# Auteur: Language-Driven Cinematographic Framing for Human-Centric Video Generation

**M. Burak Kizil**
Koç University, UCL

**Enes Sanli**
Koç University

**Niloy J. Mitra**
UCL, Adobe

**Xuelin Chen**
Adobe

**Erkut Erdem**
Hacettepe University

**Aykut Erdem**
Koç University

**Duygu Ceylan**
Adobe

## Abstract

Generative video models have achieved remarkable visual fidelity and temporal coherence, yet intentional camera control remains elusive. Existing frameworks treat camera motion as a byproduct of pixel synthesis, producing trajectories that are stochastic, spatially inconsistent, and indifferent to the human subject driving the scene. In this work, we present Auteur, a method for language-driven, human-centric camera *framing* in generative video. Our core insight is that professional filmmakers conceive shots not as world-space trajectories but as framings defined relative to the actor, encoding shot size, angle, and composition as functions of human pose and motion. We formalize this intuition as a human-centric camera parameterization and introduce a Domain-Specific Language (DSL) that is convertible to standard 6-DoF camera parameters. A fine-tuned multimodal large language model then acts as a virtual *director*, mapping natural language descriptions and coarse human motion to sparse DSL keyframes that are deterministically interpolated into continuous camera trajectories, which are then provided as input to video generators. We train and evaluate Auteur on a new dataset of 34K aligned text, human motion, and DSL-annotated camera trajectories drawn from procedural synthesis and real-world movie footage from the CondensedMovies (Bain et al., 2020) dataset. Auteur enables cinematographic framing of human-centered scenes, a capability largely absent in prior generative models. To assess this behavior, we propose new framing-focused metrics, and our experiments show that Auteur consistently outperforms existing methods in framing quality.

*"There are no good and bad movies, only good and bad directors."*
— FRANÇOIS TRUFFAUT

## 1 Introduction

The essence of cinematography lies not in simply recording a scene but in intentionally orchestrating perspective to evoke narrative meaning. While recent generative video models (Kong et al., 2024; OpenAI, 2025; Wan et al., 2025; Google DeepMind, 2025) have made remarkable strides in visual fidelity and temporal coherence, they lack the structured, deliberate camera behavior required to tell a coherent visual story. Current methods for camera control in video generation enable direct conditioning on explicit 6-DoF geometric trajectories (He et al., 2025; Cheong et al., 2025) but specify them in world-space coordinates, decoupled from the actors on-screen. Often, transferring a trajectory extracted from reference footage to a new scene produces shots that are geometrically plausible but narratively meaningless; the camera moves, but not in response to the subject's performance. Object-centric approaches (Kizil et al., 2025) and sampling-based alternatives (Courant et al., 2026) reduce subject-frame violations but still do not capture the deliberate synchronization between actor dynamics and camera response that defines cinematography.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

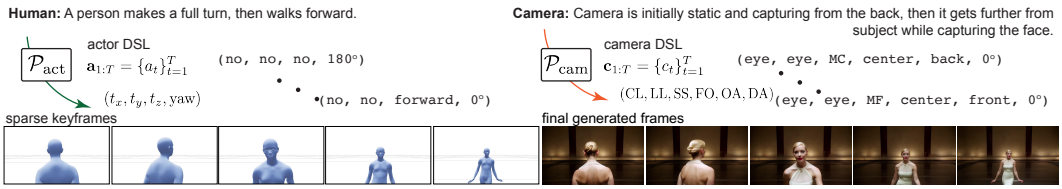



Figure 1: **Auteur defines every camera decision relative to the human subject:** how much of the body is visible (*scale*), from which angle the actor is viewed (*orientation*), and how they are placed in the frame (*composition*). Given a natural-language description (top), a fine-tuned multimodal LLM generates a structured DSL program (both actor and camera DSLs) that encodes these actor-relative parameters at sparse keyframes, which are then deterministically interpolated into a continuous 6-DoF camera trajectory conditioned on the human mesh (blue). The trajectory then guides video synthesis.

