Where is Motion From? Scalable Motion Attribution for Video Generation Models

Anonymous ICCV submission

Paper ID *****

Abstract

Despite the rapid progress of video generative models, the role of data in shaping motion quality is poorly understood. We present MOTIVE*(MOtion Training Influence 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 flow-weighted loss masks, yielding scalable influence scores practical for modern, large, and high-quality datasets and models. 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, our method improves both motion smoothness and dynamic degree on VBench, achieving a 76.7% human preference win rate compared with the pretrained base model. To our knowledge, this is the first framework that attributes motion (not just appearance) in video generative models and uses it to curate data.*

1. Introduction

Motion is the defining element of videos. Unlike image generation, which produces a single frame, video generation must capture how objects move, interact, and obey physical constraints. For video diffusion models, a central question remains:

> *Which training clips drive the motion observed in a video diffusion sample?*

Why it matters. Diffusion models are data-driven, and their progress has tracked the scaling of data and compute [14, 26, 29, 34]. Generative properties such as visual quality [33], semantic fidelity [25], and compositionality [8, 45] emerge from training data [4, 17, 32]. Motion is no exception. We use *motion* to mean 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 finetuning, where large pretraining corpora are inaccessible and carefully chosen clips can have an outsized impact. Motion-specific attribution is therefore especially valuable in the finetuning regime, where the goal is to identify which clips most influence temporal coherence and physical plausibility.

Why existing approaches fail 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.

Our method. We introduce MOTIVE, a motion attribution framework for video diffusion models that isolates motion-specific influence. MOTIVE computes gradients with motion-aware masking, so 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. The resulting scores trace generated motion back to training clips, enabling targeted curation and improving motion quality when used to guide fine-tuning.

2. Method

We formalize the problem setup in §C.3 and develop a practical framework for motion attribution in video diffusion models with three key components: scalable gradient computation (§2.1), frame-length bias fix (§2.2), motion-aware weighting (§2.3) and data selection for targeted finetuning (§C.1). We also provide a computational efficiency analy-

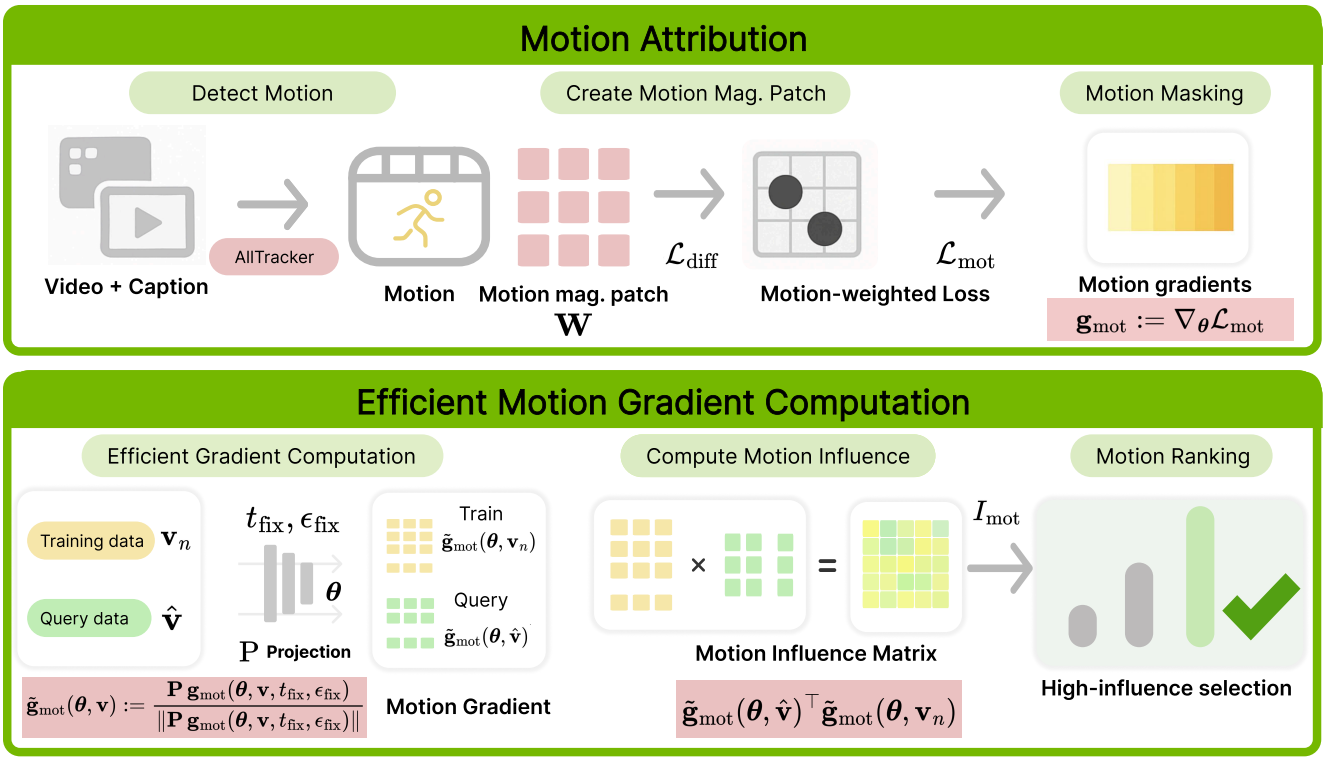

Figure 1. **Motive. Top.** Motion-gradient computation (§2.3) has three steps: (1) detect motion with AllTracker; (2) compute motion-magnitude patches; (3) apply loss-space motion masks to focus gradients on dynamic regions. **Bottom.** Our method (§2.1) is made scalable via a single-sample variant with common randomness and a projection, computed for each pair of training and query data, aggregated (§C.1) for a final ranking, and eventually used to select finetuning subsets.

sis (§C.4) demonstrating the scalability of our approach to billion-parameter models and large-scale video datasets.

