Motion Attribution for Video Generation

Xindi Wu[1,2], **Despoina Paschalidou**[1], **Jun Gao**[1,4], **Antonio Torralba**[3], **Laura Leal-Taixé**[1], **Olga Russakovsky**[2], **Sanja Fidler**[1,5,6], **Jonathan Lorraine**[1]

[1]NVIDIA [2]Princeton University [3]MIT [4]University of Michigan [5]University of Toronto [6]Vector Institute

https://research.nvidia.com/labs/sil/projects/MOTIVE/

arXiv:2601.08828v2 [cs.CV] 3 Jul 2026

Abstract

Despite the rapid progress of video generation models, the role of data in influencing motion is poorly understood. We present Motive (MOTIon attribution for Video gEneration), a motion-centric, gradient-based data attribution framework that scales to modern, large, high-quality video datasets and models. We use this to study which fine-tuning clips improve or degrade temporal dynamics. Motive isolates temporal dynamics from static appearance via motion-weighted loss masks, yielding efficient and scalable motion-specific influence computation. On text-to-video models, Motive identifies clips that strongly affect motion and guides data curation that improves temporal consistency and physical plausibility. With Motive-selected high-influence data, we improve both motion smoothness and dynamic degree on VBench, achieving a 74.1% human preference win rate compared with the pretrained base model. To our knowledge, this is the first framework to attribute motion rather than visual appearance in video generative models and to use it to curate fine-tuning data.

1. Introduction

Motion is the defining element of videos. Unlike image generation, which produces a single frame, video generative models capture how objects move, interact, and obey physical constraints [Wiedemer et al., 2025, Kang et al., 2024]. Yet even with the rapid progress of video generation, a fundamental question remains:

Which training data influence the motion in generated videos?

Why it matters. Diffusion models are data-driven, and their progress has tracked the scaling of data and compute [Saharia et al., 2022, Nichol and Dhariwal, 2021, Ho et al., 2022, Peebles and Xie, 2023]. Prior work [Blattmann et al., 2023, Kaplan et al., 2020, Ravishankar et al., 2025] shows that training data shapes key generative properties, including visual quality [Rombach et al., 2022], semantic fidelity [Namekata et al., 2024], and compositionality [Wu et al., 2025, Favero et al., 2025]. Motion is no exception. *Motion* refers to temporal dynamics captured by optical flow, including trajectories, deformations, camera movement, and interactions. If generated motion reflects the data distribution that shaped the model, then attributing motion to influential training clips provides a direct lens on why a model moves the way it does and enables targeted data selection for desired dynamics. High-quality data often matters most in fine-tuning, where large pretraining corpora are inaccessible and carefully selected clips can have an outsized impact. Motion-specific attribution is therefore especially valuable in the fine-tuning regime, where the goal is to identify which clips most influence temporal coherence and physical plausibility.

Why existing methods are limited for motion. Prior diffusion data attribution focuses on images and explains static content. Extending these methods to videos naïvely collapses motion into appearance, missing the temporal structure that distinguishes videos from images. Three challenges drive this gap: (i) localizing motion so attribution focuses on dynamic regions rather than static backgrounds, (ii) scaling to sequences since gradients must integrate across time, and (iii) capturing temporal relations like velocity, acceleration, and trajectory coherence that single-frame attribution cannot measure. Addressing motion attribution requires methods that explicitly model temporal structure, rather than treating time as an additional spatial axis.

© 2026 NVIDIA. All rights reserved.

Our method. We introduce **Motive**, a motion attribution framework for video generation models that isolates motion-specific influence. **Motive** computes gradients with motion-aware masking. As a result, the attribution signal emphasizes dynamic regions rather than static appearance. Efficient approximations make the method practical for large, high-quality datasets and video generative models. Our scores trace generated motion back to training clips, enabling targeted curation and improving motion quality when used to guide fine-tuning. Our key contributions are:

1. Proposing a scalable gradient-based attribution approach for video generation models that is computationally efficient, even at the scale of modern, high-quality data and large generative models (§3.2).
2. Addressing a video-specific bias by correcting frame-length effects in gradient magnitudes, ensuring fair attribution across clips of different durations (§3.3).
3. Introducing an attribution that emphasizes temporal dynamics to trace which training clips most strongly influence motion quality (§3.4).
4. Showing that we improve motion smoothness and dynamic degree on VBench and in human evaluation (§4), matching, or surpassing, full-dataset fine-tuning performance with only 10% of the data, and outperforming motion-unaware attribution baselines (Tables 1 and 2).

2. Background

Notation and extended related work are in App. §A and §B.

2.1. Video Generation with Diffusion and Flow-Matching Models

Diffusion and flow matching in latent space. Let $p_{\boldsymbol{\theta}}(\mathbf{v} \mid \mathbf{c})$ be a conditional generator with parameters $\boldsymbol{\theta}$, where $\mathbf{v} \in \mathbb{R}^{F\times H\times W\times 3}$ is a clip of height H, width W, and F frames, and $\mathbf{c}$ denotes conditioning such as text or other multimodal metadata (e.g., fps, depth, pose). We operate in VAE latents: $\mathbf{h} = E(\mathbf{v})$ and train a denoiser or velocity field on noisy latents. A noise scheduler supplies time-dependent coefficients (α_t, σ_t) controlling signal and noise scales, and the forward noising is:

$$\mathbf{z}(t, \boldsymbol{\epsilon}) \;=\; \alpha_t\, \mathbf{h} \;+\; \sigma_t\, \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \quad t \in \{1, \dots, T\}. \tag{1}$$

Denoising diffusion [Ho et al., 2020] trains a network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}, \mathbf{c}, t)$ to predict the injected noise:

$$\mathcal{L}_{\text{diff}}(\boldsymbol{\theta}; \mathbf{v}, \mathbf{c}) \;=\; \mathbb{E}_{t,\boldsymbol{\epsilon}}\Big[\big\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}(t, \boldsymbol{\epsilon}), \mathbf{c}, t) - \boldsymbol{\epsilon}\big\|_2^2\Big]. \tag{2}$$

Flow matching [Lipman et al., 2022, Albergo et al., 2023] learns a time-dependent vector field $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}_t, \mathbf{c}, t)$ that matches the instantaneous velocity $\dot{\mathbf{z}} = \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{z}$ induced by a chosen interpolant:

$$\mathcal{L}_{\text{flow}}(\boldsymbol{\theta}; \mathbf{v}, \mathbf{c}) \;=\; \mathbb{E}_{t,\boldsymbol{\epsilon}}\Big[\big\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}(t, \boldsymbol{\epsilon}), \mathbf{c}, t) - \dot{\mathbf{z}}(t, \boldsymbol{\epsilon})\big\|_2^2\Big]. \tag{3}$$

Both objectives train time-indexed predictors over the latent space by integrating over t and $\boldsymbol{\epsilon}$, thus gradient-based methods like attribution share similar challenges.

From images to video for generation. Adding a temporal axis materially changes modeling and training. Generation must capture spatial appearance and temporal dynamics such as object and camera motion, deformations, and interactions. Modern systems extend image backbones with temporal capacity, for example, 3D U-Nets or 2D U-Nets augmented with temporal attention, causal or sliding-window context, and factorized space-time blocks, often trained in a latent-video VAE that compresses frames while preserving temporal cues. Training departs from images along several axes, which we address in §3: (i) *Compute and storage.* Longer sequences multiply the cost of sampling timesteps, noise draws, and frames, motivating fixed-timestep or small-subset estimators that reduce variance without prohibitive cost (§3.2). (ii) *Variable horizon.* Clips vary in F and frame rate (§3.3). (iii) *Time-specific failure modes.* Typical artifacts include inconsistent trajectories, temporal flicker, identity drift, and physically implausible dynamics despite sharp individual frames (§3.4).

Motion representations in videos. We denote our video as $\mathbf{v} = [\mathbf{f}_f]_{f=1}^{F}$ with $\mathbf{f}_f \in \mathbb{R}^{H\times W\times 3}$ being the f-th frame. We represent motion via optical flow between consecutive frames: $\mathbf{F}_f : \{1,\dots,H\} \times \{1,\dots,W\} \to \mathbb{R}^2$, where each flow vector in $\mathbb{R}^2$ encodes the horizontal displacement $\mathrm{d}w$ and vertical displacement $\mathrm{d}h$ of a pixel. The motion magnitude is $M_f(h,w) = \|\mathbf{F}_f(h,w)\|_2$. The M_f over frames f and pixels h, w summarizes the amount and spatial layout of motion in a clip, which we use to provide masks in our motion-weighted loss in §3.

2.2. Data Attribution

Data attribution measures how individual training samples affect a model's predictions [Bae et al., 2024]. A classic approach to data attribution is to use influence functions [Koh and Liang, 2017]. Intuitively, the influence of a training sample measures: if we upweight this training example, how much would the model's prediction on a test datum change? Consider a loss function $\mathcal{L}(\boldsymbol{\theta};\mathbf{x})$ and a test sample $\mathbf{x}_{\text{test}}$, the influence of a training point $\mathbf{x}_n$ can be quantified as:

$$I(\mathbf{x}_n,\mathbf{x}_{\text{test}}) \;=\; -\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta};\mathbf{x}_{\text{test}})^{\top}\,\mathbf{H}_{\boldsymbol{\theta}}^{-1}\,\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta};\mathbf{x}_n), \quad \mathbf{H}_{\boldsymbol{\theta}} = \tfrac{1}{N}\sum\nolimits_{n=1}^{N}\nabla_{\boldsymbol{\theta}}^2\mathcal{L}(\boldsymbol{\theta};\mathbf{x}_n), \tag{4}$$

where the inverse Hessian captures the curvature of the loss landscape, yet computing or storing it is infeasible at modern model and dataset scales. Thus, practical methods (e.g., TracIn [Pruthi et al., 2020] and TRAK [Park et al., 2023]) approximate influence via gradient inner products or gradient feature projections.

Attribution in diffusion models. Diffusion training aggregates gradients over timesteps t and noise draws $\boldsymbol{\epsilon}$, where gradient norms vary systematically with t, producing a timestep bias where examples aligned with large-norm timesteps appear spuriously influential. Diffusion-ReTrac [Xie et al., 2024] reduces this bias by normalizing gradients and sub-sampling t and $\boldsymbol{\epsilon}$ for influence. Let $\mathcal{L}_{\text{diff}}$ denote the diffusion loss, and with the sampled-timestep-and-noise set $\mathcal{T}$, we compute a cosine-style score for normalized test and train gradients:

$$I_{\text{diff}}(\mathbf{x}_n,\mathbf{x}_{\text{test}}) = \underbrace{\frac{1}{|\mathcal{T}_{\text{test}}|}\sum_{t,\boldsymbol{\epsilon}\in\mathcal{T}_{\text{test}}}\frac{\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\mathbf{x}_{\text{test}},t,\boldsymbol{\epsilon})}{\left\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\mathbf{x}_{\text{test}},t,\boldsymbol{\epsilon})\right\|}^{\top}}_{\text{normalized test gradients}} \underbrace{\frac{1}{|\mathcal{T}_n|}\sum_{t,\boldsymbol{\epsilon}\in\mathcal{T}_n}\frac{\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\mathbf{x}_n,t,\boldsymbol{\epsilon})}{\left\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\mathbf{x}_n,t,\boldsymbol{\epsilon})\right\|}}_{\text{normalized training gradients}}. \tag{5}$$

Averaging gradients over $(t,\boldsymbol{\epsilon})$ stabilizes estimates, and normalization mitigates timestep-induced scale effects. Attribution quality is also sensitive to the measurement function used to score examples, such as denoising loss versus likelihood proxies [Zheng et al., 2023].

Why vanilla attribution is insufficient for videos. Naïvely applying gradient-based attribution to video diffusion risks treating appearance and motion alike, overemphasizing appearance (objects, textures, backgrounds) while overlooking dynamics [Park et al., 2025, Tulyakov et al., 2018]. Its cost grows with clip length, sampled timesteps, noise draws, and gradient dimensionality, making naïve methods impractical at modern video scales. To improve motion, we need attribution that suppresses static appearance, emphasizes motion-specific signals, and remains efficient (§3). Motion is distributed across frames and entangled with static cues, thus influence cannot be assigned frame-independently.

