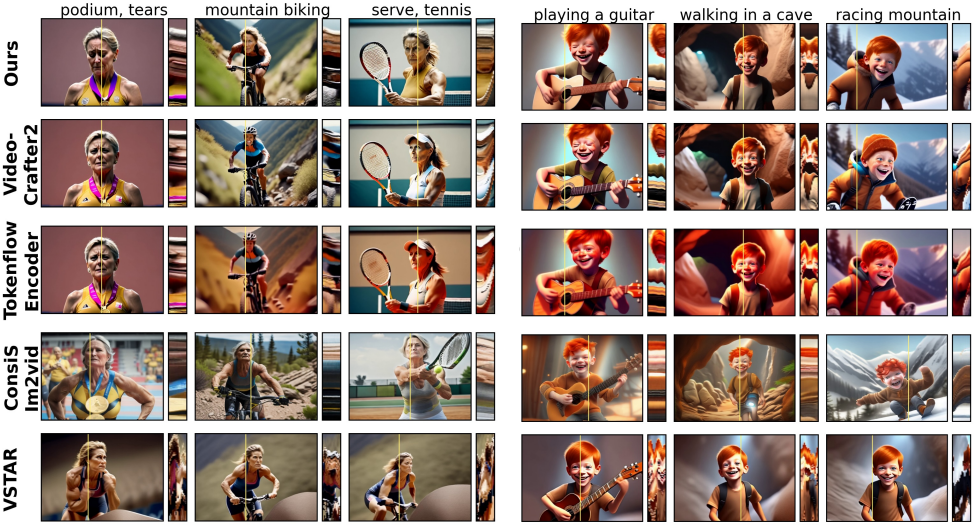
B.10 Framewise Subject-Driven Self-Attention - Implementation Details

This section provides a detailed explanation of our proposed Framewise-SDSA mechanism.



Cheerful 90's anime-style girl

Unreal Engine animated scene, bird



Cinematic, middle-aged female athlete

Painting animation, happy ginger boy



	MULTI-SHOT CONSISTENCY	TEXT SIMILARITY	DYNAMIC DEGREE	SUBJECT CONSISTENCY
CONSIStory IM2VID	63.7 \pm 1.4	27.3 \pm 0.5	3.3 \pm 1.5	99.1 \pm 0.1
VSTAR	83.9 \pm 1.6	19.8 \pm 0.4	90.7 \pm 2.4	92.6 \pm 0.3
TOKENFLOW	65.3 \pm 1.5	27.9 \pm 0.4	26.0 \pm 3.6	97.7 \pm 0.2
VIDEOCRAFTER2	63.2 \pm 1.7	28.7 \pm 0.4	29.3 \pm 3.7	97.3 \pm 0.2
OURS + VIDEOCRAFTER2	68.8 \pm 1.8	27.7 \pm 0.4	20.0 \pm 3.3	97.7 \pm 0.2
TURBO-V2	63.3 \pm 1.7	28.6 \pm 0.4	63.3 \pm 3.9	96.2 \pm 0.2
OURS + TURBO-V2	67.3 \pm 2.1	27.4 \pm 0.4	62.0 \pm 4.0	96.8 \pm 0.2

Table 3: **Quantitative Evaluation Metrics.** Comparison of different models across three metrics: Multi-Shot Consistency, Text Similarity, and Dynamic Degree. Values are reported as mean \pm standard error of the mean (S.E.M).

Improved Subject Localization. In video generation, subject localization becomes particularly challenging during early denoising steps, where the noise is most prominent. aggregation method proposed in ConsiStory (Sec. B.2) proved insufficient in this context, particularly during the earliest denoising steps, leading to unreliable masks both in terms of accuracy and false positive localization.

To address this, we propose using the estimated clean image \hat{x}_0 for subject localization instead of relying on internal network activations. At each denoising step t , we estimate \hat{x}_0 from the noisy latent x using: $\hat{x}_0 = (x - \sqrt{1 - \alpha_t} \cdot e_t) / \sqrt{\alpha_t}$, where e_t is the estimated noise, and α_t is the schedule parameter [47]. We then apply a zero-shot segmentation approach [56] to localize the subject in the estimated image, followed by Otsu’s method [58] to dynamically threshold the mask. This approach produces reliable subject masks from the earliest denoising steps and throughout the generation process.

Maintaining Motion Fluidity. Our experiments revealed that a direct application of SDSA – attending to all frames across all videos simultaneously – can lead to visual artifacts and frozen motion. We discovered that limiting attention to a single corresponding frame in other shots is most effective, as attending to two or more frames negatively impacts motion fluidity and introduces visual artifacts. Specifically, we propose a framewise attention scheme. Instead of attending to all frames across all video shots, frames with matching temporal indices across shots attend only to each other. This prevents visual artifacts and frozen motion, which occur when attending to multiple frames simultaneously and strikes a balance between subject consistency and natural motion.

Formal Definition of Framewise-SDSA. Let $K_{i,f}, Q_{i,f}, V_{i,f}, M_{i,f}$ be the keys, queries, values and subject-mask for frame f in video shot i . The framewise extended self-attention $A_{i,f}^+$ is defined by:

$$\begin{aligned}
 K_f^+ &= [K_{1,f} \oplus K_{2,f} \oplus \dots \oplus K_{N,f}] \\
 V_f^+ &= [V_{1,f} \oplus V_{2,f} \oplus \dots \oplus V_{N,f}] \\
 M_{i,f}^+ &= [M_{1,f} \oplus \dots \oplus M_{i-1,f} \oplus \mathbb{1} \oplus M_{i+1,f} \dots \oplus M_{N,f}] \\
 A_{i,f}^+ &= \text{softmax} \left(Q_i K_f^+ / \sqrt{d_k} + \log M_{i,f}^+ \right)
 \end{aligned}$$

$$h_{i,f} = A_{i,f}^+ \cdot V_f^+ \quad (1)$$

where \oplus indicates matrix concatenation. We use standard attention masking, which null-out softmax’s logits by assigning their scores to $-\infty$ according to the mask. Note that in this step, the Query tokens remain unaltered, and that the concatenated mask $M_{i,f}^+$ is set to be an array of 1’s for patch indices that belong to the i^{th} image itself.

B.11 Flow-based Q components injection - Formal Definition

Let $q_{f,xy} \in \mathbb{R}^F$ represent a Q feature from an originally generated video at location (x,y) in frame f . We denote by f_A and f_B the indices of the two nearest keyframes, where $f_A \leq f \leq f_B$. The locations of the most similar Q features in frames f_A and f_B , denoted by (x_A, y_A) and (x_B, y_B) respectively, are defined as:

$$(x_A, y_A) = \underset{x_0, y_0}{\operatorname{argmax}} \mathcal{S}_{\cos}(q_{f,xy}, q_{f_A x_0 y_0}) \quad (2)$$

$$(x_B, y_B) = \underset{x_0, y_0}{\operatorname{argmax}} \mathcal{S}_{\cos}(q_{f,xy}, q_{f_B x_0 y_0}) \quad (3)$$

where $\mathcal{S}_{\cos}(a, b)$ represents the cosine similarity between a and b .

We then generate the flow-adjusted Q feature by interpolating the target’s Q features at the corresponding locations:

$$\hat{q}_{f,xy} = w q_{c, f_A x_A y_A} + (1 - w) q_{c, f_B x_B y_B} \quad (4)$$

where q_c denotes the subject-consistent Q features from the target generation, and $w = \operatorname{sigmoid}\left(\frac{d_2}{d_1 + d_2}\right)$ with $d_1 = |f - f_A|$ and $d_2 = |f - f_B|$. This ensures that \hat{q} maintains the feature flow of the originally generated video, without injecting the actual features from it.

Note that in Algorithm 2, we use simplified index notation where $Q_{fl}^l(f) = w \cdot Q^l(f_A)[i_A] + (1 - w) \cdot Q^l(f_B)[i_B]$, with i_A and i_B representing the flattened indices corresponding to locations (x_A, y_A) and (x_B, y_B) respectively.

Finally, we blend the flow-adjusted features with the subject-consistent features using a linear schedule:

$$q_{final} = \alpha \hat{q}_{f,xy} + (1 - \alpha) q_{c, f, xy} \quad (5)$$

where $\alpha = 0.25 + t/1000$ and $q_{c, f, xy}$ is the subject-consistent Q feature from the target generation. This linear schedule gradually transitions from flow-based to subject-consistent features as denoising progresses.

B.12 Benchmark Dataset Construction:

We created a benchmark dataset comprising 30 video sets, each containing 5 video-shots depicting a shared subject under different prompts. The evaluation prompts were crafted using the Claude Sonnet 3.5 AI-Agent, following this protocol: each prompt consisted of three parts: (1) a subject description, e.g., “A girl” (2) a setting description, e.g., “paddling out on her surfboard”, and (3) a style descriptor encompassing both image and motion styles, e.g., “Anime cartoon animation” or “Shaky camcorder footage”. We instructed the AI-agent to choose actions that are visually striking and could be captured in a split second. Within each set, prompts shared the same subject and style but varied in settings. To ensure a challenging and representative test set, we selected a subset of 5 prompts per subject, prioritizing

those that produced videos with significant motion and subject variability when processed by the vanilla model. Importantly, to ensure fairness, this selection process relied solely on the vanilla model’s generations.

B.13 Q dropout

When Q-injection is too strong, it can compromise identity preservation. To address this, we introduce Q dropout, which reduces the strength of Q-injection. Unlike SDSA dropout, which hurts identity when trying to improve the image structure, Q dropout sacrifices some visual structural (motion) to enhance identity preservation. This Identity-Motion Trade-off is illustrated in Fig. 17, where increasing Q dropout improves identity consistency but reduces motion richness.



Figure 17: **Q dropout:** Q-injection may hurt identity. Q dropout may trade-off identity for motion. At 0% the unicorn gallops at both directions. At 40%, only to the right.

B.14 Implementation Details

Anchor Videos: Similar to ConsiStory, we utilize two anchor videos that share all features between themselves. Other videos in the batch only observe features derived from these anchors.