Directors of photography do not conceive of shots as mere raw spatial trajectories (Mascelli, 1965; Katz, 1991; Bordwell and Thompson, 2020). Instead, they reason in terms of *framing*: shot size (close-up, medium, wide), camera angle (eye-level, low, overhead), and screen composition (lead room, headroom), all defined relative to the actor's body and movement over time. Cinematography is therefore not an independent geometric construct, but a *function* of the subject's position, movement, and narrative role within the scene. We argue that a generative system capable of authentic cinematography must transition from a passive observer to an active *auteur*: an agent that deliberatively *authors* camera behavior with awareness of the humans and their space-time choreography.

Building on this insight, we introduce Auteur, a method for composing human-aware camera motion from natural language. Our approach is grounded in three key components: a human-centric camera representation, a two-stage generative formulation, and a dataset that anchors both in real or synthetic footage. At the core of Auteur is a *human-centric camera parameterization* that represents camera state along semantic axes, namely orientation, shot scale, camera level, framing, and look-at level, all defined relative to the actor's body coordinate frame derived from SOMA (Saito et al., 2026). We implement this continuous parameterization as a *Domain-Specific Language* (DSL): a structured, discrete form that is human-readable and interpretable, generated by a large language model, and can be deterministically converted to standard 6-DoF camera parameters. The DSL can be viewed as a *quantization* of the underlying parameterization, mapping a continuous space into a vocabulary of cinematographically meaningful values while preserving full geometric expressiveness.

Building upon this representation, Auteur operates in two stages. First, a fine-tuned multimodal LLM, Qwen-2.5-VL (Qwen Team, 2025) maps a natural language description to a coarse 3D human trajectory, capturing the subject's spatial evolution. Second, conditioned on this trajectory, the LLM acts as a virtual director, generating DSL programs that specify camera state at sparse, scene-defining keyframes, such as shot initiations and transitions. These keyframes are deterministically interpolated into a continuous camera trajectory that is both geometrically consistent and cinematographically meaningful, and is used to condition downstream video generation models.

To train Auteur, we construct a dataset combining sythetic procedural and real-world footage. For procedural data, we synthesize scenes by pairing randomized SOMA (Saito et al., 2026) motion sequences with DSL-driven camera programs, rendering them via a 3D engine to obtain (human motion, camera trajectory, DSL program) triplets. For real-world footage, we process videos from CondensedMovies (Bain et al., 2020) to recover 4D human motion and camera trajectories (Wang et al., 2024a); captions are generated with AuroraCap (Chai et al., 2024), and DSL annotations are obtained via our motion-tagging strategy. This yields 34K aligned (caption, 3D human motion, camera trajectory, DSL program) tuples that ground our learning of human-aware cinematic priors in real filmmaking practice. We supply the generated camera trajectories by Auteur to multiple downstream video generation models, demonstrating improved spatial coherence, subject alignment, and perceived cinematic quality over prior methods. In summary, our contributions are:

- **Human-centric camera parameterization and DSL.** An actor-relative representation of camera state grounded in professional cinematographic conventions, with *framing*, rather than raw trajectory, as the primary compositional primitive. The DSL is its discrete LLM-generatable operationalization, deterministically convertible to standard 6-DoF parameters.

- **Language-to-human-to-camera pipeline.** A two-stage method mapping natural language to coarse 3D human trajectories, then to sparse DSL keyframe programs that are deterministically interpolated into continuous, actor-aware 6-DoF camera paths.
- **Auteur dataset.** A dataset of 34K (caption, SOMA parameters over time, camera trajectory, DSL program) samples from real data, enabling learning of human-aware cinematic priors.
- **Comprehensive cross-dataset evaluation** We introduce Auteur Score, a new framing-centric benchmark suite measuring subject visibility, compositional adherence, temporal framing stability, and actor-camera coordination. Across both our internal Auteur benchmark and the external PulpMotion benchmark, our method outperforms prior controllable camera baselines (Kizil et al., 2025; Courant et al., 2026), achieving stronger controllability, more stable cinematic framing

## 2 Related Work

**Explicit camera conditioning in generative video.** Large-scale video diffusion models (e.g., Sora, Wan, Veo) produce compelling photorealistic visual priors of scene content but treat viewpoint as an uncoupled, passive byproduct of the sampling process. Recent work inject user-specified camera signals through a range of geometric interfaces: raw extrinsics (Wang et al., 2024c), Plücker embeddings (He et al., 2025), dense point trajectories (Feng et al., 2025; Li et al., 2025; Gokmen et al., 2025; Zheng et al., 2025; Jin et al., 2025), and persistent 3D geometry (Ren et al., 2025; Wang et al., 2025b; Agarwal et al., 2025; Wang et al., 2025a). A complementary line uses point-cloud renderings or multi-camera setups to enforce geometric adherence (Hou and Chen, 2025; Yu et al., 2025; Bai et al., 2025; Lin et al., 2026a; Zheng et al., 2026). While these methods substantially improve viewpoint controllability, they all treat the camera trajectory as an externally authored, independent variable. As a result, they function as conditional renderers $p(\text{video} \mid \text{camera}, \text{text})$ rather than addressing the prior question of *how* a camera should move to meaningfully frame the actors and their actions. Auteur instead generates the trajectory itself, conditioned on the actors' physical state, which can then be fed into camera-conditioned generators such as those above.

**Foundations of cinematic grammar.** Classical film theory formalizes visual storytelling as a structured grammar. Seminal works codify a precise vocabulary of shot scales and framing constraints (Mascelli, 1965; Mercado, 2010) and articulate rules for actor staging, blocking, and shot-by-shot directorial planning (Katz, 1991; Bordwell and Thompson, 2020). Our DSL (Section 3.1) operationalizes this vocabulary as a parameterized, machine-generatable representation, bridging abstract directorial intent and programmable continuous camera control.

**Framing-aware planning and virtual cinematography.** Prior work inspired by classical virtual cinematography has long formalized screen-space composition, visibility, and shot design through methods (e.g. Through-the-Lens control (Gleicher and Witkin, 1992) and Toric-space parameterizations (Lino and Christie, 2015)). These are later extended to drone cinematography as a constrained optimization over smoothness, occlusion, and target framing (Galvane et al., 2018). In the generative setting, E.T. (Courant et al., 2024) and GenDoP (Zhang et al., 2025) learn camera paths from text and character trajectories, Cheng et al. (2025) refine paths via Toric interaction features, and the concurrent VERTIGO (Li et al., 2026) post-trains a camera generator against learned visual preferences. Most closely related to ours, PulpMotion (Courant et al., 2026) introduces an auxiliary framing latent derived from screen-space projected joints to encourage coherence. These methods tie camera behavior to actors or cinematic heuristics, but each derives the camera from text, discrete rules, auxiliary sampling, or 2D screen-space proxies, but never from a parameterized 3D body-space representation. A separate line of work (Jiang et al., 2024; Li et al., 2024) jointly generates camera and video without conditioning on actor performance, while LLM-based scene planners (Pan et al., 2025; Lin et al., 2024; Song et al., 2025) decompose text into multi-scene prompts that downstream generators render, again without modeling the camera as a response to actor motion. While the recent LAMP (Kizil et al., 2025) work models camera tracking behavior from natural language, it does not handle the explicit framing of the camera, which is the gap Auteur addresses.

**Actor-aware camera movement.** Existing joint human–camera systems (Wang et al., 2024b; Cao et al., 2025) model both entities but largely expose the camera as a parallel, decoupled control (input) stream or rely on external descriptors for coupling. We instead formulate cinematography as an actor-conditioned framing task. By operating directly over a unified parametric body manifold via SOMA (Saito et al., 2026)), Auteur learns camera behavior as a conditional distribution $p(\text{camera} \mid$

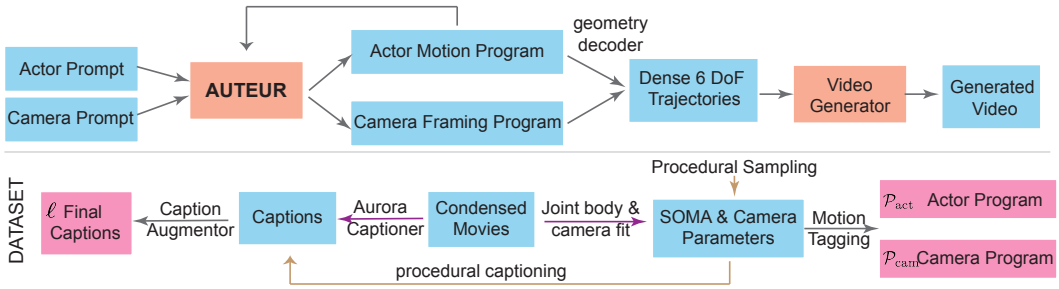



Figure 2: **Overview of Auteur.** Given a text prompt, Auteur predicts a coarse actor motion program and a camera framing program. The camera program specifies sparse actor-relative framing keyframes in our DSL. These are embedded interpolated over time, and decoded into 6-DoF trajectories that can be used for conditioning video generators. Auteur is trained with caption, actor, and camera program tuples $(\ell, \mathcal{P}_{\text{act}}, \mathcal{P}_{\text{cam}})$ obtained procedurally and from real footage Bain et al. (2020).

action), where the action is the SOMA-parameterized space-time trajectory of the actor. This shifts the paradigm from treating the camera as a geometric input or text-derived heuristic to a learned policy that responds to the subject and is consumable by downstream video generators.

## 3 Method

**Overview.** Auteur is a two-stage framework that maps free-form natural language description $\ell$ to (i) a coarse 3D actor trajectory $\mathbf{a}_{1:T} = \{a_t\}_{t=1}^{T}$ and (ii) an actor-aware 6-DoF camera trajectory $\mathbf{c}_{1:T} = \{c_t\}_{t=1}^{T}$ suitable for conditioning downstream video generators. At the core of our approach is a *human-centric camera parameterization* (Section 3.1) that expresses the camera state at each frame in the actor's corresponding local body frame, together with a *cinematographic Domain-Specific Language* (DSL) that discretizes this parameterization into a compact, LLM-generatable program. Sparse DSL keyframes are deterministically interpolated into a dense camera trajectory (Section A.2). A fine-tuned multimodal LLM acts as a *virtual director* (Section 3.2): it first generates the actor trajectory and then, conditioned on it, emits the sparse DSL keyframes. The resulting trajectory drives a range of camera-conditioned video generators (Section 3.3). Figure 2 summarizes the pipeline.

### 3.1 Human-Centric Camera Parameterization

**Why not world-space 6-DoF?** World-space $\mathrm{SE}(3)$ trajectories are the de-facto interface for camera-conditioned video models, yet they are an unnatural medium for cinematographic intent. A trajectory recorded around one actor and replayed around another produces shots that are geometrically valid but narratively misfit; the camera moves through space, but not *with respect to* the subject's pose, facing, or action. Professional filmmakers instead reason and operate in terms of *framing*, which include shot scale, camera angle, and screen composition, all defined relative to the actor's body (Mascelli, 1965; Katz, 1991; Bordwell and Thompson, 2020).

**Actor state and body frame.** We model the actor at frame $t \in \{1, \ldots, T\}$ as,

$$a_t := \big(\mathbf{p}_t,\ \psi_t,\ h_t\big) \in \mathbb{R}^3 \times \mathbb{S}^1 \times \mathbb{R}_{>0}, \tag{1}$$

where $\mathbf{p}_t$ is the pelvis position, $\psi_t$ is the body yaw (facing angle in the ground plane), and $h_t$ is the body height in meters; all derived from the SOMA parametric body model (Saito et al., 2026) in A-pose. We define the actor-anchored coordinate frame as,

$$\mathcal{F}_t := \big(\mathbf{p}_t,\ \hat{\mathbf{f}}_t,\ \hat{\mathbf{r}}_t,\ \hat{\mathbf{z}}\big), \qquad \hat{\mathbf{f}}_t = (\cos\psi_t, \sin\psi_t, 0), \qquad \hat{\mathbf{r}}_t = \hat{\mathbf{z}} \times \hat{\mathbf{f}}_t, \tag{2}$$

with $\hat{\mathbf{z}}$ the world up-axis. All camera parameters are expressed in this actor-centric local $\mathcal{F}_t$.

**Framing state.** A camera state is represented by six actor-relative variables. We let $\mathcal{A} := \{\mathrm{CL}, \mathrm{LL}, \mathrm{SS}, \mathrm{FO}, \mathrm{OA}, \mathrm{DA}\}$ index the camera axes; each axis $a \in \mathcal{A}$ takes values in a continuous set $\mathcal{K}_a$ summarized in Table A.1. They are:
(i) **CL** (Camera Level) $\in \mathcal{K}_{\mathrm{CL}} = \mathbb{R}_{\geq 0}$. Camera height above the ground plane.
(ii) **LL** (Look-at Level) $\in \mathcal{K}_{\mathrm{LL}} = [0, h_t]$. Height on the actor at which the optical axis is aimed.

(iii) **SS** (Shot Scale) $\in \mathcal{K}_{\text{SS}} = (0, 1]$. Fraction of frame height occupied by the actor's body.
(iv) **FO** (Framing Offset) $\in \mathcal{K}_{\text{FO}} = [-\frac{1}{2}, +\frac{1}{2}]$. Lateral placement of the actor in normalized image coordinates, capturing lead-room and rule-of-thirds composition conventions.
(v) **OA** (Orientation Angle) $\in \mathcal{K}_{\text{OA}} = \mathbb{S}^1$. Camera azimuth around the actor, measured from $\hat{\mathbf{f}}_t$.
(vi) **DA** (Dutch Angle) $\in \mathcal{K}_{\text{DA}} = \mathbb{S}^1$. Camera roll about the optical axis, encoding stylistic tilt.

The joint continuous state space is the product manifold

$$\mathcal{K} \;=\; \mathcal{K}_{\text{CL}} \times \mathcal{K}_{\text{LL}} \times \mathcal{K}_{\text{SS}} \times \mathcal{K}_{\text{FO}} \times \mathcal{K}_{\text{OA}} \times \mathcal{K}_{\text{DA}}, \tag{3}$$

a smooth 6-dimension manifold. Unlike world-space trajectories, this space describes *how the actor should be framed*; the 6-DoF camera pose, $\mathbf{k}_t \in \mathcal{K}$, arises as a consequence of this framing decision.

**Cinematographic DSL representation.** The continuous spaces in which the camera framing state and the actor trajectory live are geometrically expressive but unsuitable as direct language-model outputs. Token-level regression of real numbers is brittle, and continuous predictions discard the inductive bias of an established cinematographic vocabulary. We therefore introduce a unified *Domain-Specific Language* (DSL) that quantizes both spaces along semantic axes into compact, human-readable, LLM-generatable programs.

Formally, given a finite axis set $\mathcal{A}$ with per-axis continuous domains $\{\mathcal{K}_a\}_{a\in\mathcal{A}}$ and finite vocabularies $\mathcal{V}_a \subset \mathcal{K}_a$ of cinematographic tokens (e.g., $\mathcal{V}_{\text{SS}} = \{\text{ECU}, \text{CU}, \ldots, \text{EWS}\}$ for shot scale; full vocabularies in Appendix A.1), a DSL *program* is a sparse ordered sequence of *shot records*,

$$\mathcal{P} \;=\; \big((\tau_k, \delta_k)\big)_{k=1}^{K}, \qquad 1 \le \tau_1 < \cdots < \tau_K \le T, \tag{4}$$

where $\tau_k$ is a keyframe index and $\delta_k$ is a *partial assignment*

$$\delta_k \in \Delta\mathcal{V} \;:=\; \bigsqcup_{S\subseteq\mathcal{A}} \prod_{a\in S} \mathcal{V}_a, \tag{5}$$

specifying new values only for the axes $\text{dom}(\delta_k) \subseteq \mathcal{A}$ that change at $\tau_k$. The carry-forward rule

$$\tilde{\mathbf{k}}_{\tau_k}[a] \;=\; \begin{cases} \delta_k[a] & a \in \text{dom}(\delta_k), \\ \tilde{\mathbf{k}}_{\tau_{k-1}}[a] & \text{otherwise}, \end{cases} \tag{6}$$

extends each partial assignment to a fully specified discrete state, encoding the cinematographic convention that a director specifies only what *changes* at each transition. A per-axis lookup $\varphi_a : \mathcal{V}_a \hookrightarrow \mathcal{K}_a$ embeds tokens into continuous values, and axis-aware spline interpolation (Appendix A.2) densifies the keyframe sequence into a per-frame trajectory in the underlying continuous space.

**Two instantiations.** We instantiate this DSL machinery twice in Auteur:

- **Camera DSL** ($\mathcal{P}_{\text{cam}}$) over axes $\mathcal{A}_{\text{cam}}$ defined above. After carry-forward decoding, embedding via $\varphi$, and interpolation, we obtain a dense framing state $\mathbf{k}_t \in \mathcal{K}_{\text{cam}}$ at every frame, which the geometric decoder $\Phi$ (Appendix A.2) maps to a 6-DoF pose $(\mathbf{R}_t, \mathbf{t}_t) \in \text{SE}(3)$ conditioned on the actor state $a_t$.
- **Actor DSL** ($\mathcal{P}_{\text{act}}$) over axes $\mathcal{A}_{\text{act}} = \{\text{P}_x, \text{P}_y, \text{P}_z, \Psi\}$ encoding the pelvis position $\mathbf{p}_t$ and yaw $\psi_t$ from Eq. (1) at sparse keyframes. Body height $h_t$ is treated as a clip-level constant outside the DSL. The same carry-forward and interpolation pipeline produces the dense per-frame actor state $a_{1:T}$ that conditions $\Phi$.

**Three representational layers.** Auteur factors generation through three layers. The director LLM emits discrete DSL tokens in $\mathcal{V}$. The per-axis lookup $\varphi$ embeds these tokens into the continuous space $\mathcal{K}$, where axis-aware spline interpolation produces dense per-frame trajectories. For the camera branch, the geometric decoder $\Phi(\,\cdot\,;\, a_t)$ then maps each framing state to a world-space pose in $\text{SE}(3)$, conditioned on the actor state. Both $\varphi$ and $\Phi$ are deterministic and closed-form, so the only learned component in the pipeline is the LLM-based director.

### 3.2 LLM-Based Director

**Two-stage autoregressive director.** We fine-tune Qwen-2.5-VL (Qwen Team, 2025) as a policy $\pi_\phi$ that, given a free-form text description $\ell$, sequentially generates the two DSL programs introduced before: the actor program $\mathcal{P}_{\text{act}}$ and the camera program $\mathcal{P}_{\text{cam}}$. Tokens within each program are emitted under aa grammar-constrained decoding scheme that guarantees syntactically valid DSL output (Appendix A.1). The actor-first ordering reflects the cinematographic principle that subject motion defines the dramatic context to which the camera responds; we discuss this choice below.

**Factorization.** Recall that each program is a sparse sequence of shot records $(\tau_k, \delta_k)$, with $\tau_k$ a keyframe index and $\delta_k$ a partial assignment over the program's axis set. The joint distribution over both programs given the prompt factors as

$$p_\phi(\mathcal{P}_{\text{act}}, \mathcal{P}_{\text{cam}} \mid \ell) \;=\; \underbrace{\prod_{k=1}^{K_{\text{a}}} p_\phi\Big( (\tau_k^{\text{a}}, \delta_k^{\text{a}}) \;\Big|\; \mathcal{P}_{\text{act}}^{<k},\, \ell \Big)}_{\text{actor planning}} \cdot \underbrace{\prod_{k=1}^{K_{\text{c}}} p_\phi\Big( (\tau_k^{\text{c}}, \delta_k^{\text{c}}) \;\Big|\; \mathcal{P}_{\text{cam}}^{<k},\, \mathcal{P}_{\text{act}},\, \ell \Big)}_{\text{camera planning}}, \tag{7}$$

where $\mathcal{P}_\bullet^{<k}$ denotes the prefix of records strictly before index $k$ in program $\mathcal{P}_\bullet$. Each record-level conditional further decomposes into a token-level autoregression over the DSL vocabulary $\mathcal{V}_\bullet$. This actor-first factorization encodes the causal structure of cinematography; the actor's coarse trajectory defines the scene's dramatic geometry, and the camera is planned as a deliberate response to it. We validate this causally grounded design in the A.3, confirming it maintains predictive accuracy while remaining strictly necessary for decoding the actor-relative representation.

**Training objective.** Given supervised tuples $(\ell, \mathcal{P}_{\text{act}}, \mathcal{P}_{\text{cam}}) \sim \mathcal{D}$ from the dataset described in Section 3.4, we minimize the standard token-level negative log-likelihood induced by Eq. (7),

$$\mathcal{L}(\phi) \;=\; -\,\mathbb{E}_{\mathcal{D}}\Big[\; \log p_\phi(\mathcal{P}_{\text{act}} \mid \ell) \;+\; \log p_\phi(\mathcal{P}_{\text{cam}} \mid \mathcal{P}_{\text{act}}, \ell) \Big]. \tag{8}$$

Full training schedule and settings are reported in Appendix A.5.

**Editability and human-in-the-loop control.** Because $\mathcal{P}_{\text{cam}}$ is a symbolic program, users can directly inspect and modify the generated shot plan before downstream video synthesis. For example, overriding a single keyframe from MS to CU changes the intended shot scale while leaving every other axis and every other keyframe untouched. This provides interpretable control that is difficult to obtain from models that regress trajectories directly into $\mathrm{SE}(3)$.

### 3.3 Interface to Video Generators

The output of Auteur is a dense actor trajectory $\{a_t\}_{t=1}^T$ and a dense 6-DoF camera trajectory $\{(\mathbf{R}_t, \mathbf{t}_t)\}_{t=1}^T$ that is agnostic to the downstream generator. To demonstrate this generality, we integrate Auteur with two architectures spanning distinct conditioning modalities (Figure 2).

(i) **Image-to-video via VerseCrafter.** VerseCrafter (Zheng et al., 2026) is a video diffusion transformer that accepts explicit camera signals. We encode the predicted extrinsics as Plücker embeddings (He et al., 2025) and inject them into the spatial-attention layers. In addition, we extract a 3D Gaussian scene representation from the input image and warp it according to both the actor trajectory $\mathbf{a}_{1:T}$ and the camera trajectory, providing an explicit geometric scaffold for multi-view consistency. (ii) **Text-to-video via Kimodo + VACE** Given the coarse actor trajectory produced by Auteur, Kimodo (Rempe et al., 2026) augments it with full articulated 3D human motion. We then render the resulting body mesh and we use VACE (Jiang et al., 2025) by conditioning on it through a rendered control video: at each frame, we project the actor under the predicted camera and composite it onto a blank canvas. This visual control signal implicitly encodes framing and perspective without requiring any architectural modification to the generator.

Across these architectures, human-centric trajectories produced by Auteur impose actor-aware framing constraints throughout the generated clip, directly addressing the subject–frame drift that afflicts purely geometric baselines (Section 4).

### 3.4 Dataset

Training Auteur requires aligned tuples $(\ell, \mathcal{P}_{\mathrm{act}}, \mathcal{P}_{\mathrm{cam}})$ of a natural-language caption, an actor DSL program, and a camera DSL program. We construct two complementary splits and combine them for training; full pipeline details and statