
Temporal-aware Flow Matching for Video Generation with Temporally Coherent Motion

Zirui Pan¹ Xin Wang¹ Yipeng Zhang¹ Yuwei Zhou¹ Wenwu Zhu¹

Abstract

Despite rapid advances in text-to-video generation, state-of-the-art generative models still suffer from producing temporally incoherent and unrealistic motion for videos. The key weakness of existing works is that they commonly treat videos as frame sequences and directly adopt Flow Matching (FM) objectives, which are originally designed for images. This practice fails to explicitly model motion priors or temporal dependencies, resulting in suboptimal dynamics that may appear incoherent and unrealistic. To solve this problem, we propose Temporal-aware Flow Matching (TFM), a novel training paradigm that embeds inter-frame constraints into the flow objective, leading to temporally coherent motion modeling in video generation. More specifically, the proposed TFM enforces temporal correlations across frames while retaining the desirable properties of FM, and further introduces a residual-type loss that aligns naturally with this new flow. We theoretically prove that models trained with TFM are able to exhibit remarkably enhanced temporal perception ability. Notably, TFM imposes no additional cost during inference and is applicable to any model using FM. Extensive experiments demonstrate that our TFM can significantly improve motion realism across diverse motion types. Generated videos are presented at <https://pzrain.github.io/tfm>.

1. Introduction

Despite rapid progress in text-to-video generation (Brooks et al., 2024; Esser et al., 2024; Kodaira et al., 2025), existing

¹Department of Computer Science and Technology, BN-Rist, Tsinghua University, Beijing, China. Correspondence to: Xin Wang <xin.wang@tsinghua.edu.cn>, Wenwu Zhu <wwzhu@tsinghua.edu.cn>.

state-of-the-art approaches, even those billion-parameter models (Wan et al., 2025; Wu et al., 2025) capable of producing visually high-quality clips, still suffer from a critical problem, *i.e.*, how to generate temporally coherent and realistic motion (Hu et al., 2025) in videos. This problem is largely caused by two factors, *i.e.*, (i) the violation of basic physical laws (*e.g.*, Figure 1(a) demonstrates the failure of the baseline in rendering water splashes when feet step on the ground), and (ii) incoherent or unstable scenes under complex motion (*e.g.*, Figure 1(b) shows that the sample generated by baseline displays visible body deformations).

In this paper, we study the problem of video generation with temporally coherent motion, which poses a fundamental challenge in current video models, *i.e.*, the *inability* to effectively internalize motion priors from video data. Although motion naturally arises from the temporal relationships between adjacent frames and should, in principle, be learnable directly from large video corpora, empirical evidence suggests that this remains challenging in practice (Kang et al., 2025). Even when trained on massive video datasets, it is extremely difficult to capture coherent and physically consistent motion, thus resulting in dynamics that violate physical constraints or become unstable under complex scenarios.

To tackle this fundamental challenge, we first provide an unprecedented perspective to understand why video generative models struggle to learn motion priors and then propose a new paradigm for training general video generators. We argue that the root cause lies in the prevailing training objective used in current generative models—Flow Matching (FM) (Lipman et al., 2023; Liu et al., 2023a). Although effective for image generation, FM does not **explicitly** encode motion priors. As a result, video models are forced to rely on **implicit** attention mechanisms to infer temporal structure, which is fundamentally suboptimal for modeling videos. Originally designed for image synthesis, FM learns a **direct** probability path (flow) from noise to data. Since this framework is theoretically sound and empirically effective in the image domain, current video models that largely treat videos as sequences of images have directly extended it to video generation. However, these approaches typically reuse the original FM objective without necessarily conducting video-specific modifications (Davtyan et al., 2023;

Figure 1. Typical examples of motion incoherence generated by the base model Wan2.1-T2V-14B (Wan et al., 2025), where the model fails to produce realistic dynamics, e.g., no water splashes when stepping on the ground in (a) Fundamental Physics, and noticeable body deformation in (b) Complex Motion. In contrast, Wan2.1 equipped with TFM produces videos with coherent and text-faithful motions.

Lin et al., 2024), leaving the flow of each frame temporally independent and failing to capture the inherent temporal correlations that manifest as motion. Consequently, models trained in this way are prone to producing incoherent or physically implausible dynamics.

Motivated by these observations, we propose a novel training paradigm, i.e., Temporal-aware Flow Matching (TFM), to enhance motion modeling in video generation. Conceptually, TFM overcomes the limitations of the direct yet temporally decoupled flows in standard FM by introducing a new probability path that embeds motion priors into the flow itself. Specifically, in Section 3.2, we reform the original FM governing equation (Eq. 3), which only enforces intra-frame constraints to incorporate inter-frame constraints (Eq. 4). Compared to the flow in FM, the TFM flow introduces a training objective that explicitly captures the temporal correlations inherent across frames, making it better suited for temporally structured data such as videos. To optimize the model under this new flow, we propose a residual-type loss in Section 3.3, which is shown to be more effective than the solution-type loss used in FM. In Section 4, we further establish a theoretical foundation for TFM, proving that models trained under TFM exhibit enhanced temporal perception and are able to learn motion dynamics much better than those trained with FM. Importantly, TFM introduces no additional time or memory overhead during inference and is applicable to any video generative model that utilizes the flow-based training objective. Extensive experiments across diverse motion types for video generative models. To summarize, we make the following contributions:

- We propose a novel training paradigm, i.e., Temporal-aware Flow Matching (TFM), to enhance motion modeling in video generation. Being superior to FM, TFM explicitly captures temporal correlations, enabling models to effectively learn motion priors.
- We provide a theoretical foundation for TFM, proving that models trained under TFM yield stronger temporal perception compared to those trained with FM.

- We empirically validate the effectiveness of the proposed TFM through extensive experiments, demonstrating significant improvements in motion coherence.

2. Related Work

Flow-based Generative Models Diffusion models (Ho et al., 2022; Blattmann et al., 2023b; Rombach et al., 2022) have revolutionized video generation with their exceptional generative capabilities. Early works extended DDPM (Ho et al., 2020) to text- or image-to-video tasks, modeling temporal dynamics primarily through implicit attention mechanisms (Guo et al., 2023; Luo et al., 2023; Zhang et al., 2025a; Blattmann et al., 2023a; Chen et al., 2025; Zhang et al., 2025b). More recently, Flow Matching (Lipman et al., 2023; Liu et al., 2023a) reformulates generation as a distribution-mapping problem, providing a more robust and theoretically grounded training objective. It has been further generalized to general geometries (Chen & Lipman, 2024), discrete data (Gat et al., 2024) and high-order velocity fields (Su et al., 2025), establishing FM as the SoTA training framework for generative tasks.

Nevertheless, existing video generative models based on flow matching (HaCohen et al., 2024; Liu et al., 2025; Wang et al., 2025c; Ma et al., 2024; Wu et al., 2025; Liu et al., 2023b) remain fundamentally similar to DDPM-based approaches: they model videos as sequences of frames and learn frame-wise, direct yet independent probability paths. By constructing flows separately for each frame, these methods fail to explicitly capture temporal dependencies or motion priors, often leading to physically implausible or inconsistent motion, especially in scenes with complex dynamics or interactions (Kang et al., 2025), limiting the models' ability to produce temporally coherent video sequences.

Refined Motion in Generation To address the universal challenge of generating realistic motion, prior works have explored two main directions: (1) Physics-informed approaches assume that motion is strictly governed by physical laws. They incorporate physics-based constraints into

diffusion models, enforcing the generated distribution to satisfy underlying partial differential equations (Bastek et al., 2024). However, such methods typically operate on highly specialized domains, such as atmospheric systems (Wang et al., 2025b) or turbulent flows (Shu et al., 2023), and can hardly generalize to real-world videos, where multiple interacting physical processes coexist; (2) Externally guided approaches share our view of motion as temporal structure, and introduce additional signals as guidance. Early works leverage large language models to plan scene layouts (Lian et al., 2024; Huang et al., 2023; Hong et al., 2023), offering coarse guidance such as object movement or prompt refinement (He et al., 2025). More recent methods adopt fine-grained control signals, including optical flow (Liang et al., 2024; Nam et al., 2025; Chefer et al., 2025) and point clouds (Ren et al., 2025; LAN et al., 2025; Wang et al., 2025a), either as conditioning inputs or jointly learned cues. While effective, these strategies rely on external supervision, suggesting that existing models fail to fully exploit the motion information already inherent in video data. We argue that a generative model should be able to learn such motion priors directly from data without additional signals, provided that the training objective is properly designed.

diffusion models, enforcing the generated distribution to satisfy underlying partial differential equations (Bastek et al., 2024). However, such methods typically operate on highly specialized domains, such as atmospheric systems (Wang et al., 2025b) or turbulent flows (Shu et al., 2023), and can hardly generalize to real-world videos, where multiple interacting physical processes coexist; (2) Externally guided approaches share our view of motion as temporal structure, and introduce additional signals as guidance. Early works leverage large language models to plan scene layouts (Lian et al., 2024; Huang et al., 2023; Hong et al., 2023), offering coarse guidance such as object movement or prompt refinement (He et al., 2025). More recent methods adopt fine-grained control signals, including optical flow (Liang et al., 2024; Nam et al., 2025; Chefer et al., 2025) and point clouds (Ren et al., 2025; LAN et al., 2025; Wang et al., 2025a), either as conditioning inputs or jointly learned cues. While effective, these strategies rely on external supervision, suggesting that existing models fail to fully exploit the motion information already inherent in video data. We argue that a generative model should be able to learn such motion priors directly from data without additional signals, provided that the training objective is properly designed.

diffusion models, enforcing the generated distribution to satisfy underlying partial differential equations (Bastek et al., 2024). However, such methods typically operate on highly specialized domains, such as atmospheric systems (Wang et al., 2025b) or turbulent flows (Shu et al., 2023), and can hardly generalize to real-world videos, where multiple interacting physical processes coexist; (2) Externally guided approaches share our view of motion as temporal structure, and introduce additional signals as guidance. Early works leverage large language models to plan scene layouts (Lian et al., 2024; Huang et al., 2023; Hong et al., 2023), offering coarse guidance such as object movement or prompt refinement (He et al., 2025). More recent methods adopt fine-grained control signals, including optical flow (Liang et al., 2024; Nam et al., 2025; Chefer et al., 2025) and point clouds (Ren et al., 2025; LAN et al., 2025; Wang et al., 2025a), either as conditioning inputs or jointly learned cues. While effective, these strategies rely on external supervision, suggesting that existing models fail to fully exploit the motion information already inherent in video data. We argue that a generative model should be able to learn such motion priors directly from data without additional signals, provided that the training objective is properly designed.

diffusion models, enforcing the generated distribution to satisfy underlying partial differential equations (Bastek et al., 2024). However, such methods typically operate on highly specialized domains, such as atmospheric systems (Wang et al., 2025b) or turbulent flows (Shu et al., 2023), and can hardly generalize to real-world videos, where multiple interacting physical processes coexist; (2) Externally guided approaches share our view of motion as temporal structure, and introduce additional signals as guidance. Early works leverage large language models to plan scene layouts (Lian et al., 2024; Huang et al., 2023; Hong et al., 2023), offering coarse guidance such as object movement or prompt refinement (He et al., 2025). More recent methods adopt fine-grained control signals, including optical flow (Liang et al., 2024; Nam et al., 2025; Chefer et al., 2025) and point clouds (Ren et al., 2025; LAN et al., 2025; Wang et al., 2025a), either as conditioning inputs or jointly learned cues. While effective, these strategies rely on external supervision, suggesting that existing models fail to fully exploit the motion information already inherent in video data. We argue that a generative model should be able to learn such motion priors directly from data without additional signals, provided that the training objective is properly designed.

3. Method

In this section, we detail our method. We first review standard FM and explain why its direct application to video fails to explicitly model temporal dynamics. We then introduce TFM and describe the key implementation details required for practical use. Finally, we present intuitive examples that illustrate the temporal constraints imposed by our TFM.