Scalable Video Batch Processing with Sub-batch Attention: To fit large batches of video generation within available GPU memory, we process the self and cross-attention computations in smaller sub-batches. This approach uses an internal loop, and subsequently concatenates results into a single tensor. The operation remains transparent to the network, enabling the generation of larger batches of video shots.

Reproducible denoising. Our pipeline involves three denoising iterations: caching vanilla queries, applying Q-injection and Framewise SDSA, and adding refinement feature injection. To ensure consistency across these stages, we maintain identical random generators for both initial noisy latents and the denoising process. This approach guarantees that each part builds upon the previous one, preserving the reliability of our reproducible denoising pipeline.

Temporal Parameters: For Q preservation, we set t_{pres} to 750. Framewise-SDSA is applied for $t \in [550, 950]$. Our refinement feature injection step is employed during $t \in [590, 950]$.

Feature Injection: We apply our refinement feature injection step to the 32×20 self-attention layers. Other layers either produced visual artifacts or did not significantly affect identity.

Denoising Process: Videos were sampled the default VideoCrafter2 configuration, using 50 DDPM steps with a guidance scale of 12.

T2V-Turbo-V2: For T2V-Turbo-V2 we adapt our Framewise-SDSA by allowing each frame to attend to both its temporally matching frames across shots and the middle frame of each shot. Other hyper-parameters were kept the same.

B.15 User Study Protocol

The following screenshots illustrate the experimental framework used in our user study:

Task

Example 1


Example 2

Select the group of videos that shows the same spider in all of them.
Concentrate on the spider's features and identity, ignoring the background, pose or motion.


Instructions

- Read these instructions carefully
- You are given two groups of videos, each showing a spider in different situations.
- Choose the group where the spider maintains a **consistent identity** throughout all of its videos.
- If some videos are inconsistent in both groups **choose the group with greater overall consistency**.
- Do not** judge based on the pose, background, or video quality. We want to assess the consistency of the spider's identity only. For example, choose a group showing the same spider in various poses and backgrounds over a group showing a slightly different spider in the same pose and room.
- Pay attention to the spider's identity** - things like eye color, texture, and facial features. Choose the group where these details are most consistent.
- Pay close attention to subtle details** that confirm it's the same spider in each video.
- Do not** judge based on video quality. Our goal is to assess the consistency of the spider's identity, not the quality of the videos.
- Do not** abuse the system, we take measures to spot that.
- Make sure** to always choose exactly one of the groups.
- For guidance, please refer to the examples and their solutions provided at the top of this page.

Current subject is **spider**



☐ Top set better show the same subject across examples



☐ Bottom set better show the same subject across examples

Submit


Figure 18: One trial of the visual consistency user study.

Task


Example 1

Example 2

In this example, we show that the choice need to be made according to subject identity, rather than consistent pose or environment.



This set should be chosen because its the same doll across examples, despite the different poses and combinations.




This set should not be chosen because despite the similar material, the doll is different.

Task


Example 1

Example 2

In this example, we show the you need to look at the fine details of each example in the set.



This set should not be chosen because it is not the same turtle in all examples, although similar.



This set should be chosen, because it is the same turtle in most examples, as can be seen by the fine details of the head and shell.

Figure 19: Examples provided in the user study for visual set consistency.

Task

Example 1

Example 2

Example 3

Choose the video that best matches the following text description.

Instructions

• Read these instructions carefully

• You will receive a textual description along with two videos. Your task is to determine in which video the motion of 6-year-old grandmother better matches the action in the description.

• Do not judge based on video quality. Our goal is to assess which motion better follows the action in the text description.

• Prioritize motion that correspond to the text. Do not choose frozen videos that correspond to the action.

• If you think that the motion quality is equal in both videos, you can select this option.


• Do not abuse the system, we take measures to spot that.

• Make sure to always choose exactly one of the videos.


• For guidance, please refer to the example videos and their solutions provided at the top of this page.

Text Description: a whimsical blue-haired 6-year-old grandmother transforming origami animals into real creatures, the camera circling as they come to life

video 1



video 2



☐ Video 1 better reflects the textual description

☐ Motion quality is equivalent in both videos

☐ Video 2 better reflects the textual description

Submit

Figure 20: One trial of the text-motion alignment user study.

Task


Example 1

Example 2


Example 3

Text Description: A dragon flying over a fantasy land.

video 1



video 2



This video should be chosen, since the dragon motion in this video better reflect the described action.

This video should not be chosen, since the dragon is portrayed as flying, however the motion does not corresponds with the action.

Task


Example 1

Example 2


Example 3

Text Description: A unicorn galloping over a rainbow.

video 1



video 2



This video should not be chosen, since the motion is less rich

This video should be chosen, because the motion is richer, as the unicorn is indeed galloping, although not over the rainbow.

Task


Example 1

Example 2


Example 3

Text Description: An owl snowboarding.

video 1



video 2



The motion is quality is equivalent in both videos. In this case it is best to select 'Motion quality is equivalent in both videos'

The motion is quality is equivalent in both videos. In this case it is best to select 'Motion quality is equivalent in both videos'

Figure 21: Examples provided in the user study for text-motion alignment.