2.1. Efficient Gradient-based Attribution for Generative Models

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 [18, 27, 30].

Common randomness for stable rankings. To reduce variance without changing the target, we evaluate train and test gradients under the same $(t, \boldsymbol{\epsilon})$ pairs and average over a small set $\mathcal{T}$ [20, 46]. This paired averaging stabilizes rankings compared to independent draws:

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

Single-sample variant for reduced compute. We then fix a single t_{fix} and a single shared draw $\boldsymbol{\epsilon}_{\text{fix}} \sim \mathcal{N}(0, \mathbf{I})$ for all train–test pairs at the final checkpoint. Sharing $(t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})$ is key to having low enough variance, for the low-cost single-sample estimator to maintain relative ordering [20, 46]. The estimator collapses to:

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

Structured projection for reduced storage. To operate at model scale, we apply a Johnson–Lindenstrauss projection via Fastfood [19] and normalize after projection. Let

$$\mathbf{P} \in \mathbb{R}^{D' \times D} \text{ be implemented as } \mathbf{P} := \frac{1}{\xi \sqrt{D'}} \mathbf{SQG\Pi QB} \tag{3}$$

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:

$$\text{Let } \mathbf{g} = \mathbf{P}\,\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{diff}}(\boldsymbol{\theta}, \mathbf{x}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})$$
$$\tilde{\mathbf{g}}(\boldsymbol{\theta}, \mathbf{x}) := \frac{\mathbf{g}}{\|\mathbf{g}\|} \tag{4}$$

Then the influence score is the compact cosine in $\mathbb{R}^{D'}$:

$$I^3_{\text{diff}}(\mathbf{x}_n, \mathbf{x}_{\text{test}}) \;=\; \tilde{\mathbf{g}}(\boldsymbol{\theta};\, \mathbf{x}_{\text{test}})^{\top}\; \tilde{\mathbf{g}}(\boldsymbol{\theta};\, \mathbf{x}_n) \tag{5}$$

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 [27].

2.2. Video-specific Frame-length Bias Fix

Raw gradient magnitudes depend on the number of frames F in the video $\mathbf{v}$, which biases scores toward video length. We correct this at measurement time by normalizing for frame count before the projection–normalization step:

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\, \mathbf{v}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}}) \;\leftarrow\; \frac{1}{F}\,\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\, \mathbf{v}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}}) \tag{6}$$

We still apply ℓ_2 normalization in Eq. 5, further stabilizing scales across examples. Together, single-timestep, common randomness, projection, and frame-length correction form a compact, scalable estimator that we use throughout. Fig. 3 shows the results of attributions with and without our fix. However, naïve video-level attribution conflates appearance with motion, often ranking clips high just because they share visual appearance, while offering little insight into dynamics.

2.3. 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 [4, 11, 29], 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 [14, 36, 39, 47]. 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.

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 [12] 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:

$$\mathbf{D}_f(h, w) = (A_{f,h,w,0}, A_{f,h,w,1}) = (\mathrm{d}w, \mathrm{d}h) \tag{7}$$

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 $[0, 1]$: 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 [7]. Let $(\tilde{h}, \tilde{w})$ index the latent grid. We obtain latent-aligned weights by bilinear downsampling:

$$\tilde{\mathbf{W}}(f, \tilde{h}, \tilde{w}) \;=\; \text{Bilinear}\big(\mathbf{W}(\cdot, \cdot, \cdot), F,\; \tfrac{H}{s},\; \tfrac{W}{s}\big) \tag{8}$$

We compute per-location squared error at fixed $(t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})$ at each frame f and "latent pixel" $(\tilde{h}, \tilde{w})$: and define the motion-weighted loss by averaging over frames and latent spatial locations:

$$\mathcal{L}_{\text{mot}}(\boldsymbol{\theta}; \mathbf{v}, \mathbf{c}) \;=\; \frac{1}{F_{\mathbf{v}}}\,\text{mean}_{f,\tilde{h},\tilde{w}}\left[\tilde{\mathbf{W}}_{\mathbf{v},\mathbf{c}}(f, \tilde{h}, \tilde{w}) \cdot \tilde{\mathcal{L}}_{\boldsymbol{\theta},\mathbf{v},\mathbf{c}}(f, \tilde{h}, \tilde{w})\right] \tag{9}$$

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:

$$I_{\text{mot}}(\mathbf{v}_n, \hat{\mathbf{v}}) = \tilde{\mathbf{g}}_{\text{mot}}(\boldsymbol{\theta}, \hat{\mathbf{v}})^{\top}\tilde{\mathbf{g}}_{\text{mot}}(\boldsymbol{\theta}, \mathbf{v}_n)$$
$$\text{where } \tilde{\mathbf{g}}_{\text{mot}}(\boldsymbol{\theta}, \mathbf{v}) := \frac{\mathbf{P}\mathbf{g}_{\text{mot}}(\boldsymbol{\theta}, \mathbf{v}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})}{\|\mathbf{P}\mathbf{g}_{\text{mot}}(\boldsymbol{\theta}, \mathbf{v}, t_{\text{fix}}, \boldsymbol{\epsilon}_{\text{fix}})\|}$$
$$\text{and } \mathbf{g}_{\text{mot}} := \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{mot}} \tag{10}$$

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. More details about subset selection are in Appendix C.1.