3.1. Preliminaries

Flow Matching (Lipman et al., 2023; Liu et al., 2023a) is a methodology widely adopted in recent state-of-the-art generative models. It formulates generation as a transportation problem between probability distributions: the model learns a mapping that transports samples from a simple prior distribution, i.e., standard Gaussian, to the unknown data distribution. This is accomplished by fitting a probability path, i.e., flow, which in standard FM corresponds to a straight path between the two distributions. Specifically in training, given data $x_1 \in \mathbb{R}^d$, random noise $x_0 \in \mathbb{R}^d \sim \mathcal{N}(0, I)$ and a timestep $t \in [0, 1]$, the noised intermediate latent and the corresponding constant velocity is calculated as:

$$x_t = tx_1 + (1-t)x_0; \quad u_t = x_1 - x_0; \quad (1)$$

which forces the flow between the two distributions to be a direct line in latent space. The model is then optimized by:

$$J = \mathbb{E}_{x_1, x_0 \sim \mathcal{N}(0, I); t \in [0, 1]} [\|j(u(x_t; t) - u_t)\|_2^2]; \quad (2)$$

where $u(x_t; t)$ denotes the model's predicted velocity, and θ represents the learnable parameters. In its subse-

$$d(t(x_1, x_0); x_1) = (t)d(x_0, x_1); \quad (3)$$

where (t) is a monotonically decreasing function satisfying $(0) = 1$ and $(1) = 0$, and $d(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a metric function satisfying the following constraints:

1. Non-negative: $d(x; y) \geq 0$.
2. Positive: $d(x; y) = 0$ iff $x = y$.
3. Non-degenerate: $d(x; y) > 0$ iff $x \neq y$.

These constraints ensure that the flow reaches its target at $t = 1$, i.e., $x_1(x_1, x_0) = x_1$. When setting $(t) = 1 - t$, $d(\cdot, \cdot)$ the Euclidean distance, and constraining the transport direction to be $x_1 - x_0$, the solution to Eq(3) collapses precisely to the straight-constant flow used in standard FM.

Building on this formulation, we now consider its extension to video generation. Suppose we have a video consisting of n frames $x_1 = [x_1^{(1)}; x_1^{(2)}; \dots; x_1^{(n)}]$, where each frame $x_1^{(i)} \in \mathbb{R}^{c \times h \times w}$ is paired with a noise $x_0^{(i)}$ of the same shape, $i \in [1, n]$. While FM was originally proposed for images, video data introduce temporal structure, a further dimension of complexity. A video is not merely a collection of images but an ordered sequence with dependencies across time. Nevertheless, existing FM-based video models (Wan et al., 2025; Wu et al., 2025) commonly simplify this structure by treating all frames as independent and constructing a separate flow for each of them. Concretely, they define

as a linear interpolation between noise and data, and

$$x_t^{(i)} = tx_1^{(i)} + (1-t)x_0^{(i)}; \quad 1 \leq i \leq n; \quad t \in [0, 1]; \quad (4)$$

as a linear interpolation between noise and data, and

$$u_t^{(i)} = x_1^{(i)} - x_0^{(i)}; \quad 1 \leq i \leq n; \quad (5)$$

as the velocity, enforcing a frame-wise straight-constant path. Such a setup fails to explicitly model the temporal relationships across frames, which we hypothesize may impair the model's ability to capture motion priors inherent in video data and generate coherent dynamics.

3.2. Temporal-aware Flow Matching

Motivated by our hypothesis that standard FM fails to explicitly capture the temporal dynamics inherent in videos, we propose a novel training paradigm Temporal-aware Flow Matching (TFM). TFM extends the basic FM governing equation (Eq (3)) to the following system of equations:

$$\begin{aligned} & d(t(x_1^{(i)}; x_1^{(i)}); x_1^{(i)}) + \sum_{i>1} d(t(x_1^{(i-1)}; x_1^{(i-1)}); t(x_1^{(i)}; x_1^{(i)})) \\ & + \sum_{i<n} d(t(x_1^{(i)}; x_1^{(i)}); t(x_1^{(i+1)}; x_1^{(i+1)})) \\ & = (t)d(x_0^{(i)}; x_1^{(i)}) \\ & + \sum_{i>1} [(t)d(x_0^{(i-1)}; x_0^{(i)}) + (1-t)d(x_1^{(i-1)}; x_1^{(i)})] \\ & + \sum_{i<n} [(t)d(x_0^{(i)}; x_0^{(i+1)}) + (1-t)d(x_1^{(i)}; x_1^{(i+1)})]; \end{aligned} \quad (6)$$

its corresponding ground-truth velocity \mathbf{v}_{eld} . While Theorem 3.2 allows us to efficiently compute each $\mathbf{u}_t(x|x_1^{(i)})$ given the latents $x|x_1^{(i)}$, the latents themselves do not admit a closed-form expression. In fact, except for special cases, such as the straight-constant flow in FM or certain interpolation-based curves, most non-trivial flows lack analytical solutions. Consequently, the latent trajectories must be obtained by numerically integrating the ODE system.

In practice, we find that a simple Euler discretization suffices to produce accurate latent trajectories. Moreover, the additional computational overhead incurred by this integration step is marginal relative to the overall optimization cost. Further analysis is provided in Appendix D.3.

Optimizing the model We disentangle the target velocity into magnitude and direction, and define the training objectives respectively:

$$J(\theta) = \mathbb{E}_{t;x_1;x_0} [\|\mathbf{u}_t(x) - \mathbf{C}_{jj} \frac{\dot{x}}{2}\|_2^2] + \mathbb{E}_{t;x_1;x_0;1 \leq i \leq n} [\cos(\text{sim}(\mathbf{u}_t(x|x_1^{(i)}); \mathbf{x}_1^{(i)} - \mathbf{x}_0^{(i)}))]; \quad (12)$$

For the direction term, we use cosine similarity, and for the magnitude term we adopt a residual-type loss, denoted as $\text{loss}^{\text{TFM}} = \mathbb{E}_{jj} \|\mathbf{u}_t(x) - \mathbf{C}_{jj} \frac{\dot{x}}{2}\|_2^2$. An alternative is to solve the system to obtain the unique solution $\mathbf{u}_t(x)$, as in standard FM, and define a solution-type loss $\text{loss}^{\text{FM}} = \mathbb{E}_{jj} \|\mathbf{u}_t(x) - \mathbf{u}_t(x)\|_2^2$. We next explain why loss^{TFM} is more effective compared to loss^{FM} in our setting.

For a well-trained model, the predicted velocity should closely follow the probability path. To measure deviations, we define the equation-residual indicator:

$$(\mathbf{u}) = \mathbb{E}_{jj} \|\mathbf{u} - \mathbf{C}_{jj} \frac{\dot{x}}{2}\|_2^2; \quad (13)$$

A smaller (\mathbf{u}) indicates better adherence to the TFM equations. Our loss^{TFM} is constructed to directly minimize (\mathbf{u}) . Although both losses share the same global minimum, they differ in how they rank intermediate predictions. To clarify this, we introduce the notion:

Definition 3.3. (Consistent objective) Let $L(\theta)$ be a training objective for model parameters $\theta \in \mathcal{M}$. We call L a consistent objective (with respect to (\mathbf{u})) if and only if

$$L(\theta_1) < L(\theta_2), (\mathbf{u}(\theta_1)) < (\mathbf{u}(\theta_2)); \quad (14)$$

for all parameter choices; $\theta_1, \theta_2 \in \mathcal{M}$.

In other words, a consistent objective must produce the same ranking of model parameters as (\mathbf{u}) . By design, loss^{TFM} is consistent. In contrast, the following proposition holds:

Proposition 3.4. loss^{FM} is not a consistent objective.

Proposition 3.4 implies that loss^{FM} may assign equal penalties to two predictions even if one satisfies the TFM equations substantially better. Thus, although both losses share

Figure 2. Illustration of flow interactions (dashed lines) under FM and TFM using simple 2D data. The middle sub figures show only two frames. In the left and right sub figures, we add and shift a third frame, observing how the flow changes accordingly.

a global minimum, only loss^{TFM} aligns with the probability path principle central to TFM. In the standard FM setting, the system reduces to independent equations, and loss^{TFM} will degenerate to the standard loss^{FM} .

3.4. Discussion

In this section, we discuss the connection between TFM and standard FM, together with the numerical stability of the proposed optimization process. First, to provide an intuitive comparison between FM and TFM, we visualize their induced probability paths on simple 2D data in Figure 2. In the middle sub figures, we consider two flows $x_0^{(1)} \rightarrow x_1^{(1)}$ and $x_0^{(2)} \rightarrow x_1^{(2)}$, and connect their intermediate states $x_t^{(1)}$ and $x_t^{(2)}$ at each timestep using dashed lines to illustrate their interactions. Under FM, the velocity remains constant, resulting in evenly spaced dashed lines. In contrast, under TFM, the trajectory accelerates after the intersection point, leading to non-uniform spacing.

We then add a third flow $x_0^{(3)} \rightarrow x_1^{(3)}$ at different locations (shown in the left and right sub figures of Figure 2) and again draw the interactions between the first two flows. We observe that introducing a third frame alters the TFM trajectories: it directly affects the flow $x_0^{(2)} \rightarrow x_1^{(2)}$ and indirectly influences $x_0^{(1)} \rightarrow x_1^{(1)}$. When the third frame is placed on the left, $x_0^{(2)} \rightarrow x_1^{(2)}$ begins with a higher velocity; when placed on the right, its initial velocity slows, which all rigorously follow Eq(6) mathematically. In contrast, FM trajectories remain unchanged regardless of where the third frame is placed, highlighting the absence of explicit temporal constraints. Further discussions on the connection between FM and TFM are provided in Appendix D.2.

In addition, despite these additional temporal interactions, TFM remains numerically stable throughout training. In our formulation, the matrix \mathbf{M} (Eq (11)) is tridiagonal, diagonally dominant, and has bounded entries $\|\mathbf{M}_{ij}\|_2 + 2 = \|\mathbf{M}\|_2$. These properties ensure that the operator norm of

and M^T remain bounded, which directly constrains the C (Eq (11)) can be simplified to ($1 < i < n$):

gradient of the magnitude term $\mu_j = 2 M^T (\mu_j - C)$

(Eq (12)) and prevents uncontrolled gradient explosion.

Meanwhile, diagonal dominance guarantees that the eigenvalues of $M^T M$ stay bounded away from zero, mitigating gradient vanishing across different directions. Combined

with the tridiagonal structure, which avoids long-range amplification or attenuation effects, these properties ensure that gradients remain well-conditioned during optimization.

4. Theoretical Analysis

In this section, we theoretically investigate why our method is effective for video data by analyzing its temporal perception ability. We first make some assumptions. Let all frames in a video clip $\{x_1^{(1)}; x_1^{(2)}; \dots; x_1^{(n)}\}$ be sampled from the same (unknown) distribution D , i.e., $x_1^{(i)} \sim D$ for $1 \leq i \leq n$, consistent with standard FM. We further impose the following three assumptions:

Assumption 4.1. The first-order differences $x_1^{(i)} - x_1^{(i-1)}$, for $2 \leq i \leq n$ follow the same unknown distribution. Consequently $\{x_1^{(i)}; x_1^{(i-1)}\}$, for $2 \leq i \leq n$ all follow the same distribution, denoted $D^{(1)}$.

Assumption 4.2. The second-order differences $x_1^{(i)} - x_1^{(i-2)}$, for $3 \leq i \leq n$ follow the same unknown distribution. Consequently $\{x_1^{(i)}; x_1^{(i-2)}\}$, for $3 \leq i \leq n$ all follow the same distribution, denoted $D^{(2)}$.

Assumption 4.3. The two distributions introduced in Assumption 4.1 and 4.2 are distinct, i.e., $D^{(1)} \neq D^{(2)}$.

Assumption 4.1 states that the first-order temporal differences, corresponding to changes between consecutive frames, share a common unknown distribution $D^{(1)}$, which may be highly related to motion information, since motion can be interpreted as the temporal evolution of natural images. Assumption 4.2 extends this to second-order differences, assumed to follow $D^{(2)}$. Finally, Assumption 4.3 reflects the natural expectation that $D^{(1)}$ and $D^{(2)}$ are distinct, as differences between consecutive frames generally differ from those separated by larger temporal gaps. We note that these assumptions, which generally follow those in standard FM, are applicable to natural videos.

We then hypothesize that the model is trained to fit different distributions based on their training objectives. For FM, the training objective is simple, i.e., $\mathcal{L}_{FM}(\text{Predict}_{FM;i}^t(x_j|x_1^{(i)})) = \mathbb{E}_{x_1^{(i)} \sim x_1^{(i)}; x_0^{(i)} \sim D} \mathbb{E}_{(x,y);(k,z) \sim H} [e^{itx} e^{ity} e^{-itk} e^{-itz}]^2$; therefore, the prediction made by model trained under FM is expected to follow the distribution:

$$D_{\text{predict}}^{FM}, \mathcal{L}[\text{Predict}_{FM}^t(x_j|x_1^{(i)})] \stackrel{d}{=} D \sim \mathcal{N}(0; 1); \quad (15)$$

where $\mathcal{L}[\cdot]$ represents the distribution (law) of a random variable. While for TFM, the training objective $\mathcal{L}_{TFM}(\text{Predict}_{TFM;i}^t(x_j|x_1^{(i)})) =$

$$\mathcal{L}[\text{Predict}_{TFM;i}^t(x_j|x_1^{(i)})]$$

$$= \mathbb{E}_{x_1^{(i)} \sim D} \mathbb{E}_{(x,y);(k,z) \sim H} [u(-t(x_j|x_1^{(i)})); u(-t(x_j|x_1^{(i)})); u(-t(x_j|x_1^{(i+1)}))]$$

$$= \mathbb{E}_{x_1^{(i)} \sim D} [d(x_0^{(i)}; x_1^{(i)}) + d(x_1^{(i)}; x_1^{(i+1)}) + d(x_1^{(i+1)}; x_1^{(i)})]$$

$$= \mathbb{E}_{x_1^{(i)} \sim D} [g_i(x) + d(x_1^{(i)}; x_1^{(i)})];$$

$$(16)$$

where $f_i(\cdot; \cdot; \cdot)$ is a linear combination of the three adjacent frame velocities $g_i(x)$, $d(x_0^{(i)}; x_1^{(i)}) + d(x_1^{(i)}; x_1^{(i+1)})$, and we exclude the constant terms w.r.t the standard gaussian distribution that are independent of the rest in C .

Consequently, the prediction made by model trained under TFM is expected to be the following distribution:

$$D_{\text{predict}}^{TFM}, \mathcal{L}[\text{Predict}_{TFM;i}^t(x_j|x_1^{(i)})] \stackrel{d}{=} H_{Y;D^{(1)}}; \quad (17)$$

$$H_{Y;D^{(1)}}(z) = \int h_{g_i(x);d(x_1^{(i-1)}; x_1^{(i)})}(a; z - a) da; \quad (18)$$

where $h_{g_i(x);d(x_1^{(i-1)}; x_1^{(i)})}(\cdot; \cdot)$ is the joint distribution of $g_i(x)$ and $\{x_1^{(i-1)}; x_1^{(i)}\}$, with marginal distributions Y and $D^{(1)}$, respectively.

Based on the above assumptions, we can compare and measure a model's sensitivity to temporal coherence under different training objectives. In particular, perturbing the temporal structure of the data induces a training distribution D_{train} that deviates from the model's fitted distribution D_{predict} . A model that effectively leverages temporal information should detect such deviations.

Without loss of generality, we consider a simple perturbation in which the positions of two consecutive frames are swapped, i.e., exchanging $x_1^{(i)}$ and $x_1^{(i+1)}$ for a selected $1 < i < n$, which is generalizable, as any permutation can be decomposed into a sequence of such pairwise swaps. This swap introduces temporal incoherence and should be detectable by a well-trained model capable of capturing motion-related dynamics. To measure such effect, we introduce KL divergence $\mathcal{G}_{KL}(D_{\text{predict}} \| D_{\text{train}})$ and present the following Theorem.

Theorem 4.4. For FM and TFM,

$$\mathcal{G}_{KL}^{TFM} > 0 = \mathcal{G}_{KL}^{FM}; \quad (19)$$

and moreover,

$$\mathcal{G}_{KL}^{TFM} = \frac{1}{2} \sup_{t \in \mathbb{R}} \mathbb{E}_{((x,y);(k,z)) \sim H} [e^{itx} e^{ity} e^{-itk} e^{-itz}]^2; \quad (20)$$

where H represents the joint distribution of $Y; D^{(1)}$ and $H_{Y;D^{(2)}}$.

The lower bound can be easily estimated through numerical simulation. We perform a bootstrap resampling of 10^4

Figure 3. Qualitative comparison between TFM and the baselines. We select several intermediate frames for the convenience of presentation. We provide more generated results in Appendix G, and the corresponding video samples in supplementary materials.

simulated data and derive a mean lower bound of 0.0868 with 95% confidence interval [0.0739,0.1095].

5. Experiments

We conduct extensive qualitative and quantitative experiments against state-of-the-art open-source video models to demonstrate the effectiveness of our method in producing temporally coherent motion. We perform ablation studies to validate the contribution of each proposed component. We then discuss the time complexity of our method and its temporal perception ability. Additional results are provided in Appendix E and Appendix G.

Implementation Details We adopt the pre-trained

as our base generative model and further re-tune it using TFM on approximately 40k data from the public ShareGPT4Video dataset (Chen et al., 2024), which accounts for only 0.01% of the original pre-training data. The model is configured to generate 81-frame videos at a fixed resolution of 832 × 480 with fps = 15. Further implementation details are provided in Appendix B.

Evaluation Details We compare our method with the following baselines: (1) CogVideoX1.5-5B (Yang et al., 2024), (2) HunyuanVideo-13B (Wu et al., 2025), and (3) Wan2.1-T2V-14B (Wan et al., 2025), all of which are recent and competitive video generation models. We evaluate performance on two benchmarks: (1) VideoJam-Bench (Chefer et al., 2025), which emphasizes challenging motion, and (2) Movie-Gen (Polyak et al., 2024), a general-purpose video

Table 1. Quantitative comparison between TFM and the baselines on VideoJam-Bench. User study shows percentage of votes favoring TFM. See breakdown of VBench metrics in Appendix G.

Method	Auto. Metric		User Study		
	Appearance	Motion	Semantic	Quality	Motion
CogVideoX1.5	0.750	0.894	87.8%	85.4%	82.9%
HunyuanVideo	0.752	0.898	65.6%	56.2%	68.8%
Wan2.1	0.780	0.888	82.8%	85.7%	74.3%
TFM	0.783	0.961	-	-	-

Table 2. Quantitative comparison between TFM and the baselines on Movie-Gen. User study shows percentage of votes favoring TFM. See breakdown of VBench metrics in Appendix G.

Method	Auto. Metric		User Study		
	Appearance	Motion	Semantic	Quality	Motion
CogVideoX1.5	0.783	0.801	90.9%	86.3%	88.6%
HunyuanVideo	0.775	0.806	76.9%	53.8%	88.5%
Wan2.1	0.791	0.805	58.3%	62.5%	64.6%
TFM	0.795	0.858	-	-	-

generation benchmark. We adopt two types of evaluation metrics: (1) automatic metrics supported by VBench (Huang et al., 2024), and (2) human evaluations following the two alternative forced-choice (2AFC) protocol, where raters compare two videos (one generated by TFM and the other by a baseline) and select the preferred one across multiple aspects, following prior work (Blattmann et al., 2023a). Further evaluation details are provided in Appendix C.

5.1. Main Results

Qualitative Experiment In Figure 3, we compare TFM with baseline methods across a variety of motion types. These include intricate motions requiring physical understanding (the left column), such as flipping a cast-iron skillet, where mushrooms are expected to be propelled into the air and fall back smoothly, and slicing a roll, where the knife should cut cleanly through the roll to separate a slice. We also evaluate complex human motions (the right column), such as a sprinter running or a figure skater dancing, which pose significant challenges for video generative models. From the results, we observe that baseline methods generally fail to respect basic physical laws. For example, in Figure 3(a), Wan2.1 generates mushrooms flying upward, while the skillet itself remains static. Similarly, in Figure 3(c), CogVideoX and HunyuanVideo produce physically implausible motions, where the knife appears to pass through a finger or the roll as if the objects were intangible. For complex human motions (Figure 3(c) and (d)), baseline models frequently generate deformed body structures. A common failure mode is that different body parts (e.g., legs or arms) move independently of the torso, resulting in unnatural poses where the upper body and lower body face

Figure 4. Ablation study. We ablate the choice of training loss (i.e., residual-type loss^{TFM} vs. solution-type loss^{FM}) and the flow formulation used during fine-tuning (i.e., standard FM vs. our proposed TFM). Evaluation is conducted on Movie-Gen using VBench metrics. All results are reported as ratios relative to the full TFM model; exact statistics are provided in Appendix G.

different directions. In contrast, models trained with TFM generate more coherent and physically plausible motions across a wide range of scenarios. These results demonstrate TFM’s ability to capture motion priors from training data effectively, leading to improved temporal consistency.

Quantitative Experiment In Tables 1 and 2, we quantitatively compare TFM with baseline methods. For fairness, all models are evaluated using a fixed random seed and a single run on both benchmarks. Overall, TFM consistently outperforms the baselines across the two benchmarks. On general metrics (Appearance, Semantic, and Quality), TFM achieves performance that is competitive with or superior to its base model and other state-of-the-art methods. More importantly, TFM achieves substantial improvements on the motion metric, with a 74.3% higher preference rate on VideoJam-Bench and a 64.6% higher preference rate on the Movie-Gen benchmark compared to its base model. It also beats other baseline methods in terms of motion coherence. These quantitative improvements corroborate the qualitative observations, demonstrating that TFM more effectively models motion dynamics and enhances temporal control in generated videos. Moreover, they indicate that improving motion modeling does not come at the expense of other objectives, such as visual quality or semantic coherence.

5.2. Ablation Study

In this section, we conduct an ablation study to assess the contribution of each component in our method. Specifically, we first fine-tune the base model using standard FM (denoted as FM-flow) under the same training settings and on the same data as TFM, thereby isolating the effect of the proposed flow formulation. Next, we ablate the choice of loss function by training TFM with the solution-type loss^{FM} , while keeping all other settings unchanged. The results are

Table 3. Comparison of time costs between FM and TFM. We consider two metrics: (1) Net Time Cost, which measures the time spent on flow calculation, velocity prediction, and loss computation, corresponding to lines 17–18 in Algorithm 1 for FM, and lines 28–29 in Algorithm 2 for TFM; and (2) Overall Time Cost, which records the total time for a single optimization step for each model. We report the mean time cost, with standard errors indicated in parentheses.

Model	Net Time Cost (FM)	Net Time Cost (TFM)	Overall Time Cost (FM)	Overall Time Cost (TFM)
Wan2.1-T2V-1.3B	1.65s(0.33s)	2.85s(0.37s) +72.7%	18.71s(0.9s)	19.66s(0.47s) +5.1%
Wan2.1-T2V-14B	10.67s(0.85s)	12.98s(0.39s) +21.6%	60.51s(0.78s)	61.92s(0.98s) +2.3%

rm that TFM introduces only a negligible time overhead, thanks to the efficient optimization algorithm and necessary simplifications in the formulas.

5.4. Temporal Perception Ability

In Section 4, we establish the theoretical basis for why TFM better suits video data than standard FM, showing that explicit temporal constraints enhance a model’s ability to perceive temporal structure. Here, we further illustrate this claim through an intuitive quantitative experiment. Specifically, for each video, we apply noise to a chosen timestep $t \in \{50; 100; \dots; 900\}$ out of 1000 total steps, and then randomly permute the video latents along the temporal dimension. For every timestep, we simulate $1k$ samples.

Figure 5. Quantitative comparison of the temporal perception abilities of FM and TFM. The solid line shows the mean loss-increase ratio after randomly permuting video frames, and the shaded region indicates the corresponding standard deviation.

summarized in Figure 4, where the blue curve corresponds to the full TFM model. From the results, we observe that the full TFM consistently outperforms both variants, particularly on Dynamic Degree, which directly reflects the magnitude of motion. In contrast, the two ablated variants exhibit relatively high Motion Smooth, likely due to their tendency to generate more static or under-dynamic motions. Importantly, TFM also maintains strong performance on general appearance-related metrics. Overall, these findings demonstrate that both the temporal flow formulation and the residual-type loss design are essential for achieving realistic and dynamic motion without compromising visual quality.

A model with strong temporal perception should exhibit a larger loss increase after permutation, whereas a model with weak temporal understanding will struggle to distinguish a correctly ordered video from a permuted one.

We therefore compute the mean and standard deviation of the relative loss change before and after permutation, and present the results in Figure 5. For FM (the blue line), we use the pre-trained Wan2.1-T2V-1.3B model, while for TFM (the red line), we use the same model re-tuned with TFM. As shown, the FM-trained model displays only limited temporal sensitivity, while TFM substantially boosts the model’s ability to detect temporal disruptions. This experiment further validates both the effectiveness of TFM and the importance of explicitly modeling temporal constraints.

5.3. Time Complexity Analysis

In this section, we demonstrate that the computational overhead introduced by our method is marginal relative to the overall optimization cost. The results are presented in Table 3, where we compare time costs between FM and TFM across models of different scales using $1k$ data. Although TFM exhibits a seemingly large increase in Net Time Cost for Wan2.1-T2V-1.3B, the overhead is minimal when considering the total time per optimization step, amounting to only 5.1%. For the larger Wan2.1-T2V-14B model, this relative increase reduces further to 2.3%. These results con-

6. Conclusion

In this paper, we introduce Temporal-aware Flow Matching (TFM), a method that enhances motion generation by explicitly modeling temporal structure in video data. Unlike prior approaches that rely on implicit attention mechanisms, TFM provides a principled flow formulation for capturing motion dynamics. We theoretically show that TFM improves temporal perception ability and empirically demonstrate consistent performance gains across benchmarks. As a model-agnostic approach with no additional inference cost, TFM enables more effective motion modeling for video generation.

Acknowledgements

This work is supported by the National Key Research and Development Program of China under Grant No.2022ZD0115903, National Natural Science Foundation of China No.62222209, Beijing National Research Center for Information Science and Technology under Grant No.BNR2026TD03005.

Impact Statement

This work aims to improve motion modeling in video generation by enhancing temporal coherence through Temporal-aware Flow Matching, enabling generative models to produce more consistent and realistic motion. As with other advances in generative video technologies, the proposed method could potentially be misused to create misleading visual content; however, it does not introduce new risks beyond those inherent to existing video generation models. We believe that continued efforts in responsible deployment, bias detection, and misuse mitigation are essential to ensure the safe and fair use of generative video systems, including those based on our approach.

References

- Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.
- Bastek, J.-H., Sun, W., and Kochmann, D. M. Physics-informed diffusion models. arXiv preprint arXiv:2403.14404, 2024.
- Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127, 2023a.
- Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S. W., Fidler, S., and Kreis, K. Align your latents: High-resolution video synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 22563–22575, 2023b.
- Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Ha, J., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., et al. Video generation models as world simulators. OpenAI Blog, 1(8):1, 2024.
- Chefer, H., Singer, U., Zohar, A., Kirstain, Y., Polyak, A., Taigman, Y., Wolf, L., and Sheynin, S. Videojam: Joint appearance-motion representations for enhanced motion generation in video models. arXiv preprint arXiv:2502.02492, 2025.
- Chen, H., Wang, X., Zeng, G., Zhang, Y., Zhou, Y., Han, F., Wu, Y., and Zhu, W. Videoreamer: Customized multi-subject text-to-video generation with disen-mix netuning on language-video foundation models. IEEE Transactions on Multimedia, 2025.
- Chen, L., Wei, X., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Tang, Z., Yuan, L., et al. Sharegpt4video: Improving video understanding and generation with better captions. Advances in Neural Information Processing Systems, 37:19472–19495, 2024.
- Chen, R. T. Q. and Lipman, Y. Flow matching on general geometries. In The Twelfth International Conference on Learning Representations, 2024. URL <https://openreview.net/forum?id=g7ohDITL>.
- Cramer, A. L., Wu, H.-H., Salamon, J., and Bello, J. P. Listen, and learn more: Design choices for deep audio embeddings. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3852–3856. IEEE, 2019.
- Csiszár, I. and Körner, J. Information theory: coding theorems for discrete memoryless systems. Cambridge University Press, 2011.
- Davtyan, A., Sameni, S., and Favaro, P. Efficient video prediction via sparsely conditioned flow matching. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 23263–23274, October 2023.
- Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In Forty-first international conference on machine learning, 2024.
- Gat, I., Remez, T., Shaul, N., Kreuk, F., Chen, R. T., Synnaeve, G., Adi, Y., and Lipman, Y. Discrete flow matching. Advances in Neural Information Processing Systems, 37:133345–133385, 2024.
- Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., Agrawala, M., Lin, D., and Dai, B. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. arXiv preprint arXiv:2307.04725, 2023.
- HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D., Richardson, E., Levin, E., Shiran, G., Zabari, N., Gordon, O., et al. Ltx-video: Realtime video latent diffusion. arXiv preprint arXiv:2501.00103, 2024.
- He, H., Yang, H., Tuo, Z., Zhou, Y., Wang, Q., Zhang, Y., Liu, Z., Huang, W., Chao, H., and Yin, J. Dreamstory: Open-domain story visualization by llm-guided multi-subject consistent diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2025.

- Ho, J. and Salimans, T. Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598, 2022.
- Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303, 2022.
- Hong, S., Seo, J., Shin, H., Hong, S., and Kim, S. Direct2v: Large language models are frame-level directors for zero-shot text-to-video generation. arXiv preprint arXiv:2305.14330, 2023.
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- Hu, Y., Wang, L., Liu, X., Chen, L.-H., Guo, Y., Shi, Y., Liu, C., Rao, A., Wang, Z., and Xiong, H. Simulating the real world: A unified survey of multimodal generative models. arXiv preprint arXiv:2503.04641, 2025.
- Huang, H., Feng, Y., Shi, C., Xu, L., Yu, J., and Yang, S. Free-bloom: Zero-shot text-to-video generator with llm director and ldm animator. *Advances in Neural Information Processing Systems*, 36:26135–26158, 2023.
- Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.
- Hung, C.-Y., Majumder, N., Kong, Z., Mehrish, A., Zadeh, A., Li, C., Valle, R., Catanzaro, B., and Poria, S. Tan-go-ux: Super fast and faithful text to audio generation with flow matching and clap-ranked preference optimization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=qgNs5NmQB7>.
- Kang, B., Yue, Y., Lu, R., Lin, Z., Zhao, Y., Wang, K., Huang, G., and Feng, J. How far is video generation from world model: A physical law perspective. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J. (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 28991–29017. PMLR, 13–19 Jul 2025. URL <https://proceedings.mlr.press/v267/kang25g.html>.
- Kodaira, A., Hou, T., Hou, J., Georgopoulos, M., Juefei-Xu, F., Tomizuka, M., and Zhao, Y. Streamdit: Real-time streaming text-to-video generation. arXiv preprint arXiv:2507.03745, 2025.
- LAN, Y., Zhou, S., Lyu, Z., Hong, F., Yang, S., Dai, B., Pan, X., and Loy, C. C. Gaussiananything: Interactive point cloud flow matching for 3d generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=P4DbTSDQFu>.
- Lee, W. Tridiagonal matrices: Thomas algorithm. MS6021, Scientific Computation, University of Limerick, 1054, 2011.
- Lian, L., Shi, B., Yala, A., darrell, t., and Li, B. Llm-grounded video diffusion models. In Kim, B., Yue, Y., Chaudhuri, S., Fragkiadaki, K., Khan, M., and Sun, Y. (eds.), *International Conference on Representation Learning*, volume 2024, pp. 50207–50227, 2024.
- Liang, F., Wu, B., Wang, J., Yu, L., Li, K., Zhao, Y., Misra, I., Huang, J.-B., Zhang, P., Vajda, P., et al. Flowvid: Taming imperfect optical flows for consistent video-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8207–8216, 2024.
- Lin, B., Ge, Y., Cheng, X., Li, Z., Zhu, B., Wang, S., He, X., Ye, Y., Yuan, S., Chen, L., et al. Open-sora plan: Open-source large video generation model. arXiv preprint arXiv:2412.00131, 2024.
- Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=PqvMRDCJT9t>.
- Liu, J., Liu, G., Liang, J., Yuan, Z., Liu, X., Zheng, M., Wu, X., Wang, Q., Xia, M., Wang, X., et al. Improving video generation with human feedback. arXiv preprint arXiv:2501.13918, 2025.
- Liu, X., Gong, C., and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023a. URL <https://openreview.net/forum?id=XVjTT1nw5z>.
- Liu, X., Zhang, X., Ma, J., Peng, J., and Liu, Q. Insta-ow: One step is enough for high-quality diffusion-based text-to-image generation. arXiv preprint arXiv:2309.06380, 2023b.
- Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017.

- Luo, Z., Chen, D., Zhang, Y., Huang, Y., Wang, L., Shen, Y., Su, M., Hu, J. Y.-C., Lee, Y.-C., Zhu, N., Chung, J.-H., Zhao, D., Zhou, J., and Tan, T. Videofusion: Decomposed diffusion models for high-quality video generation. arXiv preprint arXiv:2303.08320, 2023.
- Ma, N., Goldstein, M., Albergo, M. S., Bof, N. M., Vanden-Eijnden, E., and Xie, S. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In European Conference on Computer Vision, pp. 23–40. Springer, 2024.
- Mei, X., Meng, C., Liu, H., Kong, Q., Ko, T., Zhao, C., Plumbley, M. D., Zou, Y., and Wang, W. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 32:3339–3354, 2024.
- Nam, H., Kim, J., Lee, D., and Ye, J. C. Optical-flow guided prompt optimization for coherent video generation. In Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 7837–7846, 2025.
- Peebles, W. and Xie, S. Scalable diffusion models with transformers. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 4195–4205, 2023.
- Polyak, A., Zohar, A., Brown, A., Tjandra, A., Sinha, A., Lee, A., Vyas, A., Shi, B., Ma, C.-Y., Chuang, C.-Y., et al. Movie gen: A cast of media foundation models. arXiv preprint arXiv:2410.13720, 2024.
- Ren, X., Shen, T., Huang, J., Ling, H., Lu, Y., Nimier-David, M., Müller, T., Keller, A., Fidler, S., and Gao, J. Gen3c: 3d-informed world-consistent video generation with precise camera control. In Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 6121–6132, 2025.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10684–10695, 2022.
- Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pp. 234–241. Springer, 2015.
- Shu, D., Li, Z., and Farimani, A. B. A physics-informed diffusion model for high-fidelity flow field reconstruction. Journal of Computational Physics, 478:111972, 2023.
- Wu, S., Song, Z., Chen, M., and Liu, H. High-order flow matching: Unified framework and sharp statistical rates. In Belgrave, D., Zhang, C., Lin, H., Pascanu, R., Koniusz, P., Ghassemi, M., and Chen, N. (eds.), Advances in Neural Information Processing Systems, volume 38, pp. 45820–45932, 2025.
- Thomas, L. H. Elliptic problems in linear difference equations over a network. Watson Sci. Comput. Lab. Rept., Columbia University, New York, 1:71, 1949.
- Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. Towards accurate generative models of video: A new metric & challenges, 2019. URL <https://arxiv.org/abs/1812.01717>.
- Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.-W., Chen, D., Yu, F., Zhao, H., Yang, J., et al. Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314, 2025.
- Wang, C., Chen, C., Huang, Y., Dou, Z., Liu, Y., Gu, J., and Liu, L. Physctrl: Generative physics for controllable and physics-grounded video generation. In The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025a. URL <https://openreview.net/forum?id=AHEKhff4Oa>.
- Wang, H., Han, J., Fan, W., Zhang, W., and Liu, H. Phyda: Physics-guided diffusion models for data assimilation in atmospheric systems. arXiv preprint arXiv:2505.12882, 2025b.
- Wang, Q., Luo, Y., Shi, X., Jia, X., Lu, H., Xue, T., Wang, X., Wan, P., Zhang, D., and Gai, K. Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation. In Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers, pp. 1–10, 2025c.
- Wu, B., Zou, C., Li, C., Huang, D., Yang, F., Tan, H., Peng, J., Wu, J., Xiong, J., Jiang, J., et al. Hunyuanvideo 1.5 technical report. arXiv preprint arXiv:2511.18870, 2025.
- Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al. Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072, 2024.
- Zhang, D. J., Wu, J. Z., Liu, J.-W., Zhao, R., Ran, L., Gu, Y., Gao, D., and Shou, M. Z. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. International Journal of Computer Vision, 133(4):1879–1893, 2025a.

Zhang, Y., Wang, X., Chen, H., Qin, C., Hao, Y., Mei, H., and Zhu, W. Scenariodiff: text-to-video generation with dynamic transformations of scene conditions. *International Journal of Computer Vision*, 133(7):4909–4922, 2025b.

First, we prove that the solution $\mathbf{x}(t)$ to Eq (23) generates the flow defined by Eq (6). Differentiate the left side of Eq (6) for any $1 \leq i \leq n$ with respect to time gives:

$$\begin{aligned} & \frac{d}{dt} [d(\mathbf{x}_t^{(i)}; \mathbf{x}_1^{(i)}) + I_{i>1} d(\mathbf{x}_t^{(i-1)}; \mathbf{x}_1^{(i)}) + I_{i<n} d(\mathbf{x}_t^{(i)}; \mathbf{x}_1^{(i+1)})] \\ &= [A_i + I_{i>1} B_{i,i-1} + I_{i<n} B_{i,i+1}]^T \mathbf{u}(\mathbf{x}_t^{(i)}) + I_{i>1} B_{i-1,i}^T \mathbf{u}(\mathbf{x}_t^{(i-1)}) + I_{i<n} B_{i+1,i}^T \mathbf{u}(\mathbf{x}_t^{(i+1)}) \\ &= \mathbf{C}_i; \end{aligned} \quad (25)$$

Thus the function $K_i(t) = d(\mathbf{x}_t^{(i)}; \mathbf{x}_1^{(i)}) + I_{i>1} d(\mathbf{x}_t^{(i-1)}; \mathbf{x}_1^{(i)}) + I_{i<n} d(\mathbf{x}_t^{(i)}; \mathbf{x}_1^{(i+1)})$ satisfies the ODE:

$$\frac{d}{dt} K_i(t) = \mathbf{C}_i; \quad (26)$$

with initial condition

$$K_i(0) = d(\mathbf{x}_0^{(i)}; \mathbf{x}_1^{(i)}) + I_{i>1} d(\mathbf{x}_0^{(i-1)}; \mathbf{x}_0^{(i)}) + I_{i<n} d(\mathbf{x}_0^{(i)}; \mathbf{x}_0^{(i+1)}); \quad (27)$$

where by default we start sampling from noise $\mathbf{x}_0 = \mathbf{f}_x(\mathbf{x}_0^{(1)}; \mathbf{x}_0^{(2)}; \dots; \mathbf{x}_0^{(n)})$. The general solution to the ODE defined by Eq (26) has the form

$$K_i(t) = \int_0^t \mathbf{C}_i dt + \mathbf{C}_i^0; \quad (28)$$

where \mathbf{C}_i^0 is a constant constrained by initial conditions. Substitute Eq (27) into Eq (28), we can verify that

$$\begin{aligned} K_i(t) &= \int_0^t d(\mathbf{x}_s^{(i)}; \mathbf{x}_1^{(i)}) ds + \\ & I_{i>1} \int_0^t [d(\mathbf{x}_s^{(i-1)}; \mathbf{x}_0^{(i)}) + (1-t) d(\mathbf{x}_s^{(i-1)}; \mathbf{x}_1^{(i)})] ds + \\ & I_{i<n} \int_0^t [d(\mathbf{x}_s^{(i)}; \mathbf{x}_0^{(i+1)}) + (1-t) d(\mathbf{x}_s^{(i)}; \mathbf{x}_1^{(i+1)})] ds; \end{aligned} \quad (29)$$

which is exactly the right side of Eq (6).

Conversely, suppose the flow defined by Eq (6) is $\mathbf{x}_t = \mathbf{f}_x(\mathbf{x}_t^{(1)}; \mathbf{x}_t^{(2)}; \dots; \mathbf{x}_t^{(n)})$, where $\mathbf{x}_t^{(i)} = d(\mathbf{x}_t^{(i)}; \mathbf{x}_1^{(i)})$, for any $1 \leq i \leq n$. Differentiating both side of Eq (6) we get:

$$\begin{aligned} & \frac{d}{dt} [d(\mathbf{x}_t^{(i)}; \mathbf{x}_1^{(i)}) + I_{i>1} d(\mathbf{x}_t^{(i-1)}; \mathbf{x}_1^{(i)}) + I_{i<n} d(\mathbf{x}_t^{(i)}; \mathbf{x}_1^{(i+1)})] \\ &= [A_i + I_{i>1} B_{i,i-1} + I_{i<n} B_{i,i+1}]^T \mathbf{x}_t^{(i)} + I_{i>1} B_{i-1,i}^T \mathbf{x}_t^{(i-1)} + I_{i<n} B_{i+1,i}^T \mathbf{x}_t^{(i+1)}; \end{aligned} \quad (30)$$

and

$$\begin{aligned} & \frac{d}{dt} [d(\mathbf{x}_t^{(i)}; \mathbf{x}_1^{(i)}) + I_{i>1} [d(\mathbf{x}_t^{(i-1)}; \mathbf{x}_0^{(i)}) + (1-t) d(\mathbf{x}_t^{(i-1)}; \mathbf{x}_1^{(i)})] + \\ & I_{i<n} [d(\mathbf{x}_t^{(i)}; \mathbf{x}_0^{(i+1)}) + (1-t) d(\mathbf{x}_t^{(i)}; \mathbf{x}_1^{(i+1)})]] \mathbf{g} \\ &= \mathbf{C}_i; \end{aligned} \quad (31)$$

Combining the two equations above, we get

$$[A_i + I_{i>1} B_{i,i-1} + I_{i<n} B_{i,i+1}]^T \mathbf{x}_t^{(i)} + I_{i>1} B_{i-1,i}^T \mathbf{x}_t^{(i-1)} + I_{i<n} B_{i+1,i}^T \mathbf{x}_t^{(i+1)} = \mathbf{C}_i; \quad (32)$$

which is exactly $M \mathbf{x} = \mathbf{C}$. \square

In practice, we set $t = 1 - t$, use the Euclidean distance $d(\mathbf{x}; \mathbf{y})$, and constrain each direction $d(\mathbf{x}_t^{(i)}; \mathbf{x}_1^{(i)})$ to align with $\mathbf{x}_1^{(i)} - \mathbf{x}_0^{(i)}$. Under these choices, the system reduces to an ODE of the form $\dot{\mathbf{M}} \mathbf{x} = \mathbf{C}$ (Eq (11)), where $M \in \mathbb{R}^{n \times n}$ becomes a tridiagonal matrix, with

$$\begin{aligned} & M_{i,i-1} = \mathbf{v}^{(i-1;1)T} \mathbf{v}^{(i-1)}; i > 1; \\ & M_{i,i} = I_{i>1} \mathbf{v}^{(i-1;1)T} \mathbf{v}^{(i)} + I_{i<n} \mathbf{v}^{(i+1;1)T} \mathbf{v}^{(i)}; 1 \leq i \leq n; \\ & M_{i,i+1} = \mathbf{v}^{(i+1;1)T} \mathbf{v}^{(i+1)}; i < n; \end{aligned} \quad (33)$$

Proof. Recall that we perturb the temporal structure of the data by swapping $x_1^{(i)}$ and $x_1^{(i+1)}$, for a selected $2 < i < n$. For FM, its training objective changes to $D_{\text{FM};i}^{\text{Train}} = D_{\text{FM}}^{\text{Train}}(t(x_j|x_1^{(i+1)})) = x_1^{(i+1)} x_0^{(i)} \stackrel{d}{=} x_1^{(i)} x_0^{(i+1)}$. Therefore, the same $D_{\text{train}}^{\text{FM}} \stackrel{d}{=} D_{N(0;1)} \stackrel{d}{=} D_{\text{predict}}^{\text{FM}}$ is obtained, thus $G_{\text{KL}}^{\text{FM}} = D_{\text{KL}}(D_{\text{predict}}^{\text{FM}} || D_{\text{train}}^{\text{FM}}) = 0$.

For TFM, swapping $x_1^{(i)}$ and $x_1^{(i+1)}$ will change its training objective to

$$\begin{aligned} D_{\text{TFM};i}^{\text{Train}}(t(x_j|x_1)) &= d(x_0^{(i)}; x_1^{(i+1)}) + d(x_1^{(i+1)}; x_1^{(i+1)}) + d(x_1^{(i+1)}; x_1^{(i)}) \\ &\stackrel{d}{=} d(x_0^{(i+1)}; x_1^{(i+1)}) + d(x_1^{(i+1)}; x_1^{(i+1)}) + d(x_1^{(i+1)}; x_1^{(i)}) \\ &= d(x_0^{(i+1)}; x_1^{(i+1)}) + d(x_1^{(i+1)}; x_1^{(i)}) + d(x_1^{(i+1)}; x_1^{(i+1)}) \\ &\stackrel{d}{=} d(x_0^{(i)}; x_1^{(i)}) + d(x_1^{(i)}; x_1^{(i+1)}) + d(x_1^{(i)}; x_1^{(i+2)}) \\ &= g_i(x) + d(x_1^{(i)}; x_1^{(i+2)}): \text{ (If } i+2 > n; \text{ we can just assume an } x_1^{(n+1)} \text{ which extends the video.)} \end{aligned} \quad (40)$$

Therefore, we have

$$D_{\text{train}}^{\text{TFM}}, L[D_{\text{TFM};i}^{\text{Train}}(t(x_j|x_1))] \stackrel{d}{=} H_{Y;D(2)}: \quad (41)$$

According to Assumption 4.3, we have $D_{\text{predict}}^{\text{TFM}} \stackrel{d}{=} H_{Y;D(1)} = H_{Y;D(2)} \stackrel{d}{=} D_{\text{train}}^{\text{TFM}}$, and consequently,

$$G_{\text{KL}}^{\text{TFM}} = D_{\text{KL}}(D_{\text{predict}}^{\text{TFM}} || D_{\text{train}}^{\text{TFM}}) = D_{\text{KL}}(H_{Y;D(1)} || H_{Y;D(2)}) > 0: \quad (42)$$

Next we will derive a lower bound of $G_{\text{KL}}^{\text{TFM}}$:

$$\begin{aligned} G_{\text{KL}}^{\text{TFM}} &= D_{\text{KL}}(D_{\text{predict}}^{\text{TFM}} || D_{\text{train}}^{\text{TFM}}) \\ &\geq 2 \text{TV}^2(D_{\text{predict}}^{\text{TFM}}; D_{\text{train}}^{\text{TFM}}) \\ &= \frac{1}{2} \int H_{Y;D(1)}(t) - H_{Y;D(2)}(t) j^2 \\ &= \frac{1}{2} \sup_{t \in \mathbb{R}} |H_{Y;D(1)}(t) - H_{Y;D(2)}(t)|^2; \end{aligned} \quad (43)$$

where the first inequality is the direct result of Pinsker's inequality (Csiszr & K6rner, 2011), and $\text{TV}(\cdot; \cdot)$ represents the total variance distance function. $H_{Y;D(1)}$ and $H_{Y;D(2)}$ are the characteristic functions of distributions $D_{Y;D(1)}$ and $D_{Y;D(2)}$. The second inequality is due to the fact that for any distribution p and q ,

$$\begin{aligned} \text{TV}(p; q) &= \frac{1}{2} \int |p(x) - q(x)| dx \\ &= \frac{1}{2} \int | \int e^{itx} (p(x) - q(x)) dx | \\ &= \frac{1}{2} \int | \int e^{itx} (p(x) - q(x)) dx | = \frac{1}{2} \int |p(t) - q(t)|; \quad t \in \mathbb{R} \end{aligned} \quad (44)$$

If we substitute the definition of $H_{Y;D(1)}$ and $H_{Y;D(2)}$ into Eq (43), we get

$$G_{\text{KL}}^{\text{TFM}} \geq \frac{1}{2} \sup_{t \in \mathbb{R}} E_{((x;y);(k;z)) \sim H} [e^{itx} e^{ity} - e^{itk} e^{itz}]^2; \quad (45)$$

where H represents the joint distribution of $D_{Y;D(1)}$ and $D_{Y;D(2)}$. \square

B. Implementation Details

In this section, we provide implementation details of TFM to facilitate reproducibility. The code will be released upon publication. While the core ideas of our method have been fully described in the main paper and can be readily implemented on most video generative models, we further present the training procedure of TFM in Algorithm 2, together with its FM counterpart in Algorithm 1 for comparison.

Both algorithms share the same MAIN FUNCTION, in which the training process is abstracted into two stages, namely GETLATENT and GETLOSS. Notably, for FM, the training pipeline is identical to that used in image generation and does not explicitly account for temporal structure. In contrast, TFM constructs flows that explicitly model temporal dynamics, thereby providing more appropriate supervision for video generative models.

Algorithm 1 Training procedure of FM under text-to-video modeling.

```

1: Input: Video data:  $x_1 = x_1^{(1)} \dots x_1^{(n)}$ ,
   paired text prompt:  $fhg$ .
2: Parameter:  $\theta$ : parameters for the base video generative model.

3: function GETVELOCITY( $x_0; x_1; x_t; t$ )
4:   return  $x_1 - x_0$ .
5: end function

6: function GETLATENT( $x_0; x_1; t$ )
7:   return  $t x_1 + (1 - t) x_0$ .
8: end function

9: function GETLOSS( $x_0; x_1; x_t; t; h$ )
10:   $v \leftarrow$  GETVELOCITY( $x_0; x_1; x_t; t$ ).
11:  return  $\frac{1}{2} \|v - \dot{x}_t\|_2^2$ .
12: end function
    
```

BEGIN MAIN FUNCTION:

```

13: repeat
14:   Forward data  $f_x; h_g$ .
15:   Sample  $\gamma \sim N(0, 1)$ .
16:   Sample  $t \sim U(0, 1)$ .
17:    $x_t \leftarrow$  GETLATENT( $x_0; x_1; t$ ).
18:   loss  $\leftarrow$  GETLOSS( $x_0; x_1; x_t; t; h$ ).
19:    $\arg \min$  (loss).
20: until Converged
    
```

Algorithm 2 Training procedure of TFM under text-to-video modeling.

```

1: Input: Video data:  $x_1 = x_1^{(1)} \dots x_1^{(n)}$ ,
   paired text prompt:  $fhg$ .
2: Parameter:  $\theta$ : parameters for the base video generative model.

3: function GETVELOCITY( $x_0; x_1; x_t; t$ )
4:   Compute  $M$  and  $C$  according to Eq (11).
5:   Solve  $u_t(x)$  from  $M u_t(x) = C$ , using Thomas algorithm.
6:   return  $u_t(x) \cdot (x_1 - x_0)$ . Magnitude Direction
7: end function

8: function GETLATENT( $x_0; x_1; t$ )
9:    $x_t \leftarrow x_0$ .
10:  Uniformly sample  $t_0 = t_1 < \dots < t_{T+1} = t_g$ .
11:  for 1  $i \leq T$  do Euler-step to solve ODE
12:     $v \leftarrow$  GETVELOCITY( $x_0; x_1; x_t; t$ ).
13:     $x_t \leftarrow x_{t_{i-1}} + (t_i - t_{i-1}) v$ .
14:  end for
15:  return  $x_t$ .
16: end function

17: function GETLOSS( $x_0; x_1; x_t; t; h$ )
18:  Compute  $M$  and  $C$  according to Eq (11).
19:   $loss_M \leftarrow \frac{1}{2} \|M^{-1} C\|_2^2$ .
20:   $loss_D \leftarrow \cos \text{sim}(\dot{\phi}(x_t; h), \dot{x}_t)$ .
21:  return  $loss_M + loss_D$ .
22: end function
    
```

BEGIN MAIN FUNCTION:

```

23: Initialize hyper-parameter  $\beta = 4:0$ .
24: repeat
25:   Forward data  $f_x; h_g$ .
26:   Sample  $\gamma \sim N(0, 1)$ .
27:   Sample  $t \sim U(0, 1)$ .
28:    $x_t \leftarrow$  GETLATENT( $x_0; x_1; t$ ).
29:   loss  $\leftarrow$  GETLOSS( $x_0; x_1; x_t; t; h$ ).
30:    $\arg \min$  (loss).
31: until Converged
    
```

Figure 6. Algorithmic comparison of the training procedures for FM and TFM in text-to-video modeling. The two methods share the same overall training pipeline but differ in the formulation of velocity and latent (i.e., the flow), as well as in the design of the loss function. Notably, the formulation of FM for video generation is identical to that used for image generation, as it does not impose any constraints on temporal relationships between video frames. In contrast, TFM introduces explicit temporal control, making it more suitable for modeling video data.

B.1. Model and Hyper-parameters

Our base model is built upon Wan2.1-T2V-14B (Wan et al., 2025) widely adopted and state-of-the-art open-sourced foundational video generation model. Wan2.1 employs the Diffusion Transformer (DiT) (Peebles & Xie, 2023) architecture, which has increasingly replaced U-Net (Ronneberger et al., 2015) as the dominant backbone for video generation. Given a text prompt, Wan2.1 incorporates the textual information via cross-attention and generates a corresponding video sequence. The model adopts full spatio-temporal attention to effectively capture complex motion and temporal dependencies.

Wan2.1 is trained using standard Flow Matching (FM) on billions of video-text pairs. However, its training objective does not impose explicit constraints on temporal dynamics. We refer readers to their official technical report (<https://arxiv.org/pdf/2503.20314>, Section 4) for further details.

Importantly, our proposed TFM is model-agnostic and does not modify the architecture or hyper-parameters of the base video generative model. TFM introduces only one additional hyper-parameter α (Eq (6)), which controls the relative strength between the intra-frame and inter-frame terms. In practice, we set α to ensure a unique solution and numerical stability of the Thomas algorithm. Larger values of α increase the weight of the intra-frame term, thereby weakening temporal constraints; in the limit as $\alpha \rightarrow 1$, TFM degenerates to standard FM.

B.2. Training and Inference

During training, we solve an ordinary differential equation (ODE) to obtain the latent at time t (Algorithm 2, line 11). Each ODE step is set to 0.002, resulting in fewer than 500 integration steps for 2s [0, 1]. We evaluate the stability of the ODE integration of \mathbf{z}_t and observe an integration error on the order of 10^{-3} . We fine-tune the model on the publicly available ShareGPT4Video dataset (Chen et al., 2024) one epoch, which contains 40k videos paired with text captions annotated by GPT-4V (Achiam et al., 2023). According to the Wan2.1 technical report (<https://arxiv.org/pdf/2503.20314>, Abstract), the base model is trained on billions of samples. Therefore, our fine-tuning dataset accounts for roughly 0.01% of the pre-training data, making the fine-tuning stage extremely light-weight.

We fix the number of frames to 81 and the spatial resolution to 336 × 480. Fine-tuning is performed using LoRA (Hu et al., 2022) with rank 32. We adopt the AdamW (Loshchilov & Hutter, 2017) optimizer with a learning rate of 10^{-4} and weight decay of 0.01. Training is conducted on four A100 GPUs with 80GB memory, resulting in a total batch size of 4, and completes in approximately one week.

During inference, we fix the video length and resolution to match the training configuration for both our method and all baselines in Section 5.1. Since CogVideoX1.5 does not support custom output resolutions, its generated videos are spatially resized to the target resolution for fair comparison. We use 50 inference steps and fix the classifier-free guidance scale (Ho & Salimans, 2022) to 5.0. No cherry-picking is performed: all results are generated with a fixed random seed. Inference is carried out on a single A100 GPU with 40GB memory.

C. Evaluation Details

C.1. Baselines

We compare our method with the following baselines, which are latest works for text-to-video generation that achieves superior performance. Specifically,

1. CogVideoX1.5-5B (Yang et al., 2024): <https://huggingface.co/zai-org/CogVideoX1.5-5B>.
2. HunyuanVideo-13B (Wu et al., 2025): <https://huggingface.co/tencent/HunyuanVideo>.
3. Wan2.1-T2V-14B (Wan et al., 2025): <https://huggingface.co/Wan-AI/Wan2.1-T2V-14B>.

C.2. Benchmarks

To comprehensively evaluate the performance of our method in terms of both motion generation and overall video quality, we adopt two publicly available benchmarks, each serving a distinct evaluation purpose:

¹<https://huggingface.co/Wan-AI/Wan2.1-T2V-14B>

²<https://huggingface.co/datasets/ShareGPT4Video/ShareGPT4Video>

1. VideoJam-Bench (Chefer et al. (2025), <https://arxiv.org/pdf/2502.02492>), Appendix F): VideoJam-Bench is specially designed to assess motion generation capabilities, covering a wide range of dynamic scenarios, including complex human actions and physics-driven motions. The benchmark consists of 128 carefully curated text prompts.
2. MovieGen Bench (Polyak et al. (2024), <https://github.com/facebookresearch/MovieGenBench>): MovieGen Bench is a general-purpose video evaluation benchmark that evaluates overall generation quality across diverse content. It contains approximately 1,000 prompts, each annotated with motion intensity levels (low, medium, and high). For computational efficiency, we randomly sample 500 prompts spanning all motion levels.

C.3. Metrics

The evaluation metrics we adopt can be classified into two categories:

1. Automatic Metrics: The automatic metrics are supported by VBench (Huang et al., 2024), spanning six sub-categories, i.e., motion-related as Motion Smoothness and Dynamic Degree, and appearance-related as Subject Consistency, Background Consistency, Aesthetic Quality and Imaging Quality. We give the full VBench evaluated results in Table 8 and Table 9. While in the main paper, we aggregate the motion-related and appearance-related metrics respectively, following the procedure specified in the official website of VBench (<https://github.com/Vchitect/VBench>). Concretely, each metric is first normalized then weighted averaged, resulting in the following formulas:

$$\begin{aligned} \text{Motion} &= \frac{1}{1.5} [(\text{Motion_Smoothness} \cdot 0.706) + (0.9975 \cdot 0.706) + 0.5 \cdot \text{DynamicDegree}] \\ \text{Appearance} &= \frac{1}{4} [(\text{SubjectConsistency} \cdot 0.1462) + (1.0 \cdot 0.1462) + \text{AestheticQuality} \\ &\quad + (\text{BackgroundConsistency} \cdot 0.2615) + (1.0 \cdot 0.2615) + \text{ImagingQuality}] \end{aligned} \quad (46)$$

Note that in ablation study (Figure 4), we also use the normalized VBench metrics.

2. User Study: We conduct a user study following the two-alternative forced-choice (2AFC) protocol. In each trial, participants are presented with two videos (one generated by TFM and the other generated by a baseline method), and are asked to select their preferred video under three criteria: overall quality, motion quality, and semantic coherence. The video pairs are randomly sampled to avoid ordering bias. We recruit 40 participants, each of whom evaluates 10 video pairs. All participants are university students with computer science-related backgrounds, though not necessarily specializing in generative AI. The preference rate of TFM is then computed for each evaluation criterion by aggregating the selections across all participants.

D. Additional Discussions

D.1. Formulation of TFM

To motivate the formulation of TFM, we first recall the governing equation (Eq (3)) of standard FM:

$$d_t(x_j x_1^{(i)}); x_1^{(i)} = (t) d_x x_0^{(i)}; x_1^{(i)}; \quad 1 \leq i \leq n; \quad (47)$$

which only involves an intra-frame term. Consequently, standard FM treats each frame independently and does not explicitly model temporal correlations across frames. To make the flow temporally aware, it is therefore necessary to introduce inter-frame terms that impose temporal constraints, which we interpret as motion prior.

The simplest and most natural form of inter-frame interaction is the connection between adjacent frames, i.e., between $x_j x_1^{(i-1)}$ and $x_j x_1^{(i)}$ for $t \in [0; 1]$ and $1 < i \leq n$. A minimal way to characterize such an adjacent-frame dependency is through the distance

$$d_t(x_j x_1^{(i-1)}); x_j x_1^{(i)}; \quad (48)$$

A natural extension of Eq (47) is therefore to add this inter-frame term to its left-hand side. However, a naive addition is insufficient, as the flow must still satisfy the prescribed boundary conditions $x_0 = f x_0^{(1)}; \dots; x_0^{(n)}$ and $x_1 = f x_1^{(1)}; \dots; x_1^{(n)}$ should be a valid solution, and at $t = 1$, $x_1 = f x_1^{(1)}; \dots; x_1^{(n)}$ should be the unique solution.

Enforcing these boundary conditions determines the corresponding term on the right-hand side of Eq(47), which takes the form of an interpolation between inter-frame distances at $t = 0$ and $t = 1$. This leads to

$$\begin{aligned} & d(t; \mathbf{x}_1^{(i)}; \mathbf{x}_1^{(i)}) + \mathbb{1}_{i>1} d(t; \mathbf{x}_1^{(i-1)}; \mathbf{x}_1^{(i)}) \\ = & (t) d(\mathbf{x}_0^{(i)}; \mathbf{x}_1^{(i)}) + \mathbb{1}_{i>1} ((t) d(\mathbf{x}_0^{(i-1)}; \mathbf{x}_0^{(i)} + (1-t) d(\mathbf{x}_1^{(i-1)}; \mathbf{x}_1^{(i)}); 1 - \mathbb{1}_{i=1}) \end{aligned} \quad (49)$$

This equation is essentially the governing equation of TFM (Eq(6)), except that in practice, we further extend it by considering not only the connection to the preceding frame $(\mathbf{x}_1^{(i-1)}; \mathbf{x}_1^{(i)})$, but also the connection to the succeeding frame $(\mathbf{x}_1^{(i)}; \mathbf{x}_1^{(i+1)})$. Taken together, these observations indicate that TFM provides a direct and minimal characterization of temporal correlations in video data, while preserving the boundary conditions and structure of flow formulation.

D.2. When Does TFM Degenerate to Standard FM?

In this section, we examine the relationship between TFM and standard FM. In particular, since we have already shown that TFM models a flow that is conceptually different from standard FM, we aim to characterize the conditions under which TFM degenerates to standard FM.

Recall that the solution for intermediate latent to standard FM is given by $\mathbf{x}_1^{(i)}(t) = (1-t)\mathbf{x}_1^{(i)} + (t)\mathbf{x}_0^{(i)}$, for $t \in [0, 1]$ and $1 \leq i \leq n$, which satisfies Eq(47). For any finite n , substituting this solution into the governing equation of TFM (Eq (6)) yields

$$\begin{aligned} & \mathbb{1}_{i>1} d(t; \mathbf{x}_1^{(i-1)}; \mathbf{x}_1^{(i)}) + \mathbb{1}_{i \leq n} d(t; \mathbf{x}_1^{(i)}; \mathbf{x}_1^{(i+1)}) \\ = & \mathbb{1}_{i>1} [(t) d(\mathbf{x}_0^{(i-1)}; \mathbf{x}_0^{(i)}) + (1-t) d(\mathbf{x}_1^{(i-1)}; \mathbf{x}_1^{(i)})] + \mathbb{1}_{i \leq n} [(t) d(\mathbf{x}_0^{(i)}; \mathbf{x}_0^{(i+1)}) + (1-t) d(\mathbf{x}_1^{(i)}; \mathbf{x}_1^{(i+1)})] \end{aligned} \quad (50)$$

On the other hand, we have

$$\begin{aligned} d(t; \mathbf{x}_1^{(i-1)}; \mathbf{x}_1^{(i)}) = & d((1-t)\mathbf{x}_1^{(i-1)} + (t)\mathbf{x}_0^{(i-1)}; (1-t)\mathbf{x}_1^{(i)} + (t)\mathbf{x}_0^{(i)}) \\ = & (1-t) d(\mathbf{x}_1^{(i-1)}; \mathbf{x}_1^{(i)}) + (t) d(\mathbf{x}_0^{(i-1)}; \mathbf{x}_0^{(i)}); 1 < i \leq n \end{aligned} \quad (51)$$

Combining Eq(51) with Eq(50), we observe that the inequality in Eq(51) must in fact become an equality. This occurs if and only if the two vectors $\mathbf{x}_1^{(i-1)} - \mathbf{x}_1^{(i)}$ and $\mathbf{x}_0^{(i-1)} - \mathbf{x}_0^{(i)}$ are collinear and share the same direction. Applying this condition to all $1 \leq i \leq n$, we conclude that TFM degenerates to FM only when, for every frame i in the video, the vectors $\mathbf{x}_1^{(i-1)} - \mathbf{x}_1^{(i)}$ and $\mathbf{x}_0^{(i-1)} - \mathbf{x}_0^{(i)}$ are collinear and aligned.

Conversely, consider the scenario in which, for all $1 \leq i \leq n$, the vectors $\mathbf{x}_1^{(i-1)} - \mathbf{x}_1^{(i)}$ and $\mathbf{x}_0^{(i-1)} - \mathbf{x}_0^{(i)}$ have the same direction. First, we note that such a condition is almost surely violated in real-world video data. Second, under this highly restrictive scenario, it is straightforward and intuitive that TFM degenerates to FM, since the direction and magnitude of the flow are identical for each frame, and the temporal coupling imposed by TFM introduces no additional constraints beyond those already captured by standard FM.

D.3. Time Complexity Analysis

In TFM, we introduce additional computations to model a more complex probability path, which is more suitable for capturing temporal dynamics in video data. Importantly, these additional computations are incurred only during training and do not introduce any time or memory overhead at inference. We first analyze the time complexity of TFM from a theoretical perspective. We focus on the GETLATENT function in Algorithm 2, which accounts for the majority of the additional computational cost.

Specifically, the GETLATENT function repeatedly invokes GETVELOCITY within a for loop over iterations. The GETVELOCITY function itself consists of two components:

1. Compute \mathbf{M} and \mathbf{C} : According to Eq(11), the calculation of \mathbf{M} and \mathbf{C} are $O(nd)$, with n the total number of frames, and d the dimension of latent feature. (Note that \mathbf{M} is sparse, and we only need to calculate elements along the tri-diagonal)

2. Solve $\mathbf{u}_t(x)$: Using Thomas algorithm (Thomas, 1949), we can efficiently solve the tri-diagonal system in $\mathcal{O}(n)$ time, since $M \in \mathbb{R}^{n \times n}$ is diagonally dominant.

Consequently, the overall time complexity of the proposed TFM is $\mathcal{O}(nd + Tn)$, compared to $\mathcal{O}(nd)$ for FM. The additional computational cost mainly arises from the iterative loops, which are necessary because the derived ODE does not admit a closed-form solution. We further note that, except for special cases such as the straight flow used in standard FM, most non-trivial flow formulations lack analytical solutions and therefore require numerical ODE solvers. In Section 5.3, we quantitatively show that TFM incurs only a modest additional training cost compared to standard FM, while introducing no extra computational overhead during inference.

D.4. Decomposition of Velocity in Loss Function

In Section 3.3, we introduce a residual-type loss $\mathcal{L}_{\text{res}}^{\text{TFM}}$, in contrast to the solution-type loss $\mathcal{L}_{\text{sol}}^{\text{FM}}$ used in standard FM. A key design choice underlying this formulation is the decomposition of the target velocity into two components: direction and magnitude. This decomposition is directly motivated by the structure of TFM and the properties of its governing equations.

First, in the TFM formulation, the direction of each frame-wise velocity is fixed by construction, i.e., it is aligned with $x_1^{(i)} - x_0^{(i)}$, while only the magnitudes are coupled through the ODE system. Decomposing the velocity accordingly and assigning separate loss terms to direction and magnitude therefore mirrors the intrinsic structure of the dynamics.

Second, this separation improves optimization stability. Direction prediction is naturally formulated as a normalized similarity objective, which is insensitive to scale, whereas magnitude prediction focuses exclusively on satisfying the linear constraints imposed by the TFM equations, without being confounded by angular errors. Designing distinct losses for direction and magnitude thus leads to a more stable and well-conditioned optimization process, and empirically facilitates effective training of TFM. In ablation study (Figure 4, Table 10 and Table 11), we quantitatively compare the performance of model under different choices of loss functions.

D.5. Selection of

In our formulation, the hyper-parameter β controls the relative strength between the intra-frame and inter-frame terms. We theoretically establish that TFM is well-defined, i.e., the flow reaches the clean sample at $t = 1$, when $\beta \geq 4$. In practice, we set $\beta = 4.0$ to ensure a unique solution and maintain numerical stability.

Table 4. Mean absolute error of velocities between the TFM flow and standard FM flow with respect to different values of β .

Value of β	4	5	6	8	10	100
MAE	41.5	35.0	30.4	24.0	16.8	2.2

However, increasing β places greater emphasis on the intra-frame term, thereby weakening the temporal constraints; in the limit as $\beta \rightarrow 1$, TFM degenerates to standard FM.

To further illustrate this behavior, we conduct an additional experiment (shown in Table 4) in which we measure the mean absolute error between the per-frame velocity derived by TFM (with different values of β) and the constant velocity in standard FM. As β increases, this difference decreases monotonically, indicating that the flow gradually approaches the independent formulation of standard FM, which does not explicitly capture temporal dynamics.

E. Additional Experiments

E.1. TFM across Model Scales and Architectures

In the main experiments presented in Section 5.1, we apply TFM to the Wan2.1-T2V-14B model, one of the largest and widely used open-source text-to-video models in the community. However, our method is agnostic to both model size and architecture. To demonstrate this, we conduct an additional experiment by applying TFM to the substantially smaller Wan2.1-T2V-1.3B model and a different backbone model HunyuanVideo-13B. The experimental settings are the same as in the main experiment.

Specifically, we quantitatively compare the two base models with their TFM-tuned counterparts on two benchmarks using VBench automatic metrics. The results are summarized in Table 5 and Table 6. As shown, TFM consistently outperforms the corresponding base models across both benchmarks, with particularly pronounced improvements on motion-related metrics. For instance, on VideoJam-Bench, the overall motion score increases from 0.829 to 0.874 for Wan2.1-T2V-1.3B after

Table 5. Quantitative comparison between TFM and its base model Wan2.1-T2V-1.3B on the two benchmarks, using VBench metrics.

Benchmark	Method	Motion			Appearance				
		Motion Smoothness	Dynamic Degree	Overall.	Subject Consistency	Background Consistency	Aesthetic Quality	Imaging Quality	Overall.
VideoJam-Bench	Wan2.1-1.3B	0.989(0.008)	0.547(0.498)	0.829	0.945(0.041)	0.947(0.029)	0.570(0.077)	0.606(0.093)	0.759
	TFM-1.3B	0.993(0.005)	0.653(0.498)	0.874	0.958(0.035)	0.952(0.025)	0.544(0.071)	0.646(0.092)	0.768
Movie-Gen	Wan2.1-1.3B	0.991(0.008)	0.460(0.439)	0.805	0.963(0.042)	0.961(0.028)	0.574(0.079)	0.659(0.120)	0.784
	TFM-1.3B	0.994(0.007)	0.490(0.392)	0.821	0.974(0.036)	0.964(0.026)	0.587(0.087)	0.666(0.112)	0.793

Table 6. Quantitative comparison between TFM and its base model HunyuanVideo on the two benchmarks, using VBench metrics.

Benchmark	Method	Motion			Appearance				
		Motion Smoothness	Dynamic Degree	Overall.	Subject Consistency	Background Consistency	Aesthetic Quality	Imaging Quality	Overall.
VideoJam-Bench	HunyuanVideo	0.986(0.015)	0.773(0.419)	0.898	0.931(0.053)	0.945(0.027)	0.570(0.072)	0.595(0.110)	0.752
	TFM-Hunyuan	0.987(0.012)	0.843(0.498)	0.924	0.943(0.086)	0.931(0.040)	0.577(0.093)	0.609(0.129)	0.756
Movie-Gen	HunyuanVideo	0.990(0.009)	0.470(0.499)	0.806	0.941(0.059)	0.952(0.032)	0.596(0.082)	0.642(0.108)	0.775
	TFM-Hunyuan	0.991(0.016)	0.575(0.396)	0.843	0.954(0.053)	0.978(0.032)	0.584(0.087)	0.648(0.125)	0.787

applying TFM, and from 0.898 to 0.924 for HunyuanVideo, corresponding to relative gains of 5.4% and 2.9%, respectively. Meanwhile, the TFM- netuned models maintain competitive or even superior overall appearance scores.

These results demonstrate the effectiveness, practicality, and scalability of the proposed TFM across models of varying scales, i.e., from compact to large-scale, as well as across different model architectures. Nevertheless, we note that TFM introduces a more challenging training objective than the direct flow formulation used in standard FM. As a result, higher-capacity models are generally better positioned to fully exploit the benefits of TFM and to accurately learn flow representations that are more suitable for video data.

E.2. Generality beyond Video Generation

Given the inherently temporal nature of TFM, we discuss in this section its potential extension to other temporal domains, such as audio.

Specifically, we conduct a preliminary experiment by applying TFM to TangoFlux (Hung et al., 2026), a recent text-to-audio generation framework built upon the standard FM pipeline, which enables a straightforward integration of TFM. We train TangoFlux with both FM and TFM on 50k audio samples from the WavCaps (Mei et al., 2024) dataset under identical settings, and observe consistent improvements with TFM in terms of FD (Cramer et al., 2019) and CLAP score on the validation set (Table 7). Although this experiment remains preliminary, the results provide encouraging evidence for the potential cross-domain applicability of our method.

Table 7. Quantitative comparison of TFM and standard FM on text-to-audio generation.

	FD(#)	CLAP(%)
TangoFlux-FM	75.88	0.202
TangoFlux-TFM	68.24	0.226

E.3. Trade off between Sample Quality and Inference Efficiency

In this section, we analyze the possible trade-off between sample quality and inference efficiency for TFM and FM. Specifically, we investigate whether TFM requires more inference steps to generate high-quality videos, given that its probability path is more complex. For efficiency, we conduct this experiment using the 1.3B base model and measure video quality using F chet Video Distance (FVD) (Unterthiner et al., 2019).

As shown in Figure 7, FVD generally decreases with more inference steps for both FM and TFM. Importantly, TFM does not exhibit noticeable quality degradation with fewer steps compared to standard FM, demonstrating that it can achieve high-quality video generation efficiently. We argue that, although the flow in TFM is more challenging to learn, its explicit modeling of motion priors aligns well with the underlying video dynamics, enabling effective generation without requiring additional inference steps.

Figure 7. Comparison of generation quality in terms of Fréchet Video Distance (FVD) between TFM and FM with respect to the number of inference steps on VideoJam-Bench. FM denotes the base Wan2.1-T2V-1.3B model, while TFM denotes the version fine-tuned with our proposed Temporal-aware Flow Matching, as in Table 5.

Figure 8. Failed cases. Our method may struggle with highly complex physical processes, including object fracture (e.g., a vase breaking on impact) and chain reactions (e.g., dominoes).

F. Limitations and Future Work

While TFM provides a more principled theoretical foundation for modeling video data and yields significant improvements in motion generation, it still exhibits several limitations, as illustrated in Figure 8. Empirically, TFM may struggle with highly complex physical processes, such as object fracture (e.g., a vase breaking upon impact) and long-range chain reactions (e.g., domino cascades). Typical failure modes include violations of physical laws and visually ambiguous or incoherent motion patterns, limitations that are also prevalent in most existing models trained with standard FM objectives.

We attribute these shortcomings primarily to the fact that, although TFM is the first method to explicitly introduce temporal constraints into the flow formulation, to the best of our knowledge, it currently adopts only the simplest and most natural mechanism for incorporating temporal control. This leaves substantial room for future explorations, such as modeling richer and more complex probability paths that better capture motion priors inherent in video data. In addition, the fine-tuning data used by TFM is extremely limited, accounting for only approximately 0.01% of the original pre-training data used by the base model. Owing to computational constraints and the lack of large-scale, high-quality public video-text datasets, we are unable to implement TFM under more extensive fine-tuning regimes. Nevertheless, even under this light-weight fine-tuning setting, TFM consistently outperforms existing methods.

Finally, we note that TFM emphasizes intrinsic temporal correlations induced by data structure rather than video-specific assumptions. As a result, the proposed formulation is not restricted to the video domain and can be naturally extended to other forms of temporally structured data, such as audio. A preliminary study on text-to-audio generation is provided in Appendix E.2.

G. Complete Tables and Additional Figures

In this section, we provide additional tables and figures that complement the results presented in the main paper. Specifically,

- Table 8: A detailed breakdown of the VBench metrics on VideoJam-Bench, including standard errors, corresponding to Table 1 in the main paper.
- Table 9: A detailed breakdown of the VBench metrics on Movie-Gen, including standard errors, corresponding to Table 2 in the main paper.
- Table 10 and Table 11: The exact numerical statistics underlying the ablation study in Figure 4, reported in Table 10 for VideoJam-Bench and Table 11 for Movie-Gen.
- Figure 9: Additional qualitative comparison between TFM and baseline methods.
- Figure 10: Additional generated samples of TFM.

Furthermore, we provide generated video samples in the supplementary material. For ease of evaluation, we have created a webpage (<https://pzrain.github.io/tfm>) showcasing the generated samples and qualitative comparisons with baseline methods.

Table 8. Breakdown of VBench metrics with standard errors corresponding to the quantitative comparison between TFM and baseline methods on VideoJam-Bench in Table 1. The top and second top performances have been bolded or underlined respectively.

Method	Motion			Appearance				
	Motion Smoothness	Dynamic Degree	Overall.	Subject Consistency	Background Consistency	Aesthetic Quality	Imaging Quality	Overall.
CogVideoX	0.974(0.021)	<u>0.844</u> (0.363)	0.894	0.924(0.044)	0.952(0.023)	0.536(0.078)	0.620(0.113)	0.750
Hunyuan	0.986(0.015)	0.773(0.419)	<u>0.898</u>	0.931(0.053)	0.945(0.027)	0.570(0.072)	0.595(0.110)	0.752
Wan2.1	<u>0.990</u> (0.008)	0.716(0.500)	0.888	<u>0.953</u> (0.029)	<u>0.955</u> (0.022)	0.591(0.071)	<u>0.648</u> (0.097)	<u>0.780</u>
TFM	0.991(0.009)	0.930(0.495)	0.961	0.958(0.032)	0.956(0.021)	<u>0.590</u> (0.074)	0.653(0.097)	0.783

Table 9. Breakdown of VBench metrics with standard errors corresponding to the quantitative comparison between TFM and baseline methods on Movie-Gen in Table 2. The top and second top performances have been bolded or underlined respectively.

Method	Motion			Appearance				
	Motion Smoothness	Dynamic Degree	Overall.	Subject Consistency	Background Consistency	Aesthetic Quality	Imaging Quality	Overall.
CogVideoX1.5	0.982(0.020)	<u>0.510</u> (0.500)	0.801	0.949(0.042)	0.962(0.020)	0.572(0.094)	0.673(0.114)	0.783
Hunyuan	0.990(0.009)	0.470(0.499)	<u>0.806</u>	0.941(0.059)	0.952(0.032)	<u>0.598</u> (0.082)	0.642(0.108)	0.775
Wan2.1	<u>0.994</u> (0.006)	0.440(0.427)	0.805	<u>0.968</u> (0.045)	<u>0.964</u> (0.025)	0.601(0.086)	0.649(0.112)	<u>0.791</u>
TFM	0.995(0.004)	0.590(0.392)	0.858	0.976(0.032)	0.968(0.020)	0.592(0.087)	<u>0.662</u> (0.111)	0.795

Table 10. Ablation study comparing TFM and two variants on VideoJam-Bench, using VBench metrics.

Method	Motion			Appearance				
	Motion Smoothness	Dynamic Degree	Overall.	Subject Consistency	Background Consistency	Aesthetic Quality	Imaging Quality	Overall.
loss ^{FM}	0.983(0.005)	0.784(0.500)	0.894	0.954(0.036)	0.952(0.022)	0.508(0.073)	0.622(0.109)	0.752
FM- ow	0.975(0.005)	0.600(0.400)	0.815	0.942(0.038)	0.947(0.022)	0.490(0.101)	0.609(0.140)	0.739
loss ^{TFM} +TFM- ow (TFM)	0.991(0.009)	0.930(0.495)	0.961	0.958(0.032)	0.956(0.021)	0.590(0.074)	0.653(0.097)	0.783

Table 11. Ablation study comparing TFM and two variants on Movie-Gen, using VBench metrics, corresponding to Figure 4.

Method	Motion			Appearance				
	Motion Smoothness	Dynamic Degree	Overall.	Subject Consistency	Background Consistency	Aesthetic Quality	Imaging Quality	Overall.
loss ^{FM}	0.994(0.003)	0.460(0.367)	0.811	0.974(0.034)	0.954(0.034)	0.504(0.097)	0.633(0.135)	0.756
FM- ow	0.993(0.005)	0.461(0.498)	0.810	0.952(0.036)	0.953(0.023)	0.538(0.079)	0.617(0.102)	0.751
loss ^{TFM} +TFM- ow (TFM)	0.995(0.004)	0.590(0.392)	0.857	0.976(0.032)	0.968(0.020)	0.592(0.087)	0.662(0.111)	0.795

!#\$%&'()*+,-./01 234

!%#\$\$37+&"+0..+<+,"42):+4".04+3-"7,#")-0+,"=%(*.&"&.0\$")9)/:##),7" !"#\$\$%&'()*&"+,-,.(,)"012++3"\$&")",)#"
 *\$\$38"7,\$*3+04"4*)#.&"\$%0"3.:+"9,,'208"42.((40&5 4%,-)/+5"6+"4"7,+44+7".&"\$,)&+**"&048")9&2/:5

Figure 9. Additional qualitative comparisons between TFM and baseline methods.

!&##\$,*4&2\$*6&'")*7%5&#"\$..****8("76"&9&%/4&#***%\$8()&9"8&6%,"2&.\$&")*76-7/4

!#\$"\$2\$4"*8"6&)-%*4&)+8\$;&#"%\$)\$&6"%5&##"(-&%&#"-7***1"*768"*%&#<8*4%5

!(#8"6\$1")&*(#"74\$***(-)&\$.**/4/87")&*&#"\$7***(*,"45="#%*44&'71"1\$)6&7**76""&6".)\$-1&"5-15%74\$45&""-

!#\$%&'()*+&"#,-./)+0/11)&#""2*)"3-45"45&-"&&4

!#\$"2*)&'-7**1*%&./))+6*7%#-)"43-)-71***.)\$3-71""22\$7

!#\$8*7"-#0/8(-71"\$(&'\$7"45&"#*76+"2&%%5

Figure 10. Additional generated samples of TFM across various motion types.