3. Method

We formalize the problem in §3.1 and develop a practical framework for motion attribution in video diffusion models with four components: scalable gradient computation (§3.2), frame-length bias fix (§3.3), motion-aware weighting (§3.4), and target data selection (§3.5). We provide efficiency analysis (§3.6) demonstrating scalability to billion-parameter models and large-scale video datasets.

3.1. Problem Formulation

We study data attribution for motion in the fine-tuning setting. Let $\mathcal{D}_{\text{ft}} = \{(\mathbf{v}_n,\mathbf{c}_n)\}_{n=1}^{N}$ be the fine-tuning corpus. Given a query video $(\hat{\mathbf{v}},\hat{\mathbf{c}})$, we assign to each training clip $(\mathbf{v}_n,\mathbf{c}_n)$ a motion-aware influence score $I(\mathbf{v}_n,\hat{\mathbf{v}};\boldsymbol{\theta})$ that explains how it contributes to the dynamics observed in $\hat{\mathbf{v}}$. The score should satisfy: (i)

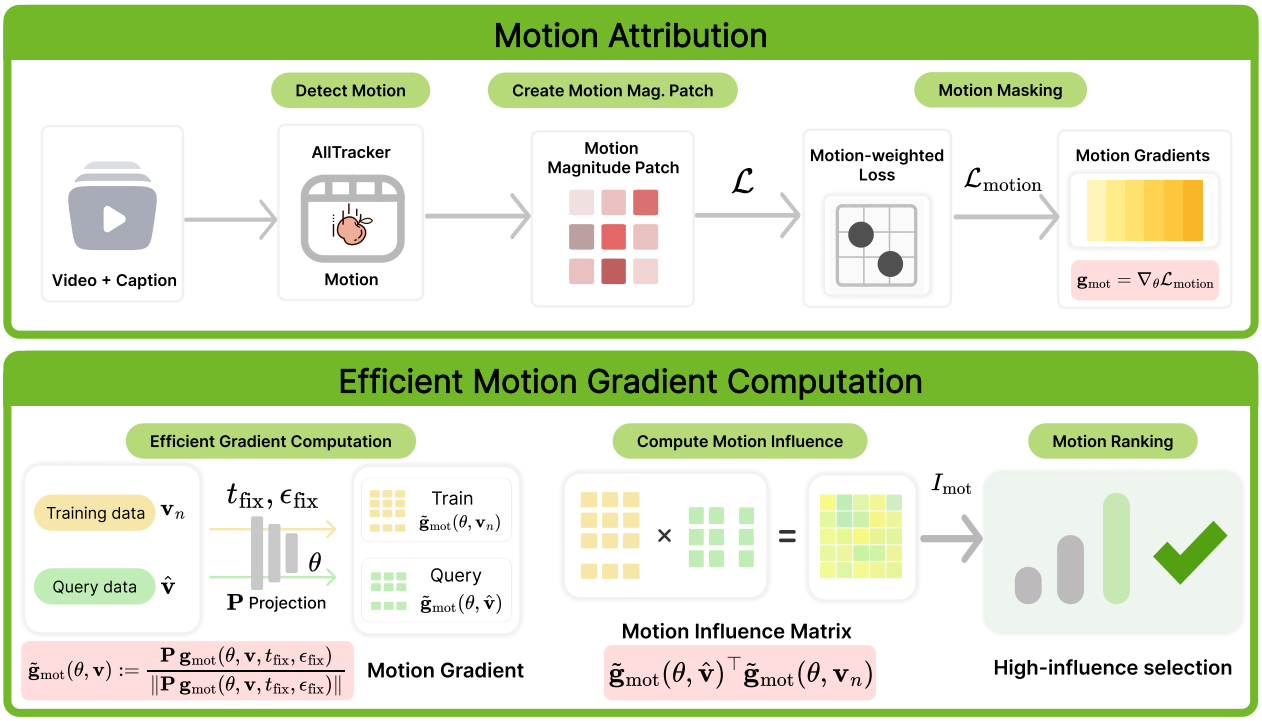


Figure 1: **Motive**. *Top.* Motion-gradient computation (§3.4) has three steps: (1) detect motion with AllTracker, (2) compute motion-magnitude patches, (3) apply loss-space motion masks to focus gradients on dynamic regions. *Bottom.* Our method (§3.2) is made scalable via a single-sample variant with common randomness and a projection, computed for each pair of training and query data, aggregated (§3.5) for a final ranking, and eventually used to select fine-tuning subsets.

predictivity, rankings correlate with observed changes from fine-tuning on the most influential subsets; (ii) *efficiency*, scales to modern video generators, such as forgoing explicit Hessian inversion, expensive per-data integration, or prohibitive storage. To do this, we augment the influence target defined in Eq. 5 to be (a) lower variance for stable rankings with feasible levels of compute, (b) more scalable to store, and (c) motion-centric.

Fine-tuning Subset Selection. For a budget $K \ll N$, we get a motion-influential subset by ranking scores and taking the top-K examples. For multiple query motions, we combine selections as described in §3.5. The resulting subsets serve as candidates for motion-centric fine-tuning.

3.2. Scalable Gradient-based Attribution

We make attribution scalable for modern, large video datasets and models via inverse-Hessian approximations, lower-variance gradient-similarity estimators, low-cost single-sample estimators, and a Fastfood projection for tractable storage.

Approximating the inverse-Hessian. Computing exact inverse-Hessian-vector products is infeasible for modern neural networks. We estimate influence via gradient similarity, using an identity preconditioner for the inverse Hessian [Koh and Liang, 2017, Pruthi et al., 2020, Park et al., 2023].

Common randomness for stable rankings. To reduce variance without changing the target, we evaluate train and test gradients under the same (t, ϵ) pairs and average over a small set $\mathcal{T}$ [Xie et al., 2024, Lin et al., 2024]. This paired averaging stabilizes rankings compared to independent draws:

$$I^1_{\text{diff}}(\mathbf{x}_n, \mathbf{x}_{\text{test}}) = \frac{1}{|\mathcal{T}|} \sum_{t,\epsilon \in \mathcal{T}} \frac{\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\, \mathbf{x}_{\text{test}}, t, \boldsymbol{\epsilon})}{\left\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\, \mathbf{x}_{\text{test}}, t, \boldsymbol{\epsilon})\right\|}^\top \frac{\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\, \mathbf{x}_n, t, \boldsymbol{\epsilon})}{\left\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\, \mathbf{x}_n, t, \boldsymbol{\epsilon})\right\|}. \tag{6}$$

normalized test gradients — normalized training gradients

Single-sample variant for reduced compute. We fix a single t_{fix} and a shared draw $\boldsymbol{\epsilon}_{\text{fix}} \sim \mathcal{N}(0, \mathbf{I})$ for all train–test pairs. Sharing $(t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})$ allows low enough variance for the efficient single-sample estimator to maintain relative ordering [Xie et al., 2024, Lin et al., 2024]. The estimator becomes:

$$I^2_{\text{diff}}(\mathbf{x}_n, \mathbf{x}_{\text{test}}) = \frac{\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\, \mathbf{x}_{\text{test}}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})}{\left\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\, \mathbf{x}_{\text{test}}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})\right\|}^{\top} \frac{\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\, \mathbf{x}_n, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})}{\left\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\, \mathbf{x}_n, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})\right\|}. \tag{7}$$

normalized test gradient — normalized training gradient

Structured projection for reduced storage. To operate at model scale, we apply a Johnson–Lindenstrauss projection via Fastfood [Le et al., 2014] and then normalize. Let

$$\mathbf{P} \in \mathbb{R}^{D' \times D} \;\text{ implemented as }\; \mathbf{P} := \tfrac{1}{\xi\sqrt{D'}}\, \mathbf{SQG\Pi QB}, \tag{8}$$

where $\mathbf{Q}$ is the Walsh–Hadamard matrix, $\mathbf{B}$ is a diagonal Rademacher matrix, $\mathbf{\Pi}$ is a random permutation, $\mathbf{G}$ is a diagonal Gaussian scaling, and $\mathbf{S}$ is a diagonal rescaling, and ξ normalizes the variance. The projected, normalized gradient is:

$$\tilde{\mathbf{g}}(\boldsymbol{\theta}, \mathbf{x}) := \frac{\mathbf{P}\, \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{diff}}(\boldsymbol{\theta}, \mathbf{x}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})}{\|\mathbf{P}\, \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{diff}}(\boldsymbol{\theta}, \mathbf{x}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})\|}. \tag{9}$$

Then the influence score is the dot product of normalized projected gradients (i.e., cosine similarity) in $\mathbb{R}^{D'}$:

$$I^3_{\text{diff}}(\mathbf{x}_n, \mathbf{x}_{\text{test}}) = \tilde{\mathbf{g}}(\boldsymbol{\theta};\, \mathbf{x}_{\text{test}})^{\top}\, \tilde{\mathbf{g}}(\boldsymbol{\theta};\, \mathbf{x}_n). \tag{10}$$

projected, normalized test gradient — projected, normalized training gradient

This keeps compute $\mathcal{O}(D' \log D')$ for projection and $\mathcal{O}(D')$ per dot product, with storage $\mathcal{O}(|\mathcal{D}|\, D')$, while staying close to the ranking behavior of full-gradient cosine similarity [Park et al., 2023].

3.3. Video-specific Frame-length Bias Fix

Raw gradient magnitudes depend on the number of frames F in the video $\mathbf{v}$, thereby biasing scores toward longer videos. We correct this by normalizing for frame count before the projection–normalization step:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\, \mathbf{v}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}}) \leftarrow \frac{1}{F}\, \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\, \mathbf{v}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}}). \tag{11}$$

We still apply ℓ_2 normalization in Eq. 10, further stabilizing scales across examples. Together, single-timestep, common randomness, projection, and frame-length correction form a compact, scalable estimator that we use throughout. However, naïve video-level attribution conflates appearance with motion, often ranking clips high due to shared backgrounds or objects, while offering little insight into dynamics.

3.4. Motion Attribution

To move beyond whole-video influence, we introduce motion attribution, which isolates the contribution of training data to temporal dynamics. Unlike video-level attribution, which treats each clip as a single unit and conflates appearance with motion, motion attribution reweights per-location gradients using motion masks, assigning influence via dynamic behavior rather than static content.

Motion Masking Attribution. Motion is what distinguishes video diffusion from image diffusion. Our goal is to understand how training data shapes motion in video diffusion models. Prior work has emphasized architectural or algorithmic changes for motion modeling [Peebles and Xie, 2023, Blattmann et al., 2023, Guo et al., 2023], many of the largest generative gains have instead come from scaling and curating massive video corpora, which in turn enable impressive motion synthesis results in video diffusion models [Ho et al., 2022, Wan et al., 2025, Tan et al., 2024, Yang et al., 2024]. Yet we lack tools that quantify how specific training clips shape particular motion patterns. We address this by attributing motion back to data via motion-weighted gradients, which yields actionable signals for targeted data selection, artifact diagnosis, and selective fine-tuning.

Motion Detection and Latent Space Mapping. Given a video $\mathbf{v} \in \mathbb{R}^{F \times H \times W \times 3}$ with F frames of resolution $H \times W$, we first encode it into the VAE latent space as $\mathbf{h} = E(\mathbf{v}) \in \mathbb{R}^{F \times H/s \times W/s \times C}$, with downsampling factor $s = 8$ and $C = 16$ following the `Wan2.1` backbone used in our experiments.

For motion computation, we use AllTracker [Harley et al., 2025] to extract motion information in pixel space: $A = \mathcal{A}(\mathbf{v}) \in \mathbb{R}^{F\times H\times W\times 4}$, where the first two channels contain optical flow maps $A_{:,:,:,0:2}$ indicating pixel displacement between frames, and the remaining channels $A_{:,:,:,2:4}$ encode visibility and confidence scores. We extract displacement vectors at each pixel location as:

$$\mathbf{D}_f(h,w) = (A_{f,h,w,0}, A_{f,h,w,1}) = (\mathrm{d}w, \mathrm{d}h). \tag{12}$$

We then bilinearly downsample motion quantities from (H, W) to the latent grid $\left(\frac{H}{s}, \frac{W}{s}\right)$ so that our masking lives where gradients are computed.