Method (↓) / Metric (→)	Subject Consist.	Background Consist.	Motion Smooth.	Dynamic Degree	Aesthetic Quality	Imaging Quality
Base	95.3	96.4	96.3	82.3	45.3	**65.7**
Full finetuning	95.9	**96.6**	96.3	84.7	45.0	63.9
Random selection	95.3	96.6	96.3	81.6	45.7	65.1
Whole video	95.4	96.1	96.3	85.3	45.7	63.2
MOTIVE (Ours)	**96.3**	96.1	96.3	**89.4**	**46.0**	64.6

Table 1. **VBench Evaluation.** Performance comparison on VBench [16]. We evaluate subject consistency, background consistency, motion smoothness, dynamic degree, aesthetic quality, and imaging quality across different data selection methods (all values in %, higher is better). Random selection and our MOTIVEboth select 10% of the training data, with our method using majority vote aggregation (§ C.1) across all motion queries.

3. Experiment

3.1. Main Results

Experiment setup details are in Appendix §F.2. **High-influence selection and negative filtering.** Fig. 6 shows that 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 carrying objects; planetary rotation). Negatives are mostly static footage, camera-only motion, or cartoon clips whose simplified kinematics do not transfer to natural scenes. Our procedure promotes informative motions and filters data that would dilute temporal learning during finetuning. These trends hold across categories and align with the quantitative gains that follow.

Qualitative improvements across motion types. Fig. 5 compares the base pretrained model, naïve motion finetuning, and our motion-aware data selection for finetuning across four scenarios. Top: rubber-ball compression and coin spinning. Bottom: coffee mug sliding and red ball drop. Our method yields higher motion fidelity and temporal consistency than both baselines, especially for complex deformation, rotational dynamics, and physics-driven motion.

Quantitative Results. We evaluate our approach across different metrics using VBench [16], demonstrating consistent improvements in motion fidelity when finetuning with attribution-selected data compared to random sampling or naïve approaches. As shown in Tab. 1, MOTIVE achieves the highest dynamic degree score (89.4%), significantly outperforming random selection (81.6%) and whole video attribution (85.3%). Our method also excels in subject consistency (96.3%) and aesthetic quality (46.0%), while maintaining competitive motion smoothness (96.3%). Using 10% of the training data, our approach surpasses the full finetuned model on dynamic degree (84.7%) and subject consistency (95.9%), demonstrating the superior empirical performance of motion-specific attribution for targeted finetuning.

Table 2. **Human evaluation results.** Pairwise comparisons across 10 motion categories. Win, tie, and loss rates show the percentage of comparisons where our method is preferred, rated equal, or outperformed by each baseline method.

Method	Win	Tie	Loss
vs. Base	**76.7**	10.0	13.3
vs. Random	**66.7**	13.3	20.0
vs. Full FT	**57.5**	15.0	27.5
vs. Whole Video	**50.8**	12.5	36.7

3.2. 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 6 annotators and evaluate 10 motion categories. For each category, we prepare two 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 the table, annotators favor our attribution-guided selection: 76.7% win rate vs. the base model and 57.5% vs. the full finetuned model, indicating perceptually meaningful motion improvements.

4. Conclusion

We address a central and underexplored question in video diffusion: where is motion from? We propose MOTIVEthat traces generated dynamics back to influential training clips by isolating motion-specific gradients. Unlike image-based attribution, our method directly targets temporal dynamics, revealing how coherence and physical plausibility emerge from data. Our results show that motion learning is traceable to specific examples, providing a quantitative tool for diagnosing artifacts and enabling targeted data selection and curation. This enables more controllable and interpretable video diffusion models, and as models scale, such data-level understanding will be essential for building robust and reliable generative systems.

References

[1] 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. 7

[2] 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. 9

[3] 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. 9

[4] 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, 3, 9

[5] 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. 9

[6] 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. 12, 13

[7] 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. 3

[8] 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

[9] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023. 9

[10] Google DeepMind. Veo-3: Advancing controllable and physically plausible video generation. Technical report, Google DeepMind, 2025. 11

[11] 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. 3

[12] 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. 3

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 7

[14] 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, 3, 9

[15] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 9

[16] 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. 4, 12

[17] 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

[18] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017. 2, 8, 9

[19] Quoc Viet Le, Tamás Sarlós, and Alexander Johannes Smola. Fastfood: Approximate kernel expansions in loglinear time. *arXiv preprint arXiv:1408.3060*, 2014. 2, 9

[20] 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. 2, 8

[21] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 7

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 12

[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*, pages 674–679, 1981. 9

[24] 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. 9

[25] 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

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

[27] Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*, 2023. 2, 3, 8, 9

[28] 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. 8, 9

[29] 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, 3, 9

[30] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930, 2020. 2, 8, 9

[31] 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. 12

[32] 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*, pages 12945–12954, 2025. 1

[33] 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

[34] 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*, 35:36479–36494, 2022. 1

[35] 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. 9

[36] 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. 3, 12, 13

[37] 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. 9

[38] 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. 8

[39] 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. 3, 9

[40] 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, 2024. 9

[41] 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*, pages 7192–7203, 2023. 9

[42] 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*, 2023. 9

[43] 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*, pages 7623–7633, 2023. 9

[44] 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. 8

[45] Xindi Wu, Dingli Yu, Yangsibo Huang, Olga Russakovsky, and Sanjeev Arora. Conceptmix: A compositional image generation benchmark with controllable difficulty. *Advances in Neural Information Processing Systems*, 37:86004–86047, 2024. 1

[46] 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. 2, 9

[47] 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. 3

[48] 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. 9

[49] 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. 8

[50] 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. 9

Appendices

A. Notation

B. Background

A table of notation is in App. §A, as well as an extended related work in App. §D.

B.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:

$$\mathbf{z}(t, \boldsymbol{\epsilon}) \;=\; \alpha_t\, \mathbf{h} \;+\; \sigma_t\, \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \quad t \in \{1, \ldots, T\} \tag{11}$$

Denoising diffusion [13] trains a network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}, \mathbf{c}, t)$ to predict the injected noise:

$$\mathcal{L}_{\text{diff}}(\boldsymbol{\theta}; \mathbf{v}, \mathbf{c}) \;=\; \mathbb{E}_{t,\boldsymbol{\epsilon}}\Big[\big\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}(t, \boldsymbol{\epsilon}), \mathbf{c}, t) - \boldsymbol{\epsilon}\big\|_2^2\Big] \tag{12}$$

Flow matching [1, 21] learns a time-dependent vector field $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}_t, \mathbf{c}, t)$ that matches the instantaneous velocity $\dot{\mathbf{z}} = \frac{d}{dt}\mathbf{z}$ induced by a chosen interpolant:

$$\mathcal{L}_{\text{flow}}(\boldsymbol{\theta}; \mathbf{v}, \mathbf{c}) \;=\; \mathbb{E}_{t,\boldsymbol{\epsilon}}\Big[\big\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}(t, \boldsymbol{\epsilon}), \mathbf{c}, t) - \dot{\mathbf{z}}(t, \boldsymbol{\epsilon})\big\|_2^2\Big] \tag{13}$$

The two objectives are closely related and both train time-indexed predictors over the same latent space while integrating over t and $\boldsymbol{\epsilon}$. Methods, such as for attribution, relying on per-sample gradients naturally transfer between them, since the estimation challenges arise from the similar integrations.

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 §2: (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 (§2.1). (ii) *Variable horizon.* Clips vary in F and frame rate (§2.2). (iii) *Time-specific failure modes.* Typical artifacts include inconsistent trajectories, temporal flicker, identity drift, and physically implausible dynamics despite sharp individual frames (§2.3).

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 will use to provide masks in our motion-weighted loss in §2.

B.2. Data Attribution

Data attribution connects model behavior to specific training examples by giving examples a score for their contribution to a target prediction or loss. A classic formulation is influence functions [18]. For a loss $\mathcal{L}(\boldsymbol{\theta};\mathbf{x})$ and a test input $\mathbf{x}_{\text{test}}$, the influence of training point $\mathbf{x}_n$ is:

$$I(\mathbf{x}_n,\mathbf{x}_{\text{test}}) = -\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta};\mathbf{x}_{\text{test}})^{\top}\,\mathbf{H}_{\boldsymbol{\theta}}^{-1}\,\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta};\mathbf{x}_n)$$
$$\mathbf{H}_{\boldsymbol{\theta}} = \tfrac{1}{N}\sum\nolimits_{n=1}^{N}\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta};\mathbf{x}_n) \qquad (14)$$

Exact Hessian inverses are infeasible at modern scales, so practical methods approximate influence via gradient similarity, for example, TracIn [30] and TRAK [27].

Attribution in diffusion models. Diffusion training aggregates gradients over timesteps t and noise draws $\boldsymbol{\epsilon}$, and gradient norms vary systematically with t. This produces a timestep bias where examples aligned with large-norm timesteps appear spuriously influential. Diffusion-ReTrac [20] reduces this bias by normalizing gradients and sub-sampling t and $\boldsymbol{\epsilon}$ when forming 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 between normalized test and train gradients:

$$I_{\text{diff}}(\mathbf{x}_n,\mathbf{x}_{\text{test}}) = \left(\frac{1}{|\mathcal{T}_{\text{test}}|}\sum_{t,\boldsymbol{\epsilon}\in\mathcal{T}_{\text{test}}}\frac{\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\mathbf{x}_{\text{test}},t,\boldsymbol{\epsilon})}{\left\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\mathbf{x}_{\text{test}},t,\boldsymbol{\epsilon})\right\|}\right)^{\top} \times \left(\frac{1}{|\mathcal{T}_n|}\sum_{t,\boldsymbol{\epsilon}\in\mathcal{T}_n}\frac{\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\mathbf{x}_n,t,\boldsymbol{\epsilon})}{\left\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{diff}}(\boldsymbol{\theta};\mathbf{x}_n,t,\boldsymbol{\epsilon})\right\|}\right) \qquad (15)$$

Averaging 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 [49].

Why vanilla attribution is insufficient for videos. Naïvely applying gradient-based attribution to video diffusion risks treating appearance and motion alike, often overemphasizing low-level appearance matches (objects, textures, backgrounds) while overlooking dynamics [28, 38]. Its cost grows with clip length, sampled timesteps, noise draws, and gradient dimensionality, making naïve methods impractical at modern video scales. Because we aim to explain and improve motion, we need attribution that suppresses static appearance, emphasizes motion-specific signals, and remains efficient, motivating the motion-centric approach in §2. Motion is distributed across frames and temporal horizons and entangled with static cues, so influence cannot be assigned by considering frames independently.

C. Additional Method Details

C.1. Most Influential Finetuning Subset Selection

Goal. Given a query clip $(\hat{\mathbf{v}},\hat{\mathbf{c}})$, we compute a motion-aware attribution value for each candidate finetuning example $(\mathbf{v}_n,\mathbf{c}_n) \in \mathcal{D}_{\text{ft}}$ using: $I_{\text{mot}}(\mathbf{v}_n,\hat{\mathbf{v}})$ from Eq. 10. Then, we construct a finetuning dataset $\mathcal{S}$ for one or many query videos $\hat{\mathbf{v}}$.

Single-query-point finetuning 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.

Multi-query-point finetuning selection: aggregating attribution scores. For Q queries, we adopt the majority voting approach from ICONS [44] 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 finetuning subset. This formulation emphasizes samples that are consistently influential across multiple queries, without requiring cross-query calibration of raw scores.

$$\mathrm{MajVote}_n = \sum\nolimits_{q=1}^{Q}\mathbb{I}\big[\,I_{\text{mot}}(\mathbf{v}_n,\hat{\mathbf{v}}_q) > \tau\,\big]$$
$$\mathcal{S}_{\text{vote}}(K) = \big\{\mathbf{v}_n \,|\, \mathbf{v}_n \text{ in top-}K \text{ by } \mathrm{MajVote}\big\} \qquad (16)$$

C.2. Scope

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 allows for the use of alternate estimators (such as dense optical flow or point tracking) with identical interfaces, enabling practitioners to swap AllTracker without modifying the attribution code.

Model-agnostic scope. Our attribution only requires per-example gradients under matched $(t,\boldsymbol{\epsilon})$, 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.

C.3. Problem Formulation

We study data attribution for motion in the finetuning setting. Let $\mathcal{D}_{\text{ft}} = \{(\mathbf{v}_n, \mathbf{c}_n)\}_{n=1}^{N}$ be the finetuning 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) predictivity: rankings correlate with observed changes doing finetuning 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. 15 to be (a) lower variance for stable rankings with feasible levels of compute, (b) more scalable to store, and (c) motion-centric.

Finetuning Subset Selection. For a budget $K \ll N$, we get a motion-influential subset by ranking scores and taking the top-K examples. When aggregating across multiple query motions, we combine selections as described in §2. The resulting subsets serve as candidates for motion-centric finetuning.

C.4. 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 $\mathcal{O}(|\mathcal{D}|\,B)$ – essential for having a reasonable cost on modern video datasets and models – while re-using the same sample across data allows the single-sample to have low enough variance for stable rankings. Projection adds $\mathcal{O}(D' \log D')$ per example using Fastfood [19], 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. 5 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 $\mathcal{O}(|\mathcal{D}| \log |\mathcal{D}|)$.

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.

D. Related Work

D.1. Data Attribution

Understanding how individual training examples shape model behavior has been a long-standing goal in machine learning. Influence functions [18] provide a principled framework by approximating the effect of removing a training point. Extensions such as TracIn [30] and TRAK [27] make attribution feasible at scale. While effective for classification, these methods assume a direct mapping between training gradients and predictions, which becomes more complex in generative models.

Data attribution refers to methods that trace 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. Before diffusion models, attribution methods were applied in supervised learning tasks such as classification and regression, where influence functions [18] and scalable approximations like TracIn [30], TRAK [27], and TDA [2] quantified the impact of training examples on downstream predictions. Recent work adapted data attribution to diffusion models, where iterative denoising introduces timestep-dependent bias. Mlodozeniec et al. [24] propose scalable approximations, while Xie et al. [46] identify timestep-induced artifacts and normalization schemes. Concept-TRAK [28] extends attribution to concepts by reweighting gradients with concept-specific rewards, enabling attribution to semantic factors. Wang et al. [41] 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 works are limited to image diffusion, which captures static appearance but not temporal dynamics.

D.2. Motion in Video Generation

Video diffusion extends image generation to time, requiring coherent motion across frames [4, 14, 29, 39]. A large body of work builds temporal structure via attention layers [43], control signals [5, 48], feature correspondences [3, 9, 40], or consistency distillation [42, 50]. Recent work has highlighted the challenge of decoupling motion from appearance in video diffusion transformers, where spatial and temporal information become entangled in the model's representations [35]. However, understanding *which* training clips influence specific motion patterns in generated videos remains an open challenge.

In parallel, motion has long been studied through optical flow and correspondence – from classical formulations [15, 23] to modern deep flows like RAFT, which improves accuracy and generalization [37]. These priors are often

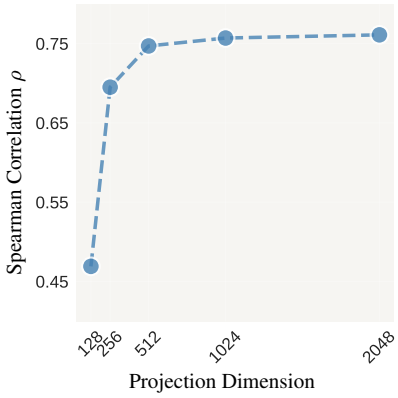

Figure 2. **Projection dimension analysis.** Spearman correlation between projected and full gradients shows rapid improvement with projection dimension, with 512 providing a strong trade-off between accuracy and efficiency.

repurposed in generation for guiding dynamics or checking temporal consistency, 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.

D.3. Ablations

Single-timestep attribution. Using a single timestep avoids the cost of averaging across timesteps while closely matching the multi-timestep baseline. With a fixed $t = 500$, we obtain $\rho = 68\%$ agreement with the 7-timestep setting $t \in \{150, 300, 400, 500, 600, 750, 850\}$. Using the same timestep for train and test is key to preserving relative rankings. Early timesteps (e.g., 250) add little noise; late timesteps (e.g., 750) heavily corrupt inputs and can obscure motion cues. $t = 500$ strikes a balance, delivering high correlation and substantial compute savings. Averaging multiple timesteps yields minimal ranking gains, and incorporating late-timestep gradients can bias rankings. A single fixed timestep is therefore sufficient for variance-reduced, scalable attribution.

