FlowMotion: Training-Free Flow Guidance for Video Motion Transfer
********** Supplementary Document **********

Zhen Wang[1], Youcan Xu[2], Jun Xiao[2], Long Chen[1,†]
[1] The Hong Kong University of Science and Technology [2]State key lab of CAD&CG, Zhejiang University
zhenwang@ust.hk, youcan@zju.edu.cn, junx@cs.zju.edu.cn, longchen@ust.hk

This supplementary document is organized as follows:

- In Sec. A, we show more analysis of flow-based T2V model for extracting the motion representation.
- In Sec. B, we provide more discussion and justification of the memory efficiency of FlowMotion.
- In Sec. C, we provide more discussion about the variant of source motion representation.
- In Sec. D, we show the inference scheme.
- In Sec. E, we show more details about the experimental setup.
- In Sec. F, we show the detailed qualitative results.
- In Sec. G, we provide ablations about hyperparameters.
- In Sec. H, we provide visualization results for long videos.
- In Sec. I, we discuss the failure cases, limitations and potential future works.

A. More Analysis of Flow-based T2V Generation for Extracting Motion Representation

As stated in Sec. 3.2, we focus on the model's predicted outputs and investigate whether the intrinsic predictions of flow-based T2V models contain sufficient motion information for direct and efficient transfer. Here, we provide a more detailed analysis of the different predicted outputs and their characteristics. As illustrated in Figure 5(a), at each denoising step of flow-based T2V generation, given a noised latent z_t as input, the model directly predicts the instantaneous *velocity* v_t. In the standard denoising process, v_t is used to update z_t with a small step, yielding the *denoised latent* $z_{t-1} = z_t - v_t\,dt$. Furthermore, we introduce a *latent prediction* defined as $\hat{z}_0(t) = z_t - t\,v_t$, which provides a single-step estimation of the clean latent.

In summary, either directly or indirectly, flow-based T2V models produce three types of predicted outputs: velocity, denoised latent, and latent prediction. In the following, we analyze these outputs through (1) visualizations during the T2V process, (2) extraction and visualization from source videos, and (3) use them as motion representations for guidance. Specifically, we apply the forward noising process described in Sec. 3.2 to source videos to obtain z_t^{src}, which are then fed into the T2V model to extract predicted outputs.

Velocity. As shown in Figure 5(b1), the visualizations of velocity exhibit only coarse and noisy motion trends. Although the motion pattern evolves progressively with the denoising steps, the instantaneous nature of velocity — representing a noisy-to-clean direction at each step — makes it inherently noisy throughout generation and difficult to interpret. Similarly, Figure 5(c1) shows that the velocities extracted from source videos remain highly noisy, making the underlying motion signals hard to disentangle. Consequently, as demonstrated in Figure 5(d1), using velocity as the motion representation in our flow guidance leads to imprecise motion transfer: the target video follows only a rough motion pattern and often exhibits abrupt changes. In addition, using velocity for guidance still incurs high memory consumption, as gradient backpropagation through internal layers is required.

Denoised Latent. As shown in Figure 5(b2), since each denoising step updates only a small portion of the latent, the denoised latents evolve extremely slowly. In the early steps (*e.g.*, the first 10), almost no recognizable motion or appearance information emerges, and meaningful structures only begin to appear at much later steps (around step 30). Moreover, the denoised latents remain highly noisy, and motion and appearance features are heavily entangled, without a clear progression of motion cues. Similarly, Figure 5(c2) shows that denoised latents extracted from source videos exhibit the same characteristics: motion signals are weak, appear only in late steps, and are deeply entangled with appearance information. Consequently, as illustrated in Figure 5(d2), using denoised latent as the motion representation in our flow guidance fails to achieve meaningful motion transfer. The motion cues are too faint and emerge too late for effective optimization, even when the guidance applied throughout the entire denoising process. Thus, although using denoised latent avoids high memory consumption, their guidance effectiveness is severely limited.

Latent Prediction. From the above analyses of velocity and denoised latent, we can observe that velocity contains