Motion-Weighted Gradient Computation. We define the motion magnitude at each location as: $M_f(h,w) = \|\mathbf{D}_f(h,w)\|_2$. To obtain comparable motion weights across frames and pixels, we min–max normalize over all frames and pixels, ensuring values lie in $[0,1]$:

$$\mathbf{W}(f,h,w) = \frac{M_f(h,w) - m_{\min}}{m_{\max} - m_{\min} + \zeta}, \tag{13}$$

where $m_{\min} = \min_{f',h',w'} M_f(h',w')$, $m_{\max} = \max_{f',h',w'} M_f(h',w')$, and $\zeta = 10^{-6}$ ensures a positive denominator. This normalization mitigates bias from absolute motion scale, yielding weights that emphasize relative motion saliency rather than raw magnitude, following prior practice in video saliency detection [Fang et al., 2013]. Let $(\tilde{h}, \tilde{w})$ index the latent grid. We obtain latent-aligned weights by bilinear downsampling:

$$\tilde{\mathbf{W}}(f,\tilde{h},\tilde{w}) \;=\; \mathrm{Bilinear}\big(\mathbf{W}(\cdot,\cdot,\cdot), F,\; \tfrac{H}{s},\; \tfrac{W}{s}\big)\,. \tag{14}$$

We compute per-location squared error at fixed $(t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})$ at each frame f and "latent pixel" $(\tilde{h}, \tilde{w})$:

$$\tilde{\mathcal{L}}_{\boldsymbol{\theta},\mathbf{v},\mathbf{c}}(f,\tilde{h},\tilde{w}) \;=\; \Big([\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}(\mathbf{v}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}}), t_{\text{fix}}, \mathbf{c})]_{f,\tilde{h},\tilde{w}} - [\boldsymbol{\epsilon}_{\text{target}}(t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})]_{f,\tilde{h},\tilde{w}} \Big)^2, \tag{15}$$

and define the motion-weighted loss by averaging over frames and latent spatial locations:

$$\mathcal{L}_{\text{mot}}(\boldsymbol{\theta}; \mathbf{v}, \mathbf{c}) \;=\; \frac{1}{F_{\mathbf{v}}} \operatorname*{mean}_{f,\tilde{h},\tilde{w}} \Big[\tilde{\mathbf{W}}_{\mathbf{v},\mathbf{c}}(f,\tilde{h},\tilde{w}) \cdot \tilde{\mathcal{L}}_{\boldsymbol{\theta},\mathbf{v},\mathbf{c}}(f,\tilde{h},\tilde{w}) \Big]. \tag{16}$$

Notably, when $\tilde{\mathbf{W}}$ is all ones, this recovers the standard objective with no motion emphasis. The $1/F_{\mathbf{v}}$ factor corrects for frame-length bias and $F_{\mathbf{v}}$ signifies how the number of frames may be video-dependent. The corresponding motion-weighted gradient for attribution is:

$$I_{\text{mot}}(\mathbf{v}_n, \hat{\mathbf{v}}) = \tilde{\mathbf{g}}_{\text{mot}}(\boldsymbol{\theta}, \hat{\mathbf{v}})^\top \tilde{\mathbf{g}}_{\text{mot}}(\boldsymbol{\theta}, \mathbf{v}_n), \quad \text{where } \mathbf{g}_{\text{mot}} := \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{mot}} \text{ and } \tilde{\mathbf{g}}_{\text{mot}}(\boldsymbol{\theta}, \mathbf{v}) := \frac{\mathbf{P}\mathbf{g}_{\text{mot}}(\boldsymbol{\theta}, \mathbf{v}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})}{\|\mathbf{P}\mathbf{g}_{\text{mot}}(\boldsymbol{\theta}, \mathbf{v}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})\|}. \tag{17}$$

Loss-space masking leaves forward noising and generation unchanged and reweights only attribution, avoiding interactions between motion weighting and noise injection. In contrast, our motion-aware attribution emphasizes dynamic regions and de-emphasizes static backgrounds, so rankings identify training clips that most strongly shape the model's motion rather than appearance.

3.5. Most Influential Fine-tuning Subset Selection

Goal. Given a query clip $(\hat{\mathbf{v}}, \hat{\mathbf{c}})$, we compute a motion-aware attribution value for each fine-tuning sample $(\mathbf{v}_n, \mathbf{c}_n) \in \mathcal{D}_{\text{ft}}$ using $I_{\text{mot}}(\mathbf{v}_n, \hat{\mathbf{v}})$ (Eq. 17). We construct a fine-tuning set $\mathcal{S}$ for one or many query videos $\hat{\mathbf{v}}$.

Single-query-point fine-tuning selection. For a budget of K data points, we select the K highest-scoring examples. In practice, K is chosen as a percentile of the dataset size (e.g., top 1–10%), ensuring the subset scales consistently across datasets.

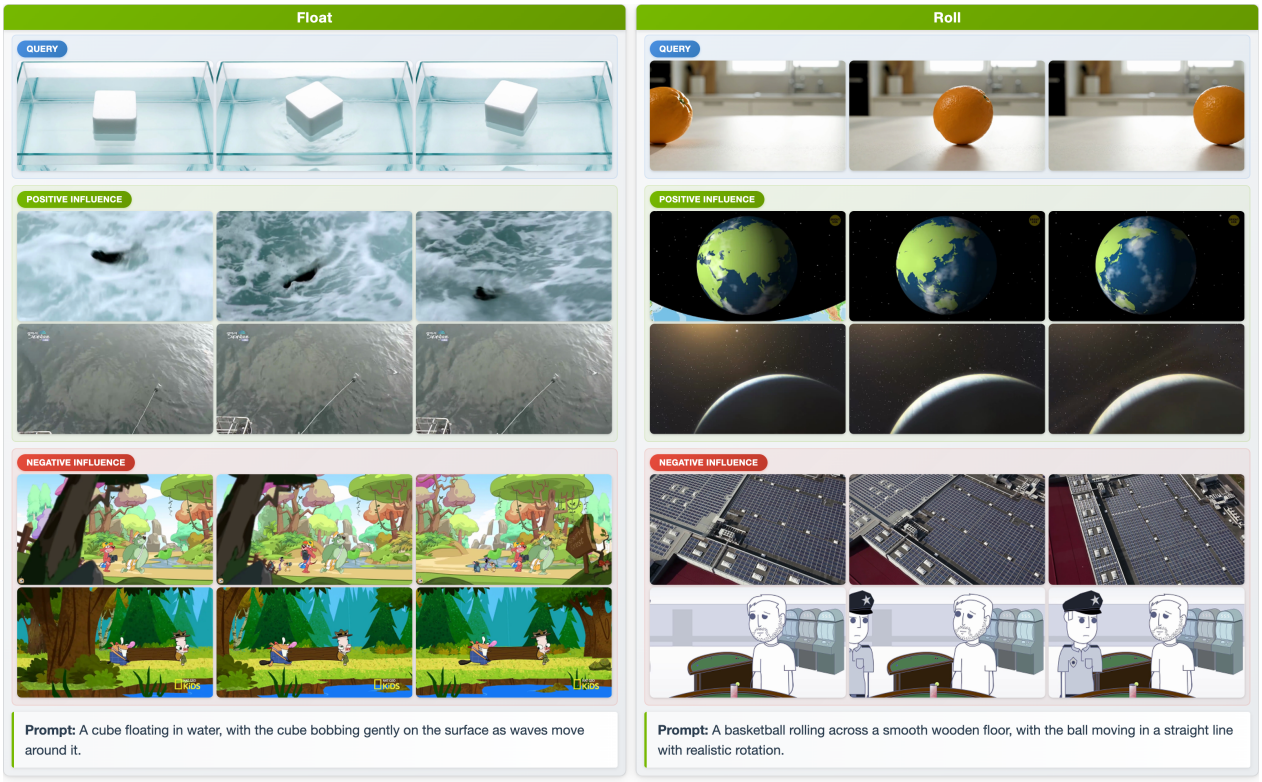


Figure 2: **Motion Attribution Samples** with Wan2.1-T2V-1.3B. *Top*: Query clips showing float (*left*) and roll (*right*) motions. *Middle*: Top-ranked positive training samples identified by **Motive** with high influence scores. *Bottom*: Negative influence samples with minimal, camera-only motion, or cartoon-style content that conflict with target motions. Videos are included in the supplementary material.

Multi-query-point fine-tuning selection: aggregating attribution scores. For Q queries, we adopt the majority voting approach from ICONS [Wu et al., 2024] and aggregate motion-aware influence scores across queries by percentile thresholding and voting. A sample receives a vote if the score is above the percentile cutoff τ for that query. The consensus score of a candidate $\mathbf{v}_n$ is the total number of queries that vote for it. We then rank all training samples by $\mathrm{MajVote}(\mathbf{v}_n)$ and select the top-K to form the fine-tuning subset. This formulation emphasizes samples that are consistently influential across multiple queries, without requiring cross-query calibration of raw scores:

$$\mathrm{MajVote}_n = \sum_{q=1}^{Q} \mathbb{I}\big[\, I_{\mathrm{mot}}(\mathbf{v}_n, \hat{\mathbf{v}}_q) > \tau \,\big], \quad \mathcal{S}_{\mathrm{vote}}(K) = \big\{ \mathbf{v}_n \,|\, \mathbf{v}_n \text{ in top-}K \text{ by } \mathrm{MajVote} \big\}. \tag{18}$$

3.6. Computational Efficiency Analysis

Gradient Compute. Naïvely averaging over timesteps and noise for every example costs $\mathcal{O}(|\mathcal{D}|\,|\mathcal{T}|\,B)$, where B is a single forward+backward cost and $|\mathcal{T}|$ is the number of sampled $t, \boldsymbol{\epsilon}$ per data. Using a single sample reduces this to $\mathcal{O}(|\mathcal{D}|\,B)$, which is key to keeping the cost reasonable for modern video datasets and models, while reusing a sample across data allows low enough variance for stable rankings. Projection adds $\mathcal{O}(D' \log D')$ per example using Fastfood [Le et al., 2014], negligible relative to a backward pass.

Gradient Storage. Storing full gradients is $\mathcal{O}(|\mathcal{D}|\,D)$. We instead store only projected vectors, $\mathcal{O}(|\mathcal{D}|\,D')$, plus the structured Fastfood state, $\mathcal{O}(D)$. Since D' is typically orders of magnitude smaller than D, this transformation makes storage tractable for billion-parameter models.

Data Ranking Compute. Influence computation in Eq. 10 is an inner product in $\mathbb{R}^{D'}$, so evaluating all train examples against a query is $\mathcal{O}(|\mathcal{D}|\,D')$, and sorting is $\mathcal{O}(|\mathcal{D}| \log |\mathcal{D}|)$.

Table 1: **VBench Evaluation.** Performance comparison on VBench [Huang et al., 2024] across different baselines (all values in %, higher is better). All selection methods use 10% of training data; our method uses majority vote aggregation (§3.5) across motion queries. MM: motion masking. Subj.: Subject Consistency, Bg.: Background Consistency, Mot.: Motion Smoothness, Dyn.: Dynamic Degree, Aesth.: Aesthetic Quality, Img.: Imaging Quality.

	Wan2.1-T2V-1.3B						Wan2.2-TI2V-5B					
Method	Subj.	Bg.	Mot.	Dyn.	Aesth.	Img.	Subj.	Bg.	Mot.	Dyn.	Aesth.	Img.
Base	95.3	96.4	96.3	39.6	45.3	**65.7**	94.9	96.4	97.5	42.0	44.4	65.5
Full FT	95.9	**96.6**	96.3	42.0	45.0	63.9	**95.3**	96.5	97.5	45.3	44.8	**66.2**
Random	95.3	96.6	96.3	41.3	45.7	65.1	94.7	96.2	97.3	41.6	44.6	65.2
Motion mag.	95.6	96.2	95.7	40.1	45.1	63.2	95.0	96.3	97.4	44.9	45.0	65.1
V-JEPA	95.7	96.0	95.6	41.6	44.9	62.7	95.2	96.4	97.3	45.6	44.9	64.8
Ours w/o MM	95.4	96.1	96.3	43.8	45.7	63.2	94.9	96.5	97.4	43.8	45.2	64.8
Ours (Motive)	**96.3**	96.1	96.3	**47.6**	**46.0**	64.6	95.1	**96.6**	**97.6**	**48.3**	**45.6**	65.5

Additional Motion-Emphasis Compute. Motion-specific overhead primarily stems from AllTracker mask extraction with complexity $\mathcal{O}(|\mathcal{D}| \cdot H \cdot W \cdot F)$ for clip length F and frame resolution $H \times W$. Masks are extracted once, cached, and negligible relative to gradient cost. We provide a detailed runtime breakdown in App. G.1.

4. Experiment

4.1. Setup

Fine-tuning Datasets. We evaluate our motion attribution framework on two large-scale video datasets: `VIDGEN-1M` [Tan et al., 2024] and `4DNeX-10M` [Chen et al., 2025], both of which offer diverse motion patterns, rich temporal dynamics, and complex scenes. For our experiments, we use 10k videos from both datasets, which provide sufficient scale and diversity to thoroughly evaluate motion attribution methods across different temporal patterns and video generation scenarios.

Motion Query Data. To evaluate our motion attribution, we curate a set of query videos representing distinct motion patterns and scenarios. Our query dataset consists of videos spanning 10 motion categories, with a focus on object dynamics: *compress, bounce, roll, explode, float, free fall, slide, spin, stretch, swing*. Five videos, totaling 50 queries, represent each motion type. These videos are chosen for their clear, isolated motions, serving as a basis for evaluating attribution quality and downstream motion generation. Details are in App. F.2.

Model & Baselines. We use pretrained models `Wan2.1-T2V-1.3B` and `Wan2.2-TI2V-5B`, with additional results on `LTX-2B` in App. C. Our baselines: Base model: pretrained, no fine-tuning; Full fine-tuning: approximate upper bound using the complete dataset; Random selection: uniform sampling; Motion magnitude: selects videos with the highest average motion magnitude; V-JEPA embeddings: selects most representative videos of motion patterns using V-JEPA [Assran et al., 2025] features, capturing high-level motion semantics; and Ours w/o motion masking: influence of the entire video level without motion-specific masking.

Benchmark. We evaluate our attribution using VBench [Huang et al., 2024] metrics across six dimensions: subject consistency, background consistency, motion smoothness, dynamic degree, aesthetic quality, and imaging quality. Motion smoothness and dynamic degree are our primary targets for temporal dynamics, while other metrics help maintain visual quality. We use custom evaluation prompts, following VBench's descriptive style, for 10 motion types, with 5 prompts each, to assess our framework's effectiveness on target motions.

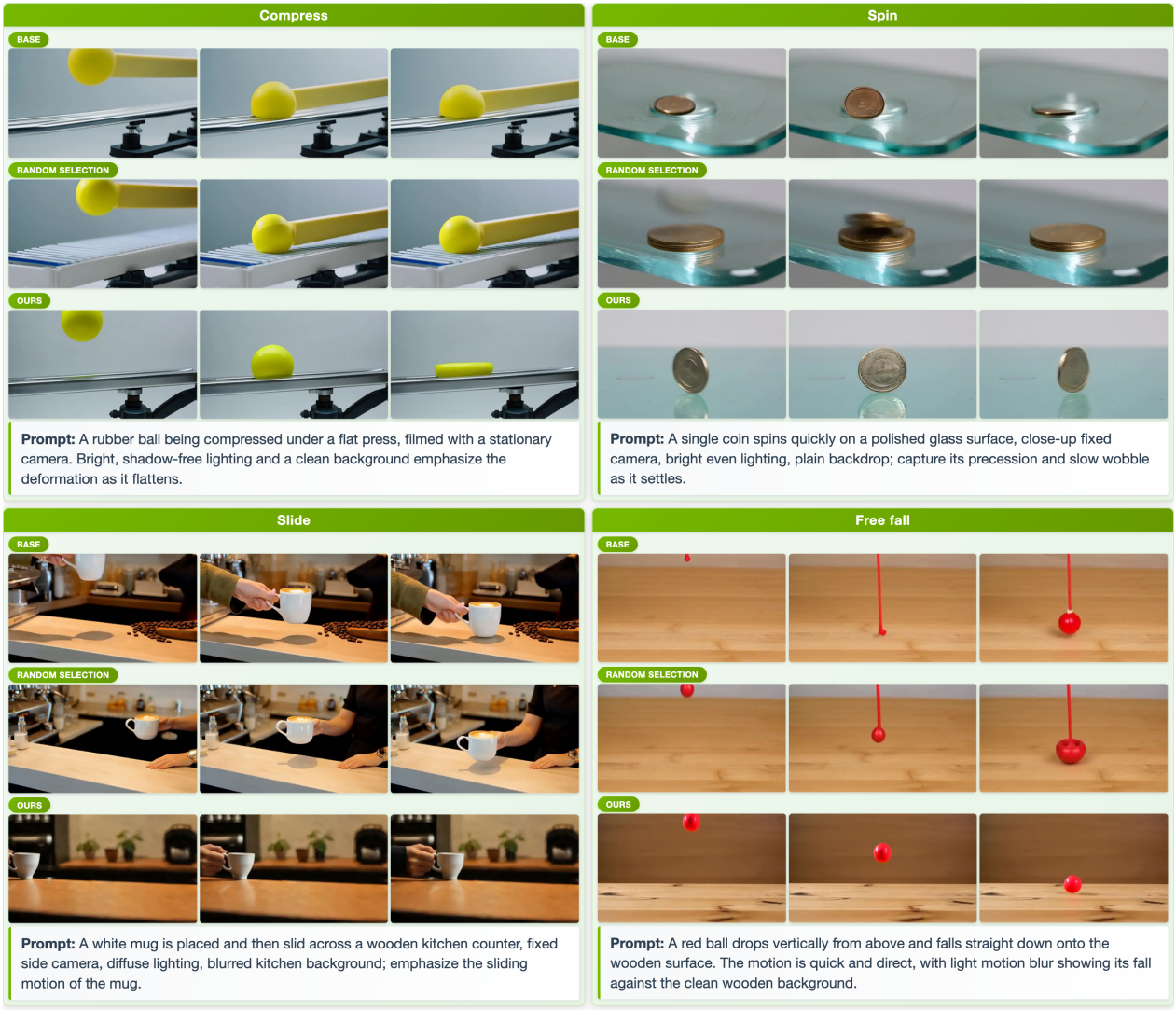


Figure 3: **Qualitative Comparisons.** We compare four motion scenarios across the base model (Wan2.1-T2V-1.3B), or fine-tuned with random selection and **Motive**. Our approach produces more realistic motion. Videos are included in the supplementary material.

Implementation Details. We finetune base models with our **Motive**-selected high-quality video data following the official and DiffSynth-Studio implementation. During fine-tuning, we update only the DiT backbone while freezing the T5 text encoder and VAE. All models are trained at a resolution of 480×832 with a learning rate of 1×10^{-5}. Specialist models are trained on single motion category selected data, while generalist models use aggregated selections (both with top 10% selection from `VIDGEN-1M` [Tan et al., 2024] or `4DNeX` [Chen et al., 2025]). Compute and runtime details are provided in App. G.1.

4.2. Main Results

High-influence selection and negative filtering. Fig. 2 shows our motion-aware attribution ranks clips with clear, physically grounded dynamics and downranks those with little transferable motion. For rolling and floating, positives show continuous trajectories and smooth temporal evolution (turbulent water, planetary rotation). Negatives are mostly static footage, camera-only motion, or cartoons whose simplified kinematics do not transfer. This pattern holds across motion categories and aligns with the quantitative gains below.

Qualitative improvements across motion types. Fig. 3 compares the base `Wan2.1-T2V-1.3B` model, finetuned with random selection or **Motive** with equal data budgets, across 4 scenarios. Our method yields higher motion fidelity and temporal consistency than baselines, especially for complex deformation and physics-driven motion.

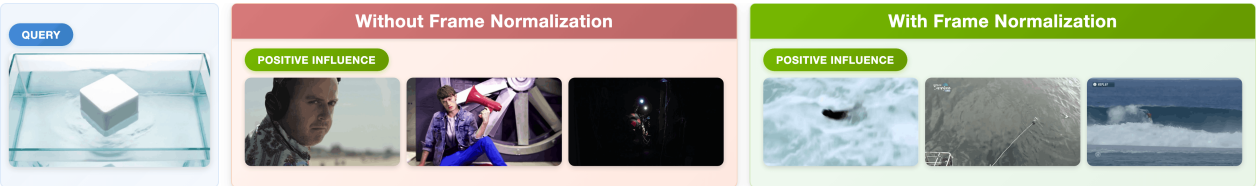


Figure 4: **Impact of Frame-Length Normalization on Motion Attribution.** Comparison of top-ranked samples for floating motion query. **Left**: With proper frame-length normalization, top samples consistently exhibit floating motion (waves, floating objects, surfing). **Right**: Without normalization, rankings are biased by video length, resulting in no coherent patterns among top samples.

Quantitative Results. We evaluate our approach using VBench [Huang et al., 2024], demonstrating consistent improvements in motion when fine-tuning with attribution-selected data. As shown in Tab. 1, **Motive** consistently achieves the highest dynamic degree scores across both models: 47.6% on `Wan2.1-T2V-1.3B` and 48.3% on `Wan2.2-TI2V-5B`, significantly outperforming random selection (41.3% and 41.6%) and whole video attribution (43.8% for both models). Our method also excels in aesthetic quality (46.0% and 45.6%), while maintaining competitive motion smoothness (96.3% and 97.6%). Notably, using only 10% of training data, our approach surpasses the full fine-tuning on dynamic degree (42.0% and 45.3%) on both models, demonstrating superior performance of motion-specific attribution for targeted fine-tuning. Additional results on `LTX-2B` are in App. C; motion magnitude distribution analysis is in App. D.

4.3. Human Evaluation

Automated scores can miss perceptual motion quality, so we run a human evaluation pairwise comparison protocol: participants view two generated videos and choose which shows better motion. We recruit 17 annotators and evaluate 10 motion categories. For each category, we prepare 5 test cases and compare our method to baselines across three pairings, yielding a balanced set of judgments. Presentation order is randomized, and ties are allowed. We report win rate (fraction our method is preferred), tie rate, and overall preference. As shown in Tab. 2, annotators favor our attribution-guided selection: 74.1% win rate vs. the base model and 53.1% vs. the full fine-tuned model, showing perceptually meaningful motion improvements.

Method	Win (%)	Tie (%)	Loss (%)
Ours vs. Base	74.1	12.3	13.6
Ours vs. Random	58.9	12.1	29.0
Ours vs. Full FT	53.1	14.8	32.1
Ours vs. w/o MM	46.9	20.0	33.1

Table 2: **Human evaluation.** Pairwise comparisons across 50 videos with 17 participants (850 total). Win, tie, and loss rates show where our method is preferred, rated equal, or outperformed.

4.4. Ablations

Projected Gradients Preserve Influence Rankings. Comparing full gradients for attribution is infeasible at a billion-parameter scale. We reduce dimensionality with structured random projections that preserve influence geometry, ablating $D' \in \{128, \ldots, 2048\}$ against the full-gradient baseline. We assess ranking preservation via Spearman correlation with unprojected scores (Fig. 5). Small projections preserve rankings poorly: $D' = 128$ yields $\rho = 46.9\%$. Preservation improves with size: $D' = 512$ reaches $\rho = 74.7\%$. Beyond that, gains are marginal while cost rises: $D' = 1024$ ($\rho = 75.7\%$) and $D' = 2048$ ($\rho = 76.1\%$). Thus, $D' = 512$ offers the best trade-off, scaling to large models while maintaining quality.

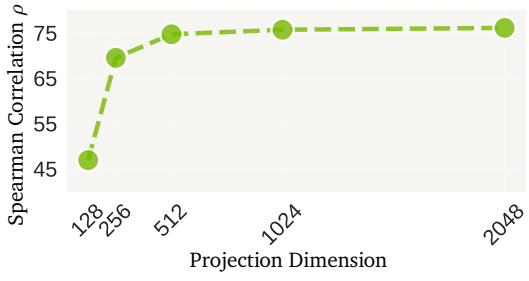


Figure 5: **Projection dimension analysis.** Spearman correlation between projected and full gradients shows rapid improvement with projection dimension, with 512 providing a strong trade-off between accuracy and efficiency.

Single-timestep attribution. Using a single timestep avoids the cost of averaging across timesteps while closely matching the multi-timestep baseline. With fixed $t = 751$ (midpoint of the 1000-step trajectory), we obtain $\rho = 66\%$ agreement with ground truth computed using 10 evenly-spaced timesteps based on flow matching schedule. High timesteps (early denoising) heavily obscure motion with noise; low timesteps (late denoising) operate on nearly formed videos, where gradients reflect fine details. $t = 751$ (mid-denoising) balances influence ranking correlation and compute efficiency. Averaging multiple timesteps yields minimal gains, a single fixed timestep is therefore sufficient for variance-reduced, scalable attribution.

Frame-Length Normalization. As in the Wan training protocol, we standardize all videos to 81 frames at 16 fps (satisfying the $4n+1$ constraint) for fair attribution across clips of different raw lengths. Without standardization, gradient-based scores correlate strongly with video length rather than motion quality ($\rho = 78.0\%$), leading to longer clips ranking higher regardless of dynamics. Standardizing frames reduces spurious length correlations by 54.0% while preserving motion-based correlation, so rankings reflect motion rather than duration. As in Fig. 4, normalization clarifies motion-specific patterns. For floating queries with frame-length normalization (left), top-ranked samples consistently show wave dynamics, floating objects, and surfing, all matching the target motion. Without normalization (right), top samples lack coherent similarity because rankings are driven by clip length, harming motion-relevant training example identification.

5. Conclusion

We address a central and underexplored question in video diffusion: which training clips influence the motion in generated videos? We introduced **Motive**, a motion-aware data attribution framework for video diffusion that isolates temporal dynamics from static appearance. By tracing generated motion back to influential training clips, our method enables targeted data curation that improves motion quality with a fraction of the data. As video models scale, such data-level understanding will be key to diagnosing failure modes and building more controllable generative systems.

Acknowledgements

We thank the following people (listed alphabetically by last name) for their helpful discussions, feedback, or participation in human studies: Allison Chen, Sanghyuk Chun, Zhiwei Deng, Amaya Dharmasiri, Xingyu Fu, Will Hwang, Yifeng Jiang, Amogh Joshi, Pang Wei Koh, Chen-Hsuan Lin, Huan Ling, Tiffany Ling, Shaowei Liu, Zhengyi Luo, Rafid Mahmood, Kaleb S. Newman, Julian Ost, Zeeshan Patel, Davis Rempe, Rulin Shao, Esin Tureci, Anya Tsvetkov, Jiachen T. Wang, Sheng-Yu Wang, Tingwu Wang, Zian Wang, Hongyu Wen, Jon Williams, Mengzhou Xia, Donglai Xiang, Yilun Xu, William Yang, and Haotian Zhang.

References

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 19

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023. 2

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. 8

Juhan Bae, Wu Lin, Jonathan Lorraine, and Roger B Grosse. Training data attribution via approximate unrolling. *Advances in Neural Information Processing Systems*, 37:66647–66686, 2024. 3, 19

David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. In *International Conference on Machine Learning*, pages 354–363. PMLR, 2018. 19

Yuxiang Bao, Di Qiu, Guoliang Kang, Baochang Zhang, Bo Jin, Kaiye Wang, and Pengfei Yan. Latentwarp: Consistent diffusion latents for zero-shot video-to-video translation. *arXiv preprint arXiv:2311.00353*, 2023. 19

Jesse Bettencourt, Xindi Wu, Matan Atzmon, James Lucas, and Jonathan Lorraine. Variance reduction for expectations with diffusion teachers. *arXiv preprint arXiv:2605.21489*, 2026. 19

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 1, 5, 19

Jonathan Brokman, Omer Hofman, Roman Vainshtein, Amit Giloni, Toshiya Shimizu, Inderjeet Singh, Oren Rachmil, Alon Zolfi, Asaf Shabtai, Yuki Unno, et al. Montrage: Monitoring training for attribution of generative diffusion models. In *European Conference on Computer Vision*, pages 1–17. Springer, 2024. 19

Weifeng Chen, Yatai Ji, Jie Wu, Hefeng Wu, Pan Xie, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video diffusion models with motion prior and reward feedback learning. *arXiv preprint arXiv:2305.13840*, 2023. 19

Zhaoxi Chen, Tianqi Liu, Long Zhuo, Jiawei Ren, Zeng Tao, He Zhu, Fangzhou Hong, Liang Pan, and Ziwei Liu. 4dnex: Feed-forward 4d generative modeling made easy. *arXiv preprint arXiv:2508.13154*, 2025. 8, 9

RDWS Cook et al. Residuals and influence in regression. 1982. 19

Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 25

Yuming Fang, Weisi Lin, Zhenzhong Chen, Chia-Ming Tsai, and Chia-Wen Lin. A video saliency detection model in compressed domain. *IEEE transactions on circuits and systems for video technology*, 24(1):27–38, 2013. 6

Alessandro Favero, Antonio Sclocchi, Francesco Cagnetta, Pascal Frossard, and Matthieu Wyart. How compositional generalization and creativity improve as diffusion models are trained. *arXiv preprint arXiv:2502.12089*, 2025. 1

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020. 19

Kristian Georgiev, Joshua Vendrow, Hadi Salman, Sung Min Park, and Aleksander Madry. The journey, not the destination: How data guides diffusion models. *arXiv preprint arXiv:2312.06205*, 2023. 19

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023. 19

Google DeepMind. Veo-3: Advancing controllable and physically plausible video generation. Technical report, Google DeepMind, 2025. URL https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf. 23

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 5

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024. 20

Zayd Hammoudeh and Daniel Lowd. Training data influence analysis and estimation: A survey. *Machine Learning*, 113(5):2351–2403, 2024. 19

Adam W Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Pavel Tokmakov, et al. Alltracker: Efficient dense point tracking at high resolution. *arXiv preprint arXiv:2506.07310*, 2025. 6

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022. 1, 5, 19

Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 19

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024. 8, 10, 20

Ruoxi Jia, Fan Wu, Xuehui Sun, Jiacen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, and Dawn Song. Scalability vs. utility: Do we have to sacrifice one for the other in data importance quantification? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 19

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. *arXiv preprint arXiv:2411.02385*, 2024. 1

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017. 3, 4, 19

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. *arXiv preprint arXiv:2310.00902*, 2023. 19

Quoc Viet Le, Tamás Sarlós, and Alexander Johannes Smola. Fastfood: Approximate kernel expansions in loglinear time. *arXiv preprint arXiv:1408.3060*, 2014. 5, 7

Jinxu Lin, Linwei Tao, Minjing Dong, and Chang Xu. Diffusion attribution score: Evaluating training data influence in diffusion models. *arXiv preprint arXiv:2410.18639*, 2024. 4, 5, 19

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 2

Jonathan Lorraine and David Duvenaud. Stochastic hyperparameter optimization through hypernetworks. *arXiv preprint arXiv:1802.09419*, 2018. 19

Jonathan Lorraine and Safwan Hossain. Jacnet: Learning functions with structured jacobians. *arXiv preprint arXiv:2408.13237*, 2024. 19

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International conference on artificial intelligence and statistics*, pages 1540–1552. PMLR, 2020. 19

Jonathan Lorraine, Paul Vicol, Jack Parker-Holder, Tal Kachman, Luke Metz, and Jakob Foerster. Lyapunov exponents for diversity in differentiable games. *arXiv preprint arXiv:2112.14570*, 2021. 19

Jonathan Lorraine, Nihesh Anderson, Chansoo Lee, Quentin De Laroussilhe, and Mehadi Hassen. Task selection for automl system evaluation. *arXiv preprint arXiv:2208.12754*, 2022a. 19

Jonathan P Lorraine, David Acuna, Paul Vicol, and David Duvenaud. Complex momentum for optimization in games. In *International Conference on Artificial Intelligence and Statistics*, pages 7742–7765. PMLR, 2022b. 19

Jonathan Peter Lorraine. *Scalable nested optimization for deep learning*. PhD thesis, University of Toronto (Canada), 2024. 19

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 23

Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, 1981. 19

Nikhil Mehta, Jonathan Lorraine, Steve Masson, Ramanathan Arunachalam, Zaid Pervaiz Bhat, James Lucas, and Arun George Zachariah. Improving hyperparameter optimization with checkpointed model weights. In *European Conference on Computer Vision*, pages 75–96. Springer, 2024. 19

Bruno Mlodozeniec, Runa Eschenhagen, Juhan Bae, Alexander Immer, David Krueger, and Richard Turner. Influence functions for scalable data attribution in diffusion models. *arXiv preprint arXiv:2410.13850*, 2024. 19

Koichi Namekata, Amirmojtaba Sabour, Sanja Fidler, and Seung Wook Kim. Emerdiff: Emerging pixel-level semantic knowledge in diffusion models. *arXiv preprint arXiv:2401.11739*, 2024. 1

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 1

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*, 2023. 3, 4, 5, 19

Yonghyun Park, Chieh-Hsin Lai, Satoshi Hayakawa, Yuhta Takida, Naoki Murata, Wei-Hsiang Liao, Woosung Choi, Kin Wai Cheuk, Junghyun Koo, and Yuki Mitsufuji. Concept-trak: Understanding how diffusion models learn concepts through concept-level attribution. *arXiv preprint arXiv:2507.06547*, 2025. 3, 19

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 1, 5, 19

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33, 2020. 3, 4, 19

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 22

Aniruddh Raghu, Jonathan Lorraine, Simon Kornblith, Matthew McDermott, and David K Duvenaud. Meta-learning to improve pre-training. *Advances in Neural Information Processing Systems*, 34:23231–23244, 2021. 19

Rahul Ravishankar, Zeeshan Patel, Jathushan Rajasegaran, and Jitendra Malik. Scaling properties of diffusion models for perceptual tasks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 1

Jessie Richter-Powell, Antonio Torralba, and Jonathan Lorraine. Score distillation sampling for audio: Source separation, synthesis, and beyond. *arXiv preprint arXiv:2505.04621*, 2025. 25

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 1

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35, 2022. 1

Qingyu Shi, Jianzong Wu, Jinbin Bai, Jiangning Zhang, Lu Qi, Yunhai Tong, and Xiangtai Li. Decouple and track: Benchmarking and improving video diffusion transformers for motion transfer. *arXiv preprint arXiv:2503.17350*, 2025. 19

Yanke Song, Jonathan Lorraine, Weili Nie, Karsten Kreis, and James Lucas. Multi-student diffusion distillation for better one-step generators. *arXiv preprint arXiv:2410.23274*, 2024. 25

Zhiyu Tan, Xiaomeng Yang, Luozheng Qin, and Hao Li. Vidgen-1m: A large-scale dataset for text-to-video generation. *arXiv preprint arXiv:2408.02629*, 2024. 5, 8, 9

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 19

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018. 3

Paul Vicol, Jonathan P Lorraine, Fabian Pedregosa, David Duvenaud, and Roger B Grosse. On implicit bias in overparameterized bilevel optimization. In *International Conference on Machine Learning*, pages 22234–22259. PMLR, 2022. 19

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong

Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 5, 19

Jiachen T Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. Data shapley in one training run. *arXiv preprint arXiv:2406.11011*, 2024a. 19

Jiachen T Wang, Tianji Yang, James Zou, Yongchan Kwon, and Ruoxi Jia. Rethinking data shapley for data selection tasks: Misleads and merits. *arXiv preprint arXiv:2405.03875*, 2024b. 19

Jiangshan Wang, Yue Ma, Jiayi Guo, Yicheng Xiao, Gao Huang, and Xiu Li. Cove: Unleashing the diffusion feature correspondence for consistent video editing. *Advances in Neural Information Processing Systems*, 37: 96541–96565, 2024c. 19

Sheng-Yu Wang, Alexei A Efros, Jun-Yan Zhu, and Richard Zhang. Evaluating data attribution for text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023a. 19

Sheng-Yu Wang, Aaron Hertzmann, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Fast data attribution for text-to-image models. *arXiv preprint arXiv:2511.10721*, 2025. 19

Xiang Wang, Shiwei Zhang, Han Zhang, Yu Liu, Yingya Zhang, Changxin Gao, and Nong Sang. Videolcm: Video latent consistency model. *arXiv preprint arXiv:2312.09109*, 2023b. 19

Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners. *arXiv preprint arXiv:2509.20328*, 2025. 1, 25

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023. 19

Xindi Wu, Mengzhou Xia, Rulin Shao, Zhiwei Deng, Pang Wei Koh, and Olga Russakovsky. Icons: Influence consensus for vision-language data selection. *arXiv preprint arXiv:2501.00654*, 2024. 7

Xindi Wu, Hee Seung Hwang, Polina Kirichenko, and Olga Russakovsky. Compact: Compositional atomic-to-complex visual capability tuning. *arXiv preprint arXiv:2504.21850*, 2025. 1

Tong Xie, Haoyu Li, Andrew Bai, and Cho-Jui Hsieh. Data attribution for diffusion models: Timestep-induced bias in influence estimation. *arXiv preprint arXiv:2401.09031*, 2024. 3, 4, 5, 19

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 5

Tianjun Zhang, Yi Zhang, Vibhav Vineet, Neel Joshi, and Xin Wang. Controllable text-to-image generation with gpt-4. *arXiv preprint arXiv:2305.18583*, 2023. 19

Xiaosen Zheng, Tianyu Pang, Chao Du, Jing Jiang, and Min Lin. Intriguing properties of data attribution on diffusion models. *arXiv preprint arXiv:2311.00500*, 2023. 3, 19

Shangchen Zhou, Peiqing Yang, Jianyi Wang, Yihang Luo, and Chen Change Loy. Upscale-a-video: Temporal-consistent diffusion model for real-world video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2535–2545, 2024. 19

Yixuan Zhu, Jiaqi Feng, Wenzhao Zheng, Yuan Gao, Xin Tao, Pengfei Wan, Jie Zhou, and Jiwen Lu. Astra: General interactive world model with autoregressive denoising. *arXiv preprint arXiv:2512.08931*, 2025. 25

A. Notation

Table 3: Glossary and notation.

Symbol	Description
Acronyms and Basic Notation	
VAE	Variational Autoencoder
DiT	Diffusion Transformer backbone
$\mathbf{I}$	Identity matrix
Video Generation	
$p_{\boldsymbol{\theta}}(\mathbf{v} \mid \mathbf{c})$	Conditional video generator with parameters $\boldsymbol{\theta}$
$\mathbf{v} \in \mathbb{R}^{F \times H \times W \times 3}$	Video clip with frames F, height H, width W
$\mathbf{c}$	Conditioning signal such as text or multimodal metadata
$\boldsymbol{\theta}$	Trainable model parameters
$f \in \{1, \ldots, F\}$	Frame index
$h \in \{1, \ldots, H\}$, $w \in \{1, \ldots, W\}$	Spatial indices for height and width respectively
$\tilde{h}$, $\tilde{w}$	Latent grid indices
Datasets	
$\mathcal{D} = \{(\mathbf{v}_n, \mathbf{c}_n)\}_{n=1}^{N}$	Training corpus with size N and index n
$\mathcal{D}_{\text{ft}} \subseteq \mathcal{D}$	Fine-tuning dataset
$\mathcal{S} \subseteq \mathcal{D}$	Selected influential subset
$k \in \{1, \ldots, K\}$	The selected subset size
Q	Number of query data
$q \in \{1, \ldots, Q\}$	Query index
$\mathbf{x}$	Generic input data pair
$\mathbf{x}_{\text{test}}$, $\mathbf{x}_n$	Test/query pair and training pair
$\hat{\mathbf{v}}$, $\hat{\mathbf{c}}$	Query video and its conditioning
Latent Space and Diffusion Components	
E	VAE encoder
$\mathbf{h} = E(\mathbf{v}) \in \mathbb{R}^{F \times (H/s) \times (W/s) \times C}$	Latent video with spatial factor s and channels C
$\mathbf{z}$	Noisy latent variable used in diffusion or flow matching
$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$	Gaussian noise
$\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}, \mathbf{c}, t)$	Predicted noise network in diffusion training
$\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}, \mathbf{c}, t)$	Time-indexed vector field in flow matching
$\dot{\mathbf{z}}$	Time derivative of the latent trajectory
α_t, σ_t	Scheduler signal and noise scales at timestep t
$\boldsymbol{\epsilon}_{\text{target}}$	Target noise or velocity used for supervision
$t \in \{1, \ldots, T\}$	Diffusion or flow-matching timestep, with total timesteps T
t_{fix}, $\boldsymbol{\epsilon}_{\text{fix}}$	Fixed timestep and shared noise draw used for low-variance gradients

Table 4: Glossary and notation (continued).

Symbol	Description
Attribution and Influence	
$I(\mathbf{v}_n, \hat{\mathbf{v}}; \boldsymbol{\theta})$	Influence score between a train clip and a query clip
$I_{\text{mot}}(\mathbf{v}_n, \hat{\mathbf{v}}; \boldsymbol{\theta})$	Motion-aware influence score
$\mathrm{TopK}(\cdot)$	Top-K operator for selecting highest scores
$\mathrm{MajVote}(\cdot)$	Majority-vote aggregation across queries
τ	Percentile cutoff for voting
ρ	Spearman correlation coefficient
Loss Functions	
$\mathcal{L}$	Generic loss
$\mathcal{L}_{\text{diff}}(\boldsymbol{\theta}; \mathbf{v}, \mathbf{c}), \mathcal{L}_{\text{flow}}(\boldsymbol{\theta}; \mathbf{v}, \mathbf{c})$	Diffusion and flow-matching objective
$\mathcal{L}_{\text{mot}}(\boldsymbol{\theta}; \mathbf{v}, \mathbf{c})$	Motion-weighted objective used for attribution
$\tilde{\mathcal{L}}$	Per-location squared error in latent space
$\mathbf{g}$, $\tilde{\mathbf{g}}$	Gradient and its projected version
$\mathbf{g}_{\text{mot}}$, $\tilde{\mathbf{g}}_{\text{mot}}$	Motion-weighted gradient and its projection
$\mathbf{H}_{\boldsymbol{\theta}}$	Hessian with respect to $\boldsymbol{\theta}$
Motion Representations	
$\mathcal{A}(\mathbf{v}) = A$	AllTracker motion extraction
$A \in \mathbb{R}^{F \times H \times W \times 4}$	Motion tensor containing flow, visibility, and confidence
$\mathbf{D}_f(h, w)$	Displacement vector at frame f and location (h, w)
$M_f(h, w)$	Motion magnitude at a location, computed from the displacement
$\mathbf{W}(f, h, w) \in [0, 1]$	Normalized motion weights used to mask per-location losses
ζ	Small numerical bias for stability (e.g., 10^{-6})
Projections and Computational Details	
D, D'	Full and projected gradient dimensions
$\mathbf{P} \in \mathbb{R}^{D' \times D}$	Projection matrix used for Fastfood-style JL projection
ξ	Variance normalization constant for projection
$\mathcal{T}$	Set of sampled $(t, \boldsymbol{\epsilon})$ pairs for gradient estimation
B	Unit compute cost used in complexity accounting

B. Extended Related Work

B.1. Data Attribution

Understanding how individual training examples shape model behavior has been a long-standing goal. Modern data attribution methods fall into two main groups [Hammoudeh and Lowd, 2024]: retraining-based methods (e.g., leave-one-out [Cook et al., 1982, Jia et al., 2021], downsampling (also known as subsampling or counterfactual influence) [Feldman and Zhang, 2020], Shapley-value [Wang et al., 2024a,b]) and gradient-based methods (influence-function family, including Influence Functions [Koh and Liang, 2017, Lorraine and Hossain, 2024], TracIn [Pruthi et al., 2020], and TRAK [Park et al., 2023]). Influence functions provide a principled framework by approximating the effect of removing a training point. TracIn [Pruthi et al., 2020] and TRAK [Park et al., 2023] make attribution feasible at scale. While effective for classification, these assume a mapping between training gradients and predictions, which becomes more complex in generative models.

Data attribution traces how individual training examples (or subsets) influence a model's predictions or behavior. Formally, it assigns an attribution score to each training sample, estimating the extent to which that sample contributes (positively or negatively) to the model's output on a given test query or behavior. Influence data attribution is an example of nested optimization [Vicol et al., 2022, Lorraine, 2024] with other examples including differentiable games [Balduzzi et al., 2018, Lorraine et al., 2021, 2022b], hyperparameter optimization [Raghu et al., 2021, Lorraine and Duvenaud, 2018, Mehta et al., 2024, Lorraine et al., 2020], and variance-reduced gradient estimation for diffusion-based teacher–student distillation [Bettencourt et al., 2026]. Before diffusion models, attribution methods were applied to supervised learning tasks such as classification and regression, where influence functions [Koh and Liang, 2017, Lorraine et al., 2022a] and scalable approximations such as TracIn [Pruthi et al., 2020], TRAK [Park et al., 2023], and TDA [Bae et al., 2024] quantified the impact of training examples on downstream predictions. Recent work adapted data attribution to diffusion models [Georgiev et al., 2023, Zheng et al., 2023, Wang et al., 2025, 2024a, Lin et al., 2024, Brokman et al., 2024, Kwon et al., 2023], where iterative denoising introduces timestep-dependent bias. Mlodozeniec et al. [2024] propose scalable approximations, while Xie et al. [2024] identify timestep-induced artifacts and normalization schemes. Concept-TRAK [Park et al., 2025] extends attribution to concepts by reweighting gradients with concept-specific rewards, enabling attribution to semantic factors. Wang et al. [2023a] instead design a customization-based benchmark for text-to-image models, where models are fine-tuned on exemplar images with novel tokens and attribution is evaluated by whether it can recover the responsible exemplars. However, these are limited to image diffusion, which captures static appearance but not temporal dynamics.

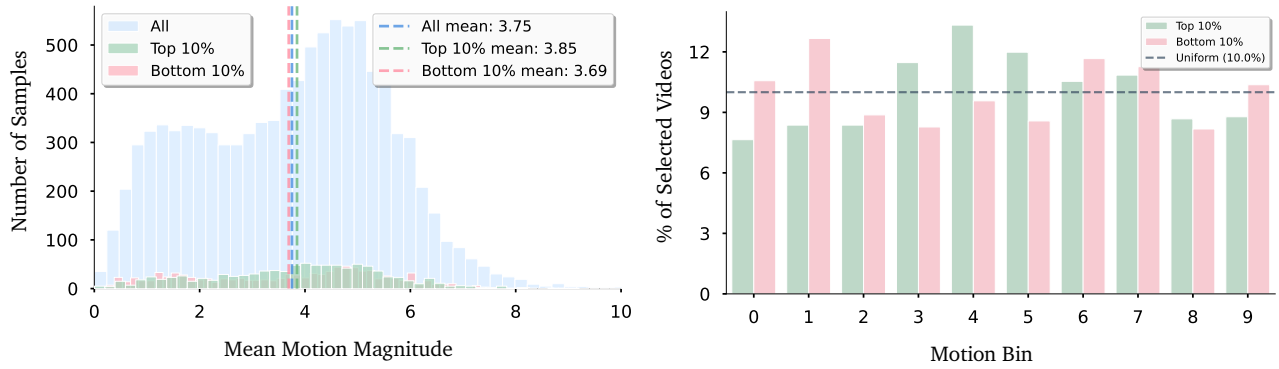
B.2. Motion in Video Generation

Video diffusion extends image generation to time, requiring coherent motion across frames [Ho et al., 2022, Blattmann et al., 2023, Peebles and Xie, 2023, Wan et al., 2025, Agarwal et al., 2025]. A large body of work builds temporal structure via attention layers [Wu et al., 2023], control signals [Chen et al., 2023, Zhang et al., 2023], feature correspondences [Geyer et al., 2023, Bao et al., 2023, Wang et al., 2024c], or consistency distillation [Wang et al., 2023b, Zhou et al., 2024]. Recent work has highlighted the challenge of decoupling motion from appearance in video diffusion transformers, in which spatial and temporal information become entangled within the model's representations [Shi et al., 2025]. However, understanding which training clips influence specific motion patterns in generated videos remains an open challenge.

In parallel, motion has long been studied using optical flow and correspondence, from classical formulations [Horn and Schunck, 1981, Lucas and Kanade, 1981] to modern approaches such as RAFT [Teed and Deng, 2020], which improve accuracy and generalization. These priors are often repurposed during generation to guide dynamics, but they do not explain which training examples shaped a model's motion behavior. Our work addresses both gaps by introducing a motion-aware data attribution framework specifically designed for video diffusion. We use motion-weighted gradients that disentangle temporal dynamics from static appearance, enabling us to trace generated motion patterns back to the most influential training clips.

Table 5: **VBench Evaluation on LTX-2B.** Performance comparison on VBench [Huang et al., 2024] across different baselines (all values in %, higher is better). All selection methods use 10% of training data; MM: motion masking.

	LTX-2B Video					
Method	Subject Consist.	Background Consist.	Motion Smooth.	Dynamic Degree	Aesthetic Qual.	Imaging Qual.
Base	93.8	95.2	94.6	36.5	32.1	48.2
Full FT	94.4	**95.7**	94.8	38.9	32.6	**49.1**
Random	93.7	95.4	94.7	37.8	**32.8**	48.4
Motion mag.	94.1	95.3	95.2	39.6	32.4	47.9
V-JEPA	94.2	95.5	95.1	40.2	31.9	47.5
Ours w/o MM	94.3	95.6	95.3	41.8	**32.8**	48.7
Ours (Motive)	**94.8**	95.6	**95.5**	**45.1**	32.7	48.6



(a) Motion Distribution (b) Distribution Across Motion Bins

Figure 6: **Motive is not simply selecting "motion-rich" clips.** Our influence scores are computed via gradients, and training videos are considered influential only when they directly improve the model's ability to generate the target motion dynamics, not because they contain more motion overall.

C. Additional Experiments

C.1. Results on Additional Video Generation Models

We further test our framework on additional video generation architectures beyond the `Wan` models shown in Sec. 4 (Tab. 1). We evaluate **Motive** on `LTX-2B` HaCohen et al. [2024], which represents a different architectural design and training paradigm. The results in Tab. 5 demonstrate that **Motive** works effectively across different model architectures and scales.

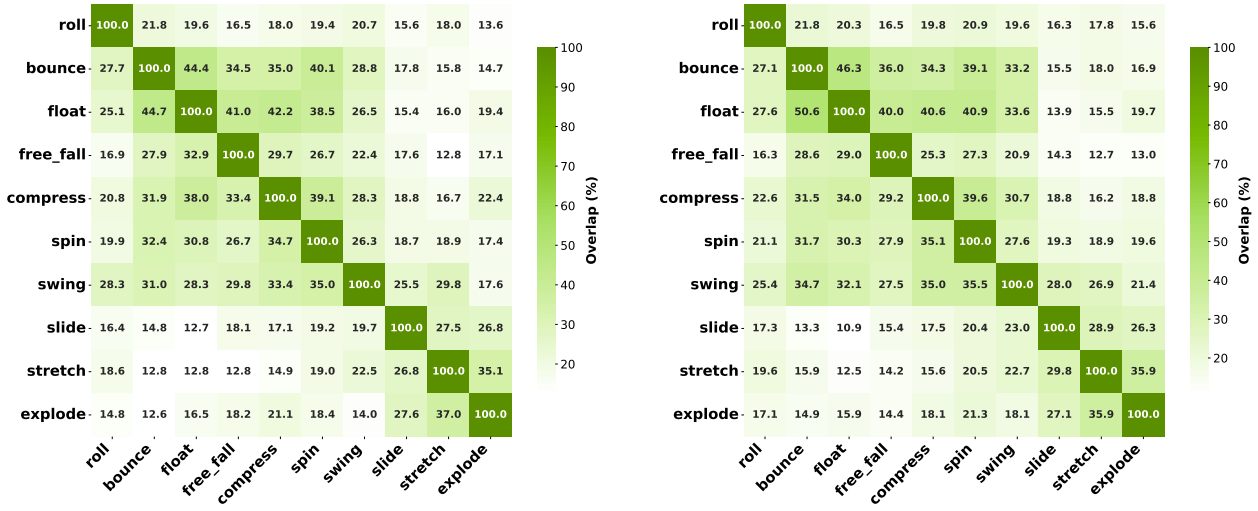
D. Analysis

D.1. Motion Distribution Analysis

Motive is not simply selecting "motion-rich" clips: The key distinction is that our influence scores are computed via gradients, and training videos are considered influential only when they directly allow the model to lower the loss, improving the model's ability to generate the target motion dynamics, not because they contain more motion overall.

To empirically validate this, we further analyze the distribution of motion magnitudes in our selected data. We compute the mean motion magnitude for 10k videos in the `VIDGEN` dataset and compare the distributions of the top 10% (highest influence scores) and the bottom 10% (lowest influence scores).

As shown in Fig. 6, the top 10% selected videos have a mean motion magnitude of 3.85, which is only 4.3%



(a) 4DNEX Influence Heatmap

(b) VIDGEN Influence Heatmap

Figure 7: **Cross-motion influence overlap across datasets.** Heatmaps showing the percentage overlap of top-100 influential training samples across motion categories for (a) 4DNEX and (b) VIDGEN datasets. Each cell (i, j) shows the percentage of motion category i's influential data (aggregated from 5 queries per category) that also appears in motion category j's top-100 influential samples. The asymmetric nature of the matrices (e.g., bounce→float $\neq$ float→bounce) arises because different motion categories have different numbers of unique influential videos, leading to directional overlap percentages. Consistent high-overlap pairs (e.g., bounce-float: $44.4\%/46.3\%$) and low-overlap pairs (e.g., free fall-stretch: $12.8\%/12.7\%$) across datasets validate that these influence patterns reflect fundamental aspects of motion representation in video generation models.

higher than the bottom 10% (3.69), despite representing opposite extremes of influence scores. The analysis also shows that within the moderate-motion range (bins 3, 4, and 5), the top 10% of positive-influence samples outnumber the bottom 10% of negative-influence samples. Yet, both groups also appear in low-motion bins (0-2) and high-motion bins (6-9).

This distribution pattern shows that high-influence videos selected by **Motive** span the entire motion spectrum, not just high-motion regions. Many high-motion videos receive low influence scores, while numerous influential videos exhibit modest or even low motion magnitude. These findings show that our motion attribution approach captures training influence, focusing on motion rather than simply acting as a motion-saliency filter.

D.2. Cross-Motion Influence Patterns

To analyze cross-motion influence patterns, we examine the percentage overlap of top-100 influential training data across different motion categories in both the 4DNEX and VIDGEN datasets. As described in §4, we use 5 query samples to identify the top-100 most influential training videos, aggregating results across queries. As shown in Fig. 7, both datasets exhibit remarkably similar patterns, with mean overlaps of 24.0% and 24.3%, respectively, indicating a moderate degree of sharing of influential data across motion categories.

Both datasets consistently identify the same high-overlap pairs: bounce-float ($44.4\%/46.3\%$), compress-float ($40.1\%/34.0\%$), and compress-spin ($36.9\%/39.6\%$), suggesting these motions share fundamental characteristics that the model learns from similar training examples. Conversely, low-overlap pairs such as free fall-stretch ($12.8\%/12.7\%$) and float-slide ($14.0\%/10.9\%$) indicate more specialized influential data for mechanically dissimilar motions. The influence matrices are asymmetric because the number of unique influential samples shared among the 5 query samples varies by motion category. The similar cross-motion influence patterns observed across both the 4DNEX and VIDGEN datasets demonstrate that these relationships are generalizable across different video datasets and reflect dynamic similarity.

Algorithm 1 Motive: Motion-Aware Data Attribution Framework

Require: fine-tuning corpus $\mathcal{D}_{\text{ft}} = \{(\mathbf{v}_n, \mathbf{c}_n)\}_{n=1}^{N}$, query video $(\hat{\mathbf{v}}, \hat{\mathbf{c}})$, fixed $(t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})$, projection matrix $\mathbf{P}$
Ensure: Motion-aware influence scores $\{I_{\text{mot}}(\mathbf{v}_n, \hat{\mathbf{v}})\}_{n=1}^{N}$

for $(\mathbf{v}_n, \mathbf{c}_n) \in \mathcal{D}_{\text{ft}}$ **do**
  $A_n \leftarrow \textsc{AllTracker}(\mathbf{v}_n)$ ▷ Extract per-pixel flow displacements $\mathbf{D}_f$ (Eq. 12)
  Downsample and normalize to latent-space motion mask $\mathbf{W}_n$ (Eqs. 13–15)
  Evaluate motion-weighted loss $\mathcal{L}_{\text{mot}}(\boldsymbol{\theta}; \mathbf{v}_n, \mathbf{c}_n)$ (Eq. 16)
  Compute motion gradient $\mathbf{g}_{\text{mot}}(\boldsymbol{\theta}, \mathbf{v}_n, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}}) = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{mot}}(\boldsymbol{\theta}; \mathbf{v}_n, \mathbf{c}_n, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})$
  Normalize by frame length: $\mathbf{g}_{\text{mot}} \leftarrow \mathbf{g}_{\text{mot}}/F$
  Project motion gradient: $\tilde{\mathbf{g}}_{\text{mot}}(\boldsymbol{\theta}, \mathbf{v}_n) := \frac{\mathbf{P}\mathbf{g}_{\text{mot}}(\boldsymbol{\theta}, \mathbf{v}_n, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})}{\|\mathbf{P}\mathbf{g}_{\text{mot}}(\boldsymbol{\theta}, \mathbf{v}_n, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})\|}$ (Eq. 17)
end for
Compute query gradient: $\tilde{\mathbf{g}}_{\text{mot}}(\boldsymbol{\theta}, \hat{\mathbf{v}})$ using the same procedure for $(\hat{\mathbf{v}}, \hat{\mathbf{c}})$
for $n = 1, \ldots, N$ **do**
  $I_{\text{mot}}(\mathbf{v}_n, \hat{\mathbf{v}}) = \tilde{\mathbf{g}}_{\text{mot}}(\boldsymbol{\theta}, \hat{\mathbf{v}})^{\top} \tilde{\mathbf{g}}_{\text{mot}}(\boldsymbol{\theta}, \mathbf{v}_n)$ (Eq. 17)
end for
Rank all training clips by $I_{\text{mot}}(\mathbf{v}_n, \hat{\mathbf{v}})$ and select top-K influential samples using majority vote aggregation (Eq. 18):

$$\mathcal{S} = \mathcal{S}_{\text{vote}}(K) = \left\{\mathbf{v}_n | \mathbf{v}_n \text{ in top-}K \text{ by } \mathrm{MajVote}\right\}$$

return $\mathcal{S}$

E. Additional Method Details

Tracker-agnostic scope. We treat the motion estimator as a pluggable source of saliency rather than a training dependency. Given displacement magnitudes, we construct latent-space weights via bilinear mapping and normalization. Our implementation supports alternative estimators (such as dense optical flow or point tracking) with identical interfaces, enabling users to swap AllTracker without modifying the attribution code.

Model-agnostic scope. Our attribution only requires per-example gradients under matched $(t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})$, and therefore applies to both diffusion and flow-matching objectives. The score reduces to a gradient inner product under a fixed preconditioner; the generator architecture affects gradient statistics but not the definition of influence. In practice, replacing the denoiser or velocity field leaves the weighting and aggregation unchanged.

Algorithm Summary. For completeness, Algorithm 1 summarizes the full **Motive** pipeline, detailing the computation of motion-weighted gradients, projection into low-dimensional space, and the subsequent influence-based ranking and selection of training clips.

F. Additional Experiment Details

F.1. Hyperparameter Settings

For reproducibility, we document the hyperparameters used throughout attribution, subset selection, and fine-tuning. Where values were not explicitly tuned, we adopted defaults from DiffSynth-Studio and the official Wan repository for Wan models and from Diffusion-Pipe for LTX models.

Attribution. Motion-aware influence estimation is computed at a single fixed timestep, the midpoint of the denoising trajectory, which strongly correlates with multi-timestep averaging. A shared Gaussian draw $\boldsymbol{\epsilon}_{\text{fix}} \sim \mathcal{N}(0, \mathbf{I})$ is used across all training–query pairs to reduce stochastic variance. Gradients are projected from dimension $D = 1\,418\,996\,800$ to $D' = 512$ using a Fastfood Johnson–Lindenstrauss projection $\mathbf{P}$ selected via the search in Fig.5 to balance performance and storage. Motion weights $\mathbf{W}$ are computed from AllTracker flow magnitudes M_f, min–max normalized to $[0, 1]$ with a small bias $\zeta = 10^{-6}$. All computations use bfloat16 precision for memory efficiency.

Subset Selection & fine-tuning. For any number of query points, we select top-10% data of the datasets. We finetune the backbone while freezing both the text encoder [Raffel et al., 2020] and the VAE. The input

resolution is fixed to 480×832 pixels. We use a learning rate of 1×10^{-5} and the AdamW optimizer [Loshchilov and Hutter, 2017] following the DiffSynth-Studio defaults. We train the models for 1 epoch, repeating the dataset 50 times.

Evaluation. The test set consists of the same 10 motion categories as the query set, but with different visual appearances. We provide the prompt samples below.

Samples from Query Set

We illustrate representative prompts from our query set used to generate query videos with Veo-3.

- **compress**: "A slice of sandwich bread flattened by a flat metal plate, steady camera, soft studio lighting, plain backdrop; emphasize air pockets collapsing."
- **bounce**: "A ping-pong ball bouncing on a white table, steady side camera, neutral light, seamless backdrop; emphasize consistent bounce height and timing."
- **roll**: "A spool of thread rolling from left to right, close-up static camera, bright studio light; highlight axle rotation and smooth travel."
- **explode**: "A single balloon bursting into fragments, captured in high-speed slow motion with a fixed camera, bright even lighting, seamless background; emphasize outward debris and air release."
- **float**: "A foam cube floating on the surface of water, static overhead camera, bright light, clean tank; emphasize buoyancy and slight rocking."

Samples from Test Set

We illustrate representative prompts from our test set that our fine-tuned models use to generate test videos.

- **compress**: "A rubber ball being compressed under a flat press, filmed with a stationary camera. Bright, shadow-free lighting and a clean background emphasize the deformation as it flattens."
- **bounce**: "A basketball bouncing vertically on a wooden court plank, unmoving camera, balanced indoor lighting, plain wall background; clearly show deformation at impact."
- **roll**: "A bike tire rolling freely on a stand, static side camera, indoor neutral light; show uniform rotation without wobble."
- **explode**: "A fragile glass ornament breaking apart mid-air, fixed camera, bright controlled lighting, plain backdrop; capture shards and reflections crisply."
- **float**: "A green leaf floating gently on perfectly still water in a transparent tank, fixed top-down camera, bright even lighting; emphasize surface tension ripples."

F.2. Details on Motion Query Data

A small, controlled set of query videos is constructed to isolate specific motion primitives while minimizing confounding factors (e.g., textured backgrounds, uncontrolled camera motion). Such clean and consistent clips are challenging to obtain from natural data sources. To address this, we synthesize the query set using Veo-3 [Google DeepMind, 2025] and apply a strict post-generation screening for physical plausibility and generation realism. We target ten motion types: *compress, bounce, roll, explode, float, free fall, slide, spin, stretch, swing*. For each category, we retain 5 query samples, yielding a total of 50 queries. This scale provides adequate coverage of the motion taxonomy used in our evaluations while maintaining tractable attribution computation. We further provide a few examples of the generation prompts and the generated video query set in Fig. 8.

Rationale for synthetic queries. The query set is not used as training data; instead, it specifies targets for attribution and for multi-query aggregation. Synthetic generation offers controllability that is difficult to achieve at scale with web videos. This design yields near-realistic yet standardized stimuli aligned with our goal of probing motion-specific influence.

Figure 8: **Illustration of motion query set.** We generate near-realistic video queries with Veo-3 across ten motion categories. Each category contains five query videos synthesized with controlled prompts and manually screened for clarity and physical plausibility.

Table 6: **Runtime Breakdown.** Detailed computational complexity and runtime for each component of our motion attribution framework on 10k training samples with `Wan2.1-T2V-1.3B` model.

Component	Complexity	Runtime	Notes
Gradient computation	$\mathcal{O}(B)$ per sample	Query: $\sim$ 54 seconds Training: $\sim$ 150 hours	1 A100 GPU; Single forward+backward pass; training is dominant cost but amortized over all queries; embarrassingly parallel (with 64 GPUs, $\sim$ 2.3 hours)
Projection	$\mathcal{O}(\lvert\mathcal{D}\rvert \cdot D' \log D')$	$\sim$ 1.97 seconds per sample	$D' = 512$
Influence computation	$\mathcal{O}(\lvert\mathcal{D}\rvert \cdot D')$	$\sim$ 46 milliseconds per query	1 query $\times$ 10k training samples
Majority-vote aggregation	$\mathcal{O}(\lvert\mathcal{D}\rvert \cdot q)$	$\sim$ 139 milliseconds	50 queries $\times$ 10k samples

G. Discussion

G.1. Runtime

All training runs are conducted on 4-8 NVIDIA A100 GPUs. We provide a detailed runtime breakdown from our experiments on 10k training samples with `Wan2.1-T2V-1.3B` model to address scalability concerns. The key insight is that the dominant cost of our pipeline is computing per-training-sample gradients, which is done once and can then be reused for all subsequent queries. Each training clip's gradient is projected into a compact 512-dimensional vector, and adding a new query requires only (i) a single backward pass to obtain its own projected 512-dimensional gradient and (ii) computing cosine similarity between the query vector and stored training vectors, which is exceptionally lightweight (on the order of seconds). Thus, the computational burden does not scale with the number of queries but only with the size of the training set. As shown in Tab. 6, the dominant cost is the one-time training data gradient computation for 10k samples ($\sim$ 150 hours on 1

Table 7: **Runtime Comparison with Baselines.** Total computational time required for processing 10k training samples with `Wan2.1-T2V-1.3B` model on a single GPU across different data selection methods.

Method	Random	Motion Magnitude	Optical Flow	V-JEPA	Ours
Total for 10k (1 GPU)	< 1 second	~ 5.5 hours	~ 5.7 hours	~ 3 hours	~ 150 hours

A100), which is amortized across all queries. Once computed, adding a new query requires only ~ 54 seconds (gradient computation) + 46ms (influence computation) = $\sim$54 seconds total. The training data gradient computation is embarrassingly parallel and can be reduced to ~ 2.3 hours with 64 GPUs.

Runtime comparison with baselines. We compare the computational cost of our method with the baseline approaches for processing 10k training samples on a single GPU (Table 7). While our method requires more upfront computation than the baseline approaches, this cost is amortized across all queries, and the computed gradients can be reused for multiple selection queries, making it practical for large-scale data curation scenarios.

G.2. Limitations

Gradient-based attribution is computationally expensive, requiring a high upfront cost for per-sample gradient computation (see §G.1), though this cost is amortized across queries. Our analysis treats each video as a whole unit, thereby avoiding collapsing motion into frame-level appearance, but it risks overlooking the fact that only certain segments may carry motion-relevant information. Highly informative intervals can be diluted when averaged with static or redundant portions of the same clip. This suggests an open direction toward finer-grained attribution at the motion-segment or motion-event level, which could yield more precise insights into how different phases of a trajectory shape motion learning. Another limitation is that our motion masks may overemphasize camera-only motion; we detect this by spatial uniformity of $\mathbf{W}$ and down-weight such clips, but a full disentanglement of ego and object motion remains future work.

Additionally, our framework does not explicitly account for classifier-free guidance (CFG), which is widely used in practice to steer video generation but introduces discrepancies between training-time attribution and inference-time dynamics. As a result, our influence estimates may not fully capture how guidance alters motion behavior. In addition, while attribution-driven fine-tuning improves targeted motion quality, it may introduce trade-offs in the base model's capabilities. This underscores the need for future work to balance targeted motion adaptation with the preservation of broader generative capabilities.

G.3. Future Directions

Improving the attribution pipeline. Several directions can strengthen our motion attribution methodology. *Tracker-robust motion saliency*: replace or ensemble AllTracker with alternative estimators and use its confidence/visibility channels to weight masks. *Self-generated video queries*: use model-generated videos as queries to trace problematic motion patterns (e.g., unrealistic physics) back to training data, enabling iterative diagnostics and targeted motion improvement.

Scaling data curation. *Closed-loop data curation*: move from one-shot ranking to active selection: iteratively attribute, finetune, and re-attribute, or replace simple majority voting with learned query weights. *Sophisticated finetuning*: move to more sophisticated finetuning setups, such as multi-student distillation [Song et al., 2024].

Extending to new domains. *Other modalities*: extend our methodology to other modalities, including world models [Zhu et al., 2025], audio [Evans et al., 2025, Richter-Powell et al., 2025], or video+audio [Wiedemer et al., 2025].

Safety and governance. Use negative-influence filtering to suppress undesirable or unsafe dynamics, document curator choices, and audit motion behaviors that our framework exposes.

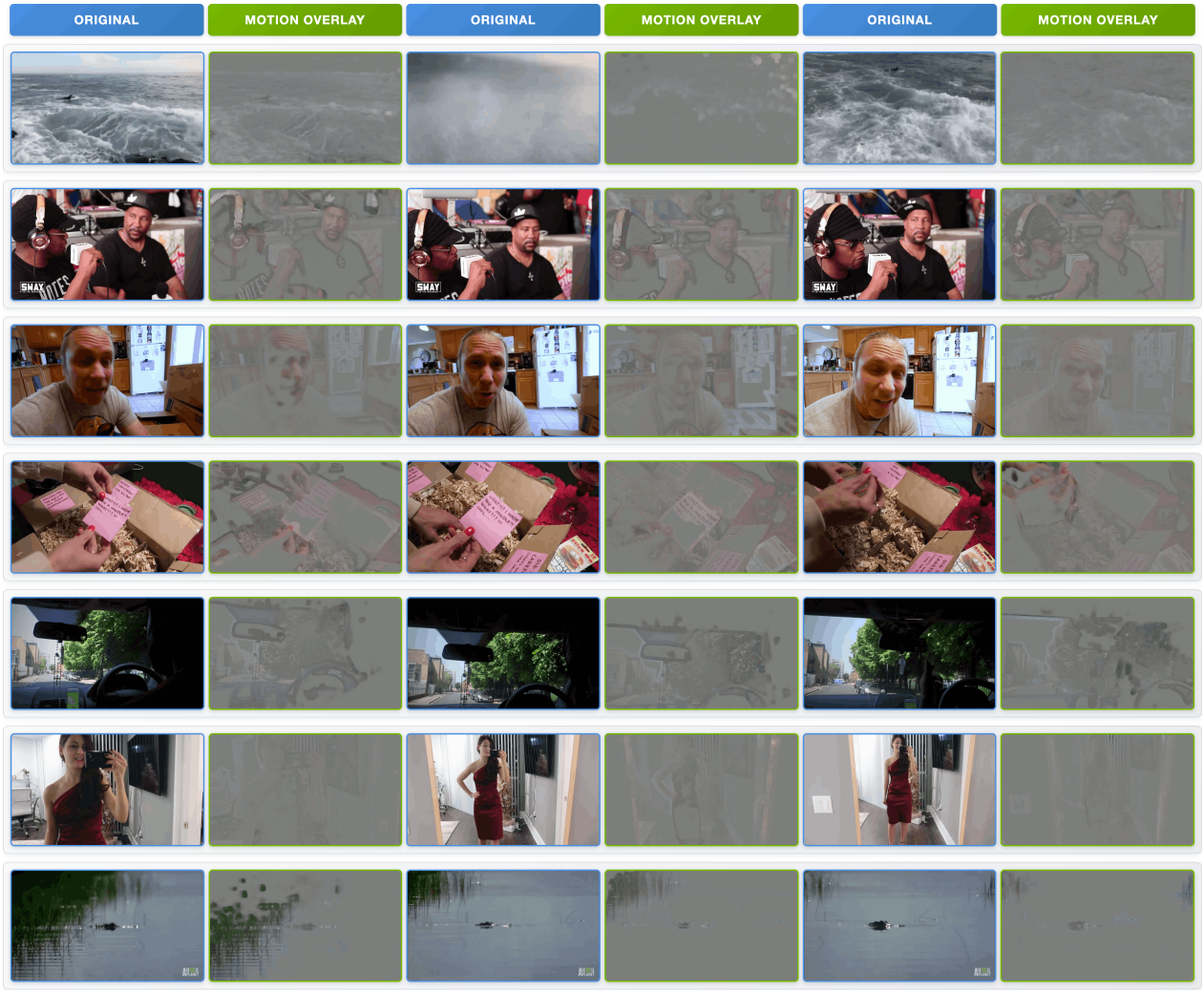


Figure 9: **Motion Overlay Visualization.** Comparison of original frames and motion overlays for seven video samples across three time points (early, middle, late). The motion overlay demonstrates the spatial weighting of our motion loss: dynamic regions remain visible, while static backgrounds are attenuated to neutral gray. *Takeaway:* This provides heuristic intuition into what information our motion attribution focuses on: the information in grayer regions, which lack motion, is down-weighted by our method.

H. Visualization

H.1. Motion Visualization

To provide intuition for the behavior of our motion-weighted loss, we visualize the motion magnitude as an overlay. We compute per-pixel motion magnitude using optical flow and apply motion-based weighting that preserves the appearance of dynamic regions while attenuating static backgrounds. This motion overlay directly illustrates the spatial weighting applied by our motion loss during training.

Fig. 9 presents representative frames from our dataset, comparing original frames with their corresponding motion overlays for seven distinct video samples. These visualizations show that the motion-weighted loss preferentially emphasizes dynamic content while down-weighting static scene elements.