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. 2). 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.

Frame-Length Normalization. Following the Wan training protocol, we standardize all videos to 81 frames at 16 fps (satisfying the $4n+1$ constraint) to enable fair attribution across clips of different raw lengths. Without standardization, gradient-based scores correlate strongly with video length rather than motion quality ($\rho = 78.0\%$), causing longer clips to rank higher regardless of dynamics. Standardizing frames reduces this spurious length correlation by 54.0% while preserving motion-based correlation, so rankings reflect motion rather than duration. As in Fig. 3, normalization clarifies motion-specific patterns. For floating queries, with frame-length normalization (left), top-ranked samples consistently show wave dynamics, floating objects, and surfing, which match the target motion. Without normalization (right), top samples lack coherent similarity because rankings are driven by clip length, hindering identification of motion-relevant training examples.

Samples from Motion Query Set

We illustrate representative prompts in our query set that are 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 a
nd 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."
```

Figure 3. **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.

E. Details on Motion Query Data

A small, controllable set of query videos is constructed to isolate specific motion primitives while minimizing confounds (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 [10] 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 five query samples, resulting in 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. 4.

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.

Samples from Motion Test Set

We illustrate representative prompts in our test set that are used to generate test videos with our finetuned models.

```
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. Details on Experiments

F.1. Hyperparameter Settings

For reproducibility, we document the hyperparameters used throughout attribution, subset selection, and finetuning. Where values were not explicitly tuned, we adopted defaults from DiffSynth-Studio and the official **`wan`** repo.

Attribution. Motion-aware influence estimation is computed at a single fixed timestep $t_{\text{fix}} = 500$, selected as a

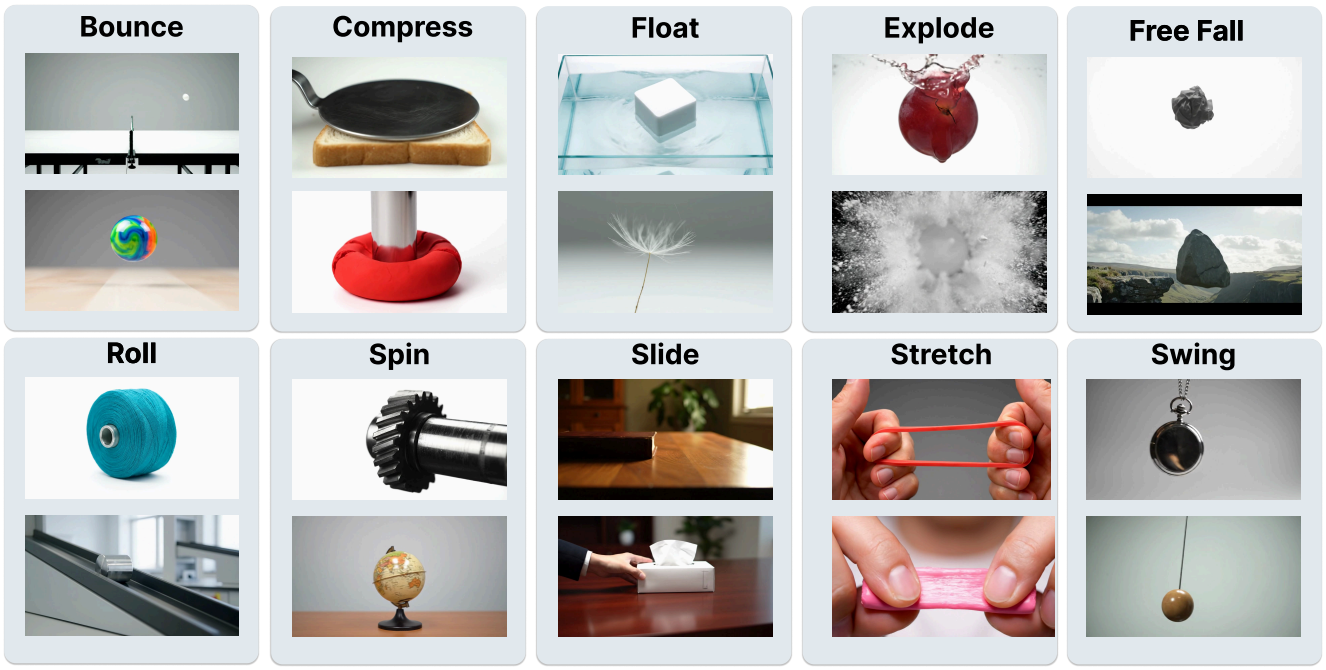

Figure 4. **Illustration of motion query set.** We generate near-realistic video queries with Veo-3 across ten motion categories. Each category contains five query videos synthesized with controlled prompts and manually screened for clarity and physical plausibility.

mid-range value that correlates strongly with multi-timestep averaging. A shared Gaussian draw $\epsilon_{\text{fix}} \sim \mathcal{N}(0, \mathbf{I})$ 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.2 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 & Finetuning. For any number of query points, we select top-10% data of the datasets. We finetune the `Wan2.1-T2V-1.3B` backbone while freezing both the T5 text encoder [31] 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 [22] following the DiffSynth-Studio defaults. We train the models for 1 epoch with the dataset repeated 50 times.

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

F.2. Setup

Finetuning Datasets. We evaluate our motion attribution framework on two large-scale video datasets: `VIDGEN-1M` [36] and `4DNeX-10M` [6], both offering diverse motion patterns with 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 types of 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 multiple motion categories, with a focus on object dynamics: compress, bounce, roll, explode, float, free fall, slide, spin, stretch, swing. Each motion type is represented by 5 videos, totaling 50 queries. These videos are chosen to exhibit clear and isolated motions, serving as the basis for evaluating attribution quality and downstream motion generation. Further details on query video curation are provided in App. E.

Model & Baselines. All experiments use pretrained `Wan2.1-T2V-1.3B`[1] as the primary video diffusion model, a widely used open-source baseline with strong performance and feasible compute. Our baselines: **Base model** (pretrained, no finetuning); **Random selection** (uniform sampling at our budget); **Full finetuning** (approximate upper bound using the complete dataset); and **Whole video attribution** (whole video level influence without motion-specific weighting).

Benchmark. We evaluate our motion attribution framework with VBench [16], a video generation benchmark. VBench provides evaluation across dimensions, including subject and background consistency, motion smoothness,

[1] https://huggingface.co/Wan-AI/Wan2.1-T2V-1.3B

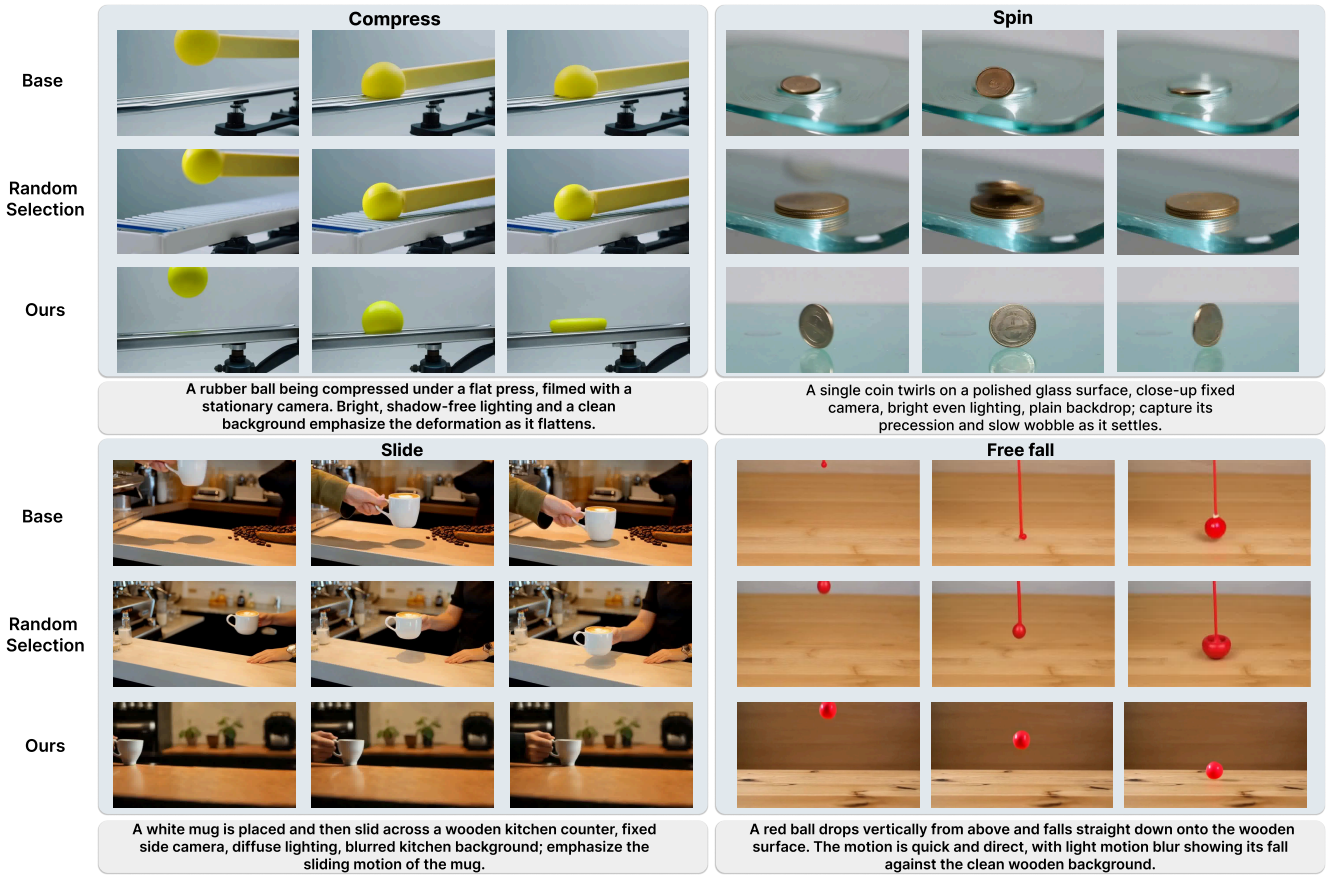

Figure 5. **Qualitative Comparisons.** We compare four motion scenarios – compress, spin, slide, and free fall – across the base model, random selection, and our method. Our approach yields more realistic dynamics than baselines, with improved object deformation under compression, consistent rotation of the spinning coin, smoother pendulum motion of the swinging key, and more accurate gravitational acceleration during free fall. Supplementary videos are included.

dynamic degree, and aesthetic and imaging quality. Since our method targets temporal dynamics, motion smoothness and dynamic degree are most relevant; other metrics confirm that improvements do not sacrifice visual or semantic consistency.

Implementation Details. We finetune `Wan2.1-T2V-1.3B` with our MOTIVE-selected high-quality video data following the official & DiffSynth-Studio [2] implementation. During finetuning, we update only the DiT backbone while freezing the T5 text encoder and VAE. All models are trained at 480×832 resolution with a learning rate of $1e{-}5$. Specialist models are trained on single motion category selected data while generalist models use aggregated selections (both with top 10% selection from `VIDGEN-1M` [36] or `4DNeX` [6] with motion-weighted loss attribution). All training runs are conducted on 4-8 NVIDIA A100 GPUs. We use one A100 GPU, taking approximately 150 hours to compute the influence score of 10k samples.

[2] https://github.com/modelscope/DiffSynth-Studio

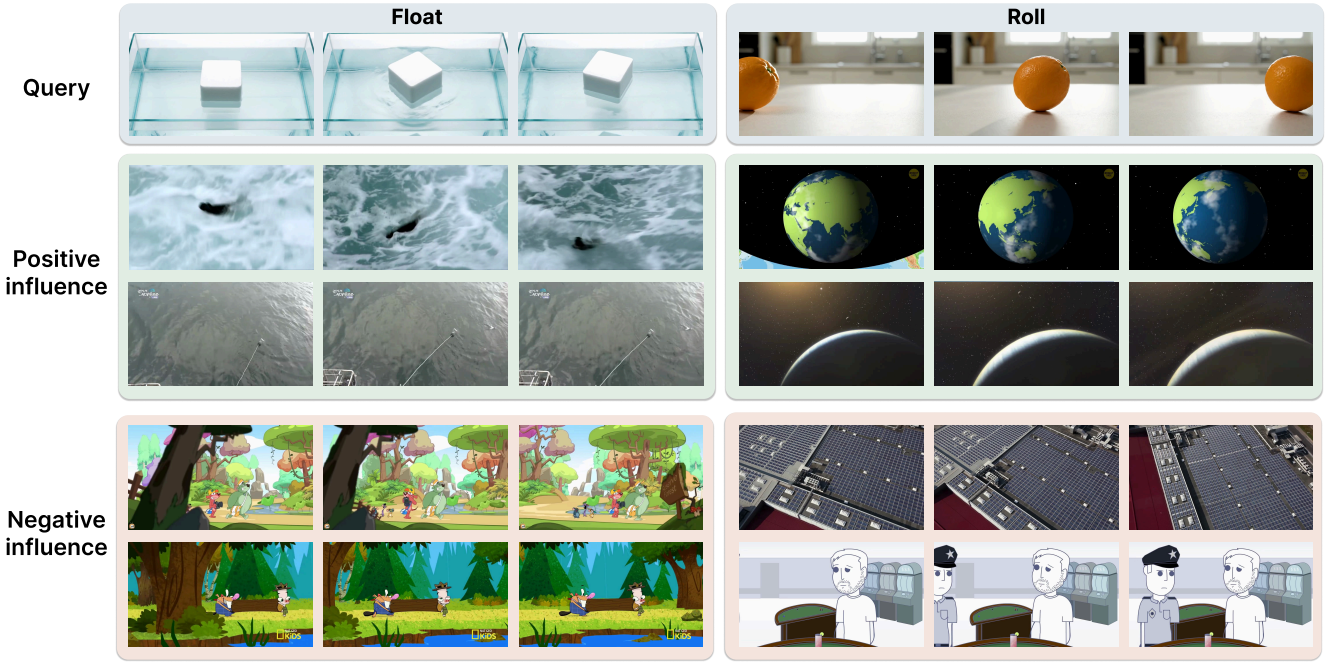

Figure 6. **Motion attribution examples.** Top row: Query clips showing floating (*right*) and rolling (*left*) motions. Middle rows: Top-ranked positive influential training samples by MOTIVE that share similar motion patterns, such as water flow with floating objects and rotations. These samples exhibit similar motion patterns and dynamics that help the model generate realistic motion, demonstrating how our method identifies motion-relevant examples across different visual appearances and object types. Bottom rows: Negative samples including minimal motion content, camera-only motion, and cartoon-style videos, which conflict with the target motions.

Table 3. Glossary and notation.

VAE	Variational Autoencoder
DiT	Diffusion Transformer backbone
$\mathbf{I}$	Identity matrix
$p_{\boldsymbol{\theta}}(\mathbf{v} \mid \mathbf{c})$	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
$t \in \{1, \ldots, T\}$	Diffusion or flow-matching timestep, with total timesteps T
$\mathcal{D} = \{(\mathbf{v}_n, \mathbf{c}_n)\}_{n=1}^{N}$	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
$q \in \{1, \ldots, Q\}$	Number of query clips
$\hat{\mathbf{v}}$, $\hat{\mathbf{c}}$	Query video and its conditioning
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_{fix}, $\boldsymbol{\epsilon}_{\text{fix}}$	Fixed timestep and shared noise draw used for low-variance gradients
$\mathcal{L}$	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 $\boldsymbol{\theta}$
$I(\mathbf{v}_n, \hat{\mathbf{v}}; \boldsymbol{\theta})$	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
$\text{TopK}(\cdot)$	Top-K operator for selecting highest scores
$\text{MajVote}(\cdot)$	Majority-vote aggregation across queries
τ	Percentile cutoff for voting
ρ	Spearman correlation coefficient
$\mathcal{A}(\mathbf{v}) = A$	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
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