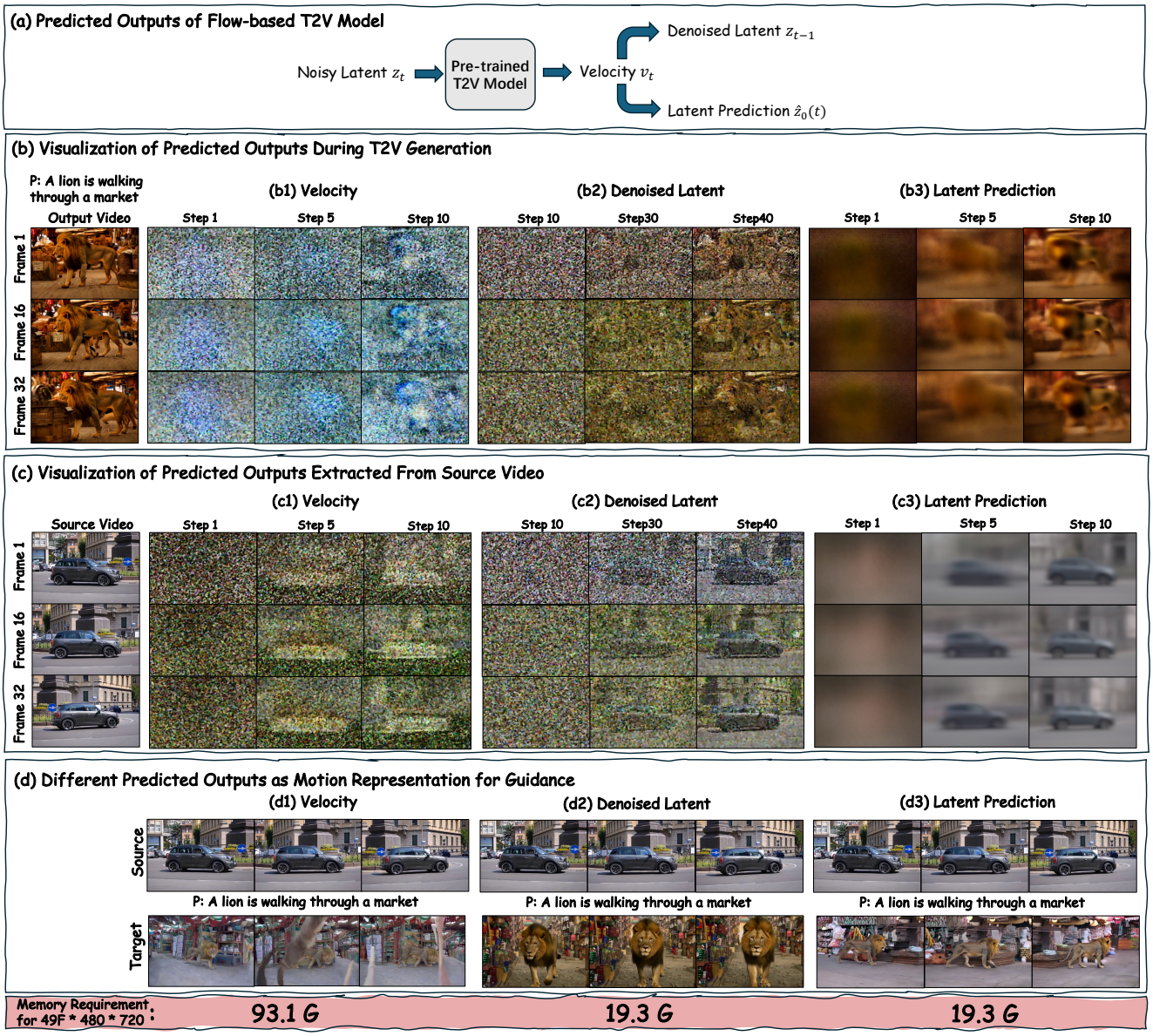

Figure 5. Analysis of Flow-based T2V Generation for Extracting Motion Representation.

highly noisy and hard-to-disentangle motion cues, while the small-step updates in denoised latents provide changes that are too weak and appear only in late denoising stages. These limitations motivate the use of a one-step approximation of the clean latent, *i.e.*, the latent prediction computed directly from velocity as described in the main paper. As shown in Figure 5(b3), early-stage latent predictions inherently encode rich temporal information, exhibiting a clear progressive evolution from coarse spatial locations, to object trajectories, and eventually to fine-grained actions and scene-level dynamics. Correspondingly, Figure 5(c3) shows that latent predictions extracted from source videos follow the same pattern, capturing motion that unfolds progressively from global movement to detailed temporal dynamics. As illustrated in Figure 5(d3), using latent predictions as the motion representation in our flow guidance achieves accurate motion transfer while avoiding the high memory cost. This makes latent prediction both an effective and efficient choice for constructing guidance for motion transfer.

B. Justification for Memory Efficiency

We provide a more detailed analysis of the guidance mechanisms used in different methods to clarify their memory consumption. In training-free video motion transfer pipelines, the core principle is to optimize the latent of the target video by minimizing the discrepancy between the motion representations of the source and the generated target video. Consequently, the overall process typically in-

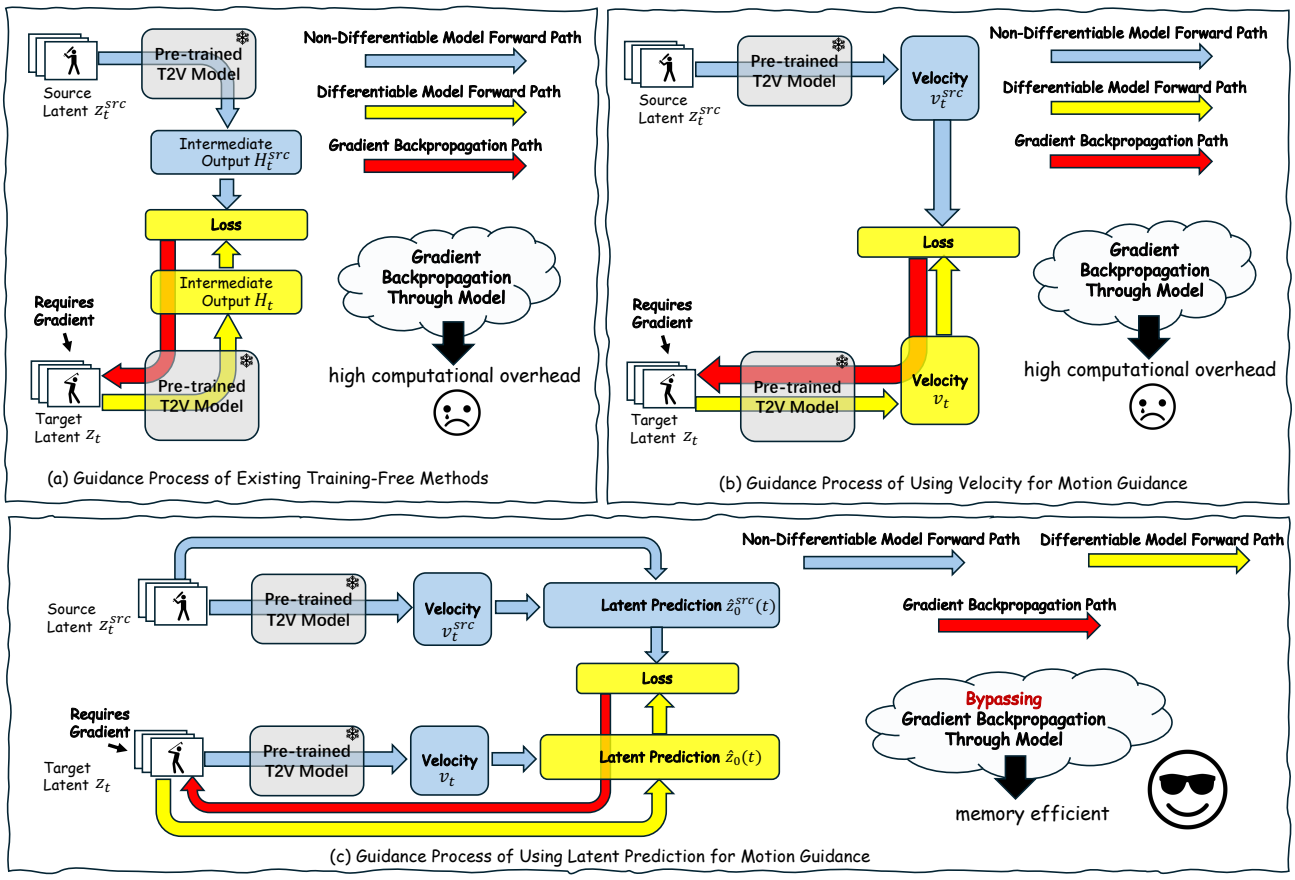

Figure 6. Visualization of the Guidance Process of Different Methods.

volves two stages: 1) extracting appropriate motion representations from both the source and target videos, and 2) computing a loss between these representations and backpropagating through the model to update the target latent.

Existing Training-Free Methods. As shown in Figure 6(a), existing training-free methods construct motion representations from intermediate model outputs, such as diffusion features or temporal attention maps. For the source video, the intermediate representation H_t^{src} is obtained through a non-differentiable forward pass, which incurs no gradient cost. In contrast, the target latent z_t must undergo a differentiable forward pass to extract its intermediate output H_t, based on which a loss is computed against H_t^{src} to update z_t via gradient descent.

Crucially, since the target latent must be updated by gradient descent, the entire computation from z_t to H_t must remain *fully differentiable*, meaning that all internal model layers involved in producing H_t are retained in the computational graph. As a result, the gradient backpropagation must traverse these deep layers, which further leads to substantial memory consumption.

Using Velocity for Motion Guidance. As discussed in Sec. A, the instantaneous predicted output, *i.e.*, the velocity v_t, can also serve as a motion representation. However, as illustrated in Figure 6(b), obtaining the target velocity still requires a differentiable forward pass from z_t through the flow model. Since the loss between v_t and the source velocity v_t^{src} is used to update z_t, this forward pass must remain fully differentiable, and the resulting gradient backpropagation again traverses the internal deep layers. Thus, directly using velocity-based guidance incurs similarly high memory consumption.

Using Latent Prediction for Motion Guidance. As introduced in the main paper and Sec. A, we define the latent prediction as $\hat{z}_0(t) = z_t - t\, v_t$, which provides a single-step estimation of the clean latent. As shown in Figure 6(c), we use these latent predictions as motion representations and compute a loss between the target latent prediction $\hat{z}_0(t)$ and the source latent prediction $\hat{z}_0^{src}(t)$.

A key design lies in how $\hat{z}_0(t)$ is constructed. Although the target velocity v_t is involved in its definition, $\hat{z}_0(t)$ is ultimately a direct function of the target latent z_t. Thus, for gradient-based optimization, only the computational graph from $\hat{z}_0(t)$ to z_t needs to be preserved. The velocity v_t can be treated as a constant obtained from a non-differentiable model forward pass, so the forward path producing v_t does not need to be differentiable and is excluded from the computational graph. Consequently, the loss gradient is back-

Figure 7. Video Motion Transfer Results with Variant of Source Motion Representations.

propagated directly through $\hat{z}_0(t)$ to z_t, completely bypassing the internal model layers. This eliminates the need to store activations for backpropagation and yields significantly lower memory consumption.

Besides, building on the above design, using the denoised latent $z_{t-1} = z_t - v_t\, dt$ for motion guidance can also reduce memory consumption. However, as discussed in Sec. A, directly using denoised latent for guidance is less effective in achieving meaningful motion transfer.

The proposed latent prediction offers two key advantages: 1) it operates directly on the model's predicted outputs, without requiring intermediate features such as cross-frame attention that depend on specific model architectures. 2) by virtue of its definition, it elegantly bypasses backpropagation through internal model layers, significantly reducing memory consumption.

In summary, this latent-level flow guidance leverages the model's predicted outputs, avoids architectural dependencies, and simultaneously achieves both computational efficiency and effective motion transfer.

C. Variant of Source Motion Representation

We provide additional discussion on the variant of source motion representation referenced in Sec. 4.4. Since the latent prediction can be viewed as a one-step approximation of the final clean latent, thus for the source video, we can actually directly obtain the "real" clean latent, *i.e.*, the clean source latent z_0^{src}. This allows a natural variant in which the flow guidance is constructed using z_0^{src} as the source motion representation, while the target motion representation remains the step-wise latent prediction $\hat{z}_0(t)$.

For this variant, we highlight two key differences: 1) Using latent prediction $\hat{z}_0^{src}(t)$ for the source yields a step-aligned representation that corresponds one-to-one with the

target motion representation $\hat{z}_0(t)$. As analyzed and visualized in Sec. A, both exhibit a similar coarse-to-fine temporal evolution across denoising steps, leading to a more naturally aligned guidance signal. 2) In contrast, using the source latent z_0^{src} provides a stronger and cleaner motion cue at every step, since it encodes the fully denoised motion information of the source video, but also introduces entangled appearance details that may lead to suboptimal motion alignment. We provide corresponding visualizations in Figure 7, where the two variants are respectively used as the source motion representation to construct our flow guidance for motion transfer, enabling a detailed comparison of their behaviors.

As shown in Figure 7(a)(b), for normal or easy motions where the dominant dynamics lie in global trajectory changes rather than shape or pose deformations, latent predictions offer a more balanced transfer. They preserve motion fidelity while maintaining text alignment and scene diversity. In contrast, using the clean source latent injects strong appearance information, which may cause appearance leakage into the target object. For instance, in Figure 7(a), the generated "drone" inherits the "swan"'s outline, and in Figure 7(b), the synthesized "toy car" resembles the source vehicle in shape and appearance, together with reduced background diversity.

Conversely, Figures 7(c)(d) illustrate cases with fine-grained or pose-dependent motion such as limb articulation, body deformation, or coordinated multi-part dynamics where the clean source latent performs better. It preserves nuanced pose transitions and detailed actions, accurately transferring the "bear"'s leg motion in Figure 7(c) and the complex human action in Figure 7(d). While latent predictions still capture the coarse motion trend, their transfer tends to be more global and less precise at the action level. Nevertheless, latent predictions consistently achieve stronger text alignment and more diverse visual appearance, whereas clean source latent sacrifices these aspects due to stronger entangled cues.

In summary, latent predictions are well suited for most normal or general motions including global trajectory changes or commonly occurring actions, offering a natural balance between motion fidelity and text alignment. Clean source latent, while appearance-entangled, serves as an effective alternative for challenging cases where motion is strongly tied to object pose or fine-grained shape dynamics. Therefore, in this work, we treat the clean source latent as an optional variant specifically for these complex scenarios.

These observations further reveal two promising future directions: 1) Combining latent prediction and clean latent to form a more comprehensive latent-level guidance that simultaneously strengthens fine-grained motion transfer and text alignment; 2) Designing adaptive guidance policies that allow controllable transfer strength, *i.e.*, dynamically selecting which aspects of the motion (global trajectory or detailed actions) to transfer based on the nature of the source video. As the first work to leverage latent-level model outputs for guidance, we hope these findings will stimulate further progress in this emerging direction.

Algorithm 1 FlowMotion inference pipeline

Input: Source video $\mathcal{V}$, pre-trained flow-based T2V model v_θ, encoder $\mathcal{E}$, decoder $\mathcal{D}$, prompt P.
Output: Generated video x_0 with transferred motion

Extract source latent representation: $z_0^{ref} \leftarrow \mathcal{E}(\mathcal{V})$
Initialize $z_1 \sim \mathcal{N}(0, I)$
for denoising step $t_d = 1$ to T with timestep $t = 1$ to 0 **do**
  if $t_d < T_{\text{opt}}$ **then**
    Sample $\epsilon_t \sim \mathcal{N}(0, I)$
    Forward Noising: $z_t^{src} = (1-t)z_0^{src} + t\epsilon_t$
    Compute $\hat{z}_0^{src}(t) = z_t^{src} - t v_\theta(z_t^{src}, t, \emptyset)$
    Compute $\triangle(\hat{z}_0^{src}(t))$
    for optimization step $k = 0$ to K_{opt} **do**
      Compute $v_t = v_\theta(z_t, t, P)$
      Compute $v_t^{avg}, v_t^{proj}, v_t^{orth}$
      Regulate $v_t^{reg} = v_t^{proj} + \gamma \cdot v_t^{orth}$
      Extract $\hat{z}_0(t) = z_t - t\, v_t^{reg}$
      Compute $\triangle(\hat{z}_0(t))$
      Get $\mathcal{L}_{FG} = \alpha \left\| \hat{z}_0^{src}(t) - \hat{z}_0(t) \right\|_2^2 +$
      $\beta \left\| \triangle(\hat{z}_0^{src}(t)) - \triangle(\hat{z}_0(t)) \right\|_2^2$
      Update z_t by minimizing $\mathcal{L}_{FG}$
    end for
  end if
  $z_{t-1} = z_t - v_\theta(z_t, t, P)dt$
end for
return $x_0 = \mathcal{D}(z_0)$

D. Inference Pipeline

We show the full inference pipeline in Algorithm1. The whole process built upon a general T2V inference process, start form a random noise $z_1 \sim \mathcal{N}(0, I)$, we perform T denoising steps ($t_d = 1$ to T) with corresponding timestep $t = 1$ to 0 to generate the final video $x_0 = \mathcal{D}(z_0)$. Specifically, we apply the guidance in the first T_{opt} denoising step for optimization (*i.e.*, $t_d < T_{\text{opt}}$). Furthermore, for each denoising step we apply optimization, we can optimize the latent for iteratively for K_{opt} optimization steps. As discussed in Sec C, we can also replace $\hat{z}_0^{src}(t)$ with z_0^{src} as a variant.

E. Experimental Setup

Datasets. MTBench [5] provides high-quality videos with broad motion diversity, covering (1) various subject domains (*e.g.*, humans, animals, and vehicles) and (2) different levels of motion difficulty, categorized into easy,

Method	Easy			Medium			Hard			All			Efficiency		
	TS↑	MF↑	TC↑	TS↑	MF↑	TC↑	TS↑	MF↑	TC↑	TS↑	MF↑	TC↑	Train(s)	Infer(s)	Mem(G)
Training-based															
LoRA Tuning [1]	0.333	0.752	0.977	0.321	0.843	0.982	0.323	0.761	0.971	0.327	0.782	0.977	8100	135	25.0
MotionDirector [10]	0.338	0.767	0.973	0.333	0.895	0.972	0.337	0.752	0.962	0.335	0.801	0.969	1662	140	28.0
MotionInversion [7]	0.327	0.802	0.973	0.325	0.909	0.973	0.334	**0.820**	0.964	0.328	0.839	0.970	1170	115	24.0
DeT [5]	0.348	0.774	0.980	0.335	0.898	0.983	0.334	0.777	0.976	0.340	0.812	0.980	2760	133	20.0
Training-free															
MotionClone [3]	0.332	0.788	0.947	0.332	0.836	0.943	0.333	0.734	0.929	0.332	0.786	0.940	-	804	51.5
MOFT [8]	0.346	0.631	0.972	0.326	0.599	0.979	0.341	0.501	0.970	0.338	0.582	0.973	-	576	75.0
SMM [9]	0.323	0.716	0.961	0.320	0.862	0.959	0.325	0.727	0.953	0.322	0.762	0.958	-	1839	89.4
DiTFlow [4]	0.350	0.692	0.986	**0.351**	0.771	0.985	**0.350**	0.611	0.976	**0.350**	0.691	0.983	-	349	63.5
Ours	**0.353**	**0.834**	**0.988**	0.345	**0.912**	**0.989**	0.339	0.808	**0.980**	0.347	**0.850**	**0.986**	-	213	19.3

Table 9. **Quantitative comparison with SOTA video motion transfer methods**. Best results in **bold** and second best are underlined.

medium, and hard based on trajectory complexity. However, we observe that MTBench has relatively limited camera motion and contains a certain degree of repetitive video patterns. In addition, some of the evaluation prompts are relatively simple. To construct a diverse yet efficient evaluation set, we primarily select videos from MTBench, supplement them with additional camera-motion videos from existing open-source datasets, and further rewrite and filter the prompts. This process results in a curated evaluation set of 50 videos, each paired with three refined prompts, ensuring both diversity and evaluation efficiency.

Implementation Details. We apply FlowMotion on two pre-trained flow-based T2V models — Wan2.1-1.3B and Wan2.2-5B [6]. Generally, for both models, we employ $T = 50$ denoising steps for T2V generation with a classifier-free guidance of 6, and apply the optimization for the initial $T_{\text{opt}} = 10$ denoising steps. We use the Adam [2] optimizer with a learning rate of 0.003 for $K_{\text{opt}} = 3$ optimization steps. We set $\alpha : \beta = 4 : 1$ for the flow guidance loss, and $\gamma = 0.1$ for the velocity regularization. Besides, based on the observation and discussion provided in Sec C, for qualitative evaluation, we use the latent prediction as source motion representation for easy and medium videos. And use the clean latent as source motion representation for hard videos. We do not employ any additional memory-saving techniques such as CPU offloading.

F. Detailed Qualitative Results

We provide more detailed qualitative results in Table 9 across different motion difficulty levels. We can observe: 1) FlowMotion achieves the most balanced performance across all metrics and difficulty levels with superior computational efficiency. 2) The training-free method DitFlow obtains high text similarity but exhibits poor motion fidelity. In contrast, the training-based method MotionInversion tends to overfit, leading to degraded text alignment and reduced temporal consistency. 3) Notably, all methods perform relatively worse on videos categorized as *hard*, which typically involve complex human actions and multiple-object movements. Besides, most approaches achieve better results on *medium* videos than on *easy* ones. Upon inspection, we find that MTBench's difficulty annotations sometimes place visually simple videos into the medium category, while certain visually complex videos appear in the easy category. This indicates that developing a more principled and reliable motion-difficulty categorization protocol remains an important direction for future work.

$\alpha : \beta$	Text Similarity↑	Motion Fidelity.↑	Temporal Consis. ↑
1 : 2	0.336	0.742	**0.988**
1 : 1	0.335	0.818	**0.988**
2 : 1	0.340	0.846	0.986
4 : 1	**0.341**	**0.854**	0.986
8 : 1	0.338	0.848	0.985

Table 10. Ablation of weighting coefficients for guidance loss.

G. More Ablation Study

We provide more ablation studies for the hyperparameter choices of flow guidance. Specifically, we run ablations on Wan2.1-1.3B with a subset of 15 videos.

Weighting Coefficients for Flow Guidance Loss. The weighting coefficients α and β control the balance between latent alignment and frame-wise difference alignment. As shown in Table 10, setting β larger than α weakens global alignment and leads to degraded motion quality. Increasing α progressively improves overall alignment, and a reasonable trade-off emerges around $\alpha : \beta = 4 : 1$. This indicates that flow guidance primarily relies on global alignment, while frame-wise differences serve as complementary refinement. The flow guidance is not sensitive to hyperparameters, where ratios within a reasonable range (*e.g.*, $2 : 1$ or $4 : 1$) all provide stable performance. We adopt $4 : 1$ as a general purpose setting.

Decay Factor for Velocity Regularization. For velocity regularization, we decompose the velocity and attenuate its orthogonal component using a decay factor $\gamma \in [0, 1]$. As shown in Table 11, removing regularization ($\gamma = 1$) leads

γ	Text Similarity↑	Motion Fidelity↑	Temporal Consis. ↑
1.0	0.316	0.844	0.974
0.5	0.324	**0.878**	0.980
0.3	0.334	0.871	0.981
0.1	**0.341**	0.854	0.986
0.0	0.338	0.809	**0.987**

Table 11. Ablation of decay factor for velocity regularization.

T_{opt}	Text Similarity↑	Motion Fidelity↑	Temp Consis. ↑
5	**0.353**	0.708	0.988
10	0.341	0.854	**0.986**
15	0.325	**0.888**	0.984

Table 12. Ablation of denoising step for optimization.

K_{opt}	Text Similarity↑	Motion Fidelity↑	Temp Consis. ↑
1	**0.356**	0.549	**0.987**
3	0.341	0.854	0.986
5	0.330	**0.897**	0.985

Table 13. Ablation of optimization step.

Motion Rep.	Text Sim.↑	Motion Fid.↑	Temp Consis. ↑
Clean Latent	0.331	0.907	0.984
Latent Prediction	0.341	0.854	0.986

Table 14. Ablation of variant for source motion representation.

to low text alignment and temporal quality, while enabling decay consistently improves the results. Increasing the decay strength (*e.g.*, γ from 0.5 to 0.3) does not yield strictly linear performance changes, instead, the results gradually stabilize toward trade-offs. However, setting $\gamma = 0$ eliminates all orthogonal updates, which overly constrains the optimization dynamics and consequently reduces motion fidelity. Besides, in some cases, visual artifacts caused by unstable optimization may not be fully reflected by quantitative metrics. Therefore, we adopt a relatively strong decay factor of $\gamma = 0.1$ as a robust default setting.

Denoising Step for Optimization. As discussed in Sec.3.2, the motion information evolving progressively during the early denoising phase (approximately the first 10 steps), after which appearance details become increasingly dominant. As shown in Table12, applying guidance for only the first 5 denoising steps results in insufficient motion transfer, while extending it to 15 steps tends to overfit appearance and reduce text similarity. Therefore, we generally optimize over the first 10 denoising steps as a balanced trade-off.

Optimization Step. For each denoising step which we apply the guidance, we can iteratively optimize the latent with multiple optimization steps. As shown in Table 13, using only a single optimization step leads to insufficient motion transfer, while using five steps tends to overfit and harm text alignment, in addition to increasing runtime. Therefore, we set the number of optimization step to 3 as a reasonable trade-off.

Variant of Source Motion Representation. We further ablate the choice of source motion representation discussed in Sec.C. As shown in Table14, using the clean source latent achieves stronger motion transfer but tends to overfit, which harms text alignment. Thus, it is more appropriate to select the representation based on the motion characteristics. In general, latent prediction is preferable for typical motions, while clean source latent is better suited for complex, pose-dependent motions. As the first work to introduce latent-level guidance for motion transfer, we believe future research can explore dynamic or hybrid guidance strategies to achieve more comprehensive improvements.

About the "trade-off". Across all ablations and visual analyses, the text alignment and motion fidelity inherently trade off against each other, which is also the core challenge observed in existing motion transfer methods (*cf*., Sec. F). Stronger guidance — such as more denoising steps, more optimization steps, or using clean source latent — indeed improves motion fidelity, but also increases the risk of overfitting and reduces text similarity and diversity.

When the overall performance is already strong in terms of quantitative metrics, the **numerical** gain in motion fidelity from stronger guidance often outweighs the corresponding drop in text similarity, yet the **visual** outcome may shows the opposite trend, *i.e*., stronger guidance tends to introduce more noticeable artifacts and appearance degradation. Specifically, we observe that once motion transfer is already visually satisfactory, pushing guidance strength can continue to raise motion fidelity metric, yet often brings little visual benefit or even noticeable degradation due to appearance overfitting. For example, in Fig. 7(a), clean latent yields higher motion fidelity score simply because the generated shape becomes closer to the source, but visually the overfitted appearance is clearly undesirable.

In practice, video motion transfer is ultimately a generative task, where users care most about perceptual quality and the ability to create diverse scenes while preserving motion. Therefore, when choosing implementation settings, we tend to place slightly more emphasis on text alignment in practice to achieve a more balanced overall result. At the same time, the long-term goal remains to improve both sides of the trade-off simultaneously.

Besides, adjusting the guidance strength according to different scenarios and motion patterns allows flexible control over the transfer behavior. In this paper, we provide a generally applicable setting based on the above observations and findings.

H. Results for Longer Videos

Thanks to the computational efficiency of FlowMotion, our method can be easily applied to longer videos. As shown in

Figure 8. Visualization for video motion transfer on longer videos.

Figure 8, for videos with 81 frames, FlowMotion achieves effective motion transfer across diverse subjects and scenes. Moreover, for a 480×720 source video with 81 frames, FlowMotion requires only 20.1 G of GPU memory. Since our approach avoids backpropagating through the diffusion model, the additional memory compared with the 49-frame case (19.3G) comes solely from the increased video latent sequence length, resulting in only a marginal GPU overhead. In summary, this efficiency allows our method to run comfortably on consumer-grade GPUs such as the NVIDIA RTX 3090 and 4090, and further underscores FlowMotion's strong potential for scaling to even longer videos.

I. Limitation and Future Work

We now discuss the failure cases, limitations, and potential future directions. As a training-free framework built on top of a pre-trained T2V model, our method inherently depends on the generation capability of the underlying model. Consequently, it struggles with motion types or prompts that are out of training distribution. In addition, similar to existing methods, as the motion becomes increasingly complex or fine-grained, the transfer ability of the framework gradually reaches its limit.

Complexity. Complex motions involving fine-grained human body movements remain challenging for motion transfer. As shown in Figure 9(a), when handling motions such

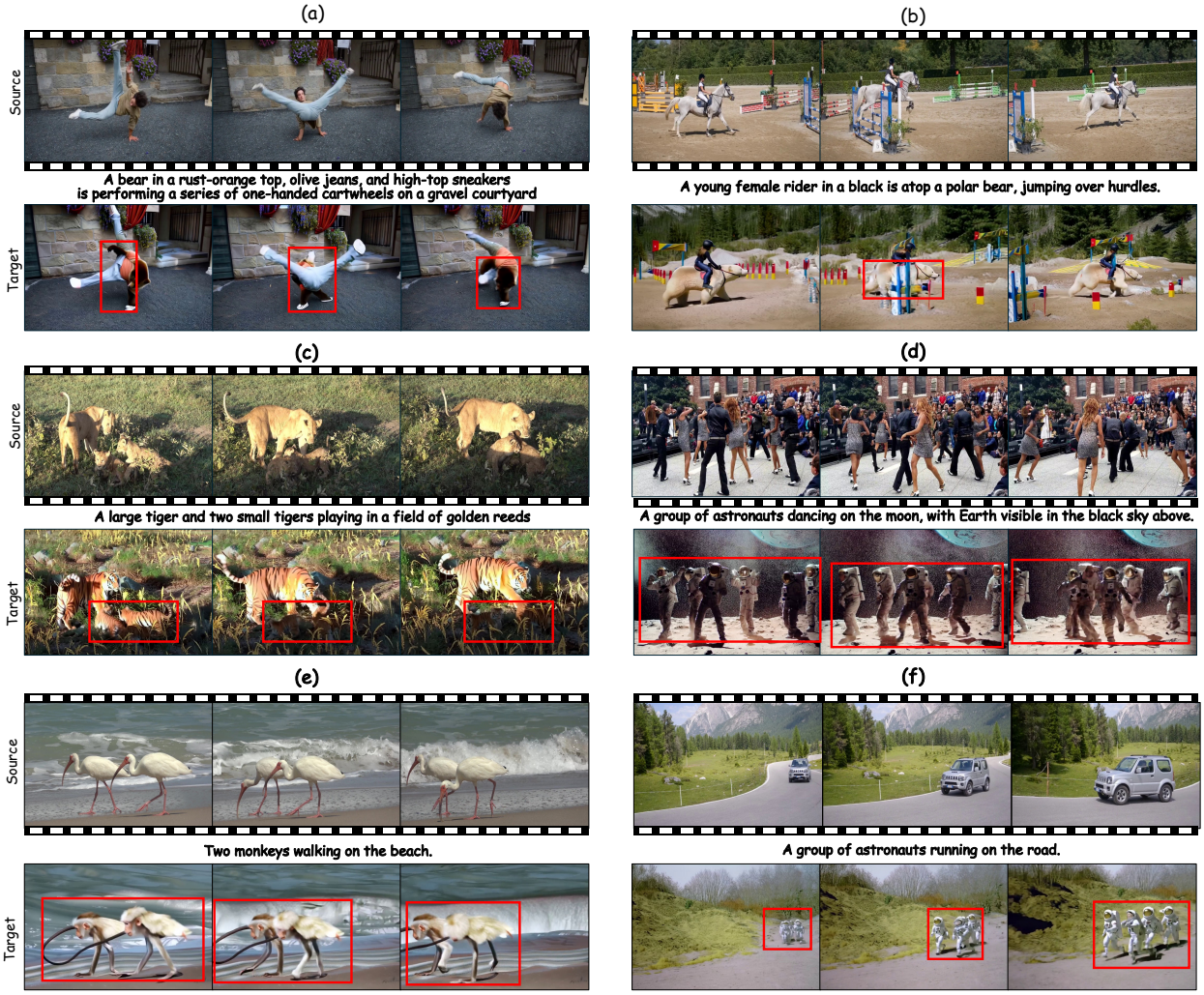

Figure 9. Visualization for the failure cases and limitations.

as "break dance" that involve fine-grained limb coordination, rotations, and uncommon inverted poses, the transferred results exhibit inconsistencies. For example, incorrect hand/foot poses, distorted limbs, or insufficient text alignment of the bear's appearance. Moreover, for composite motions, as illustrated in Figure 9(b), where multiple object movements and camera motion co-occur, the transferred motion may become inaccurate, such as failing to reproduce the "jumping" motion at the correct location.

As the first work adopting latent prediction as guidance, our approach still operates at a relatively global latent level. Future work could explore incorporating more structured and fine-grained motion cues, such as human keypoints, trajectory tracking, or other explicit motion priors to achieve more precise alignment in complex scenarios.

Number of Objects. As the number of objects increases, and as each object exhibits its own distinct motion, the transfer performance gradually becomes constrained. As shown in Figure 9(c), for a scene involving three objects, the transfer for the largest tiger remains relatively accurate, while the two smaller tigers are not well preserved. Furthermore, as illustrated in Figure 9(d), when the number of objects exceeds five together with crowd in the background, it becomes increasingly difficult to maintain accurate motion transfer for all objects, and even the object count may become inconsistent.

Notably, existing motion transfer methods primarily focus on single-object scenarios or, at most, two objects. Thus, multi-object motion remains an open challenge. For future work, incorporating spatial priors such as object masks to perform localized latent prediction may enable object-specific motion extraction and improve multi-object motion transfer.

Transfer Strength and Semantic Gap. Existing methods

still struggle to automatically determine "how much" motion should be transferred to the target video. When there is a large semantic gap between the subject described in the target prompt and the subject in the source video, the transfer quality becomes notably limited. As shown in Figure 9(e), when transferring the motion of a "sea birds" to "monkeys", the model cannot reliably decide whether to retain the global trajectory or the fine-grained pose patterns. This misalignment often leads to visually inconsistent results or distortions, such as generating monkeys that are partially shaped like birds.

Similarly, as illustrated in Figure 9(f), transferring the motion of "a single car" to "a group of astronauts" introduces semantic mismatch, resulting in degraded performance, such as sudden changes in the number of astronauts or inaccurate motion trajectories for each individual.

Future work may explore adaptive guidance strategies that enable controllable transfer strength. For example, dynamically choosing whether to emphasize global trajectories or detailed actions based on the characteristics of the source motion and the semantics of the target prompt.

References

[1] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 6

[2] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[3] Pengyang Ling, Jiazi Bu, Pan Zhang, Xiaoyi Dong, Yuhang Zang, Tong Wu, Huaian Chen, Jiaqi Wang, and Yi Jin. Motionclone: Training-free motion cloning for controllable video generation. *arXiv preprint arXiv:2406.05338*, 2024. 6

[4] Alexander Pondaven, Aliaksandr Siarohin, Sergey Tulyakov, Philip Torr, and Fabio Pizzati. Video motion transfer with diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22911–22921, 2025. 6

[5] Qingyu Shi, Jianzong Wu, Jinbin Bai, Jiangning Zhang, Lu Qi, Xiangtai Li, and Yunhai Tong. Decouple and track: Benchmarking and improving video diffusion transformers for motion transfer. *arXiv preprint arXiv:2503.17350*, 2025. 5, 6

[6] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 6

[7] Luozhou Wang, Ziyang Mai, Guibao Shen, Yixun Liang, Xin Tao, Pengfei Wan, Di Zhang, Yijun Li, and Ying-Cong Chen. Motion inversion for video customization. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–12, 2025. 6

[8] Zeqi Xiao, Yifan Zhou, Shuai Yang, and Xingang Pan. Video diffusion models are training-free motion interpreter and controller. *Advances in Neural Information Processing Systems*, 37:76115–76138, 2024. 6

[9] Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8466–8476, 2024. 6

[10] Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jia-Wei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. Motiondirector: Motion customization of text-to-video diffusion models. In *European Conference on Computer Vision*, pages 273–290. Springer, 2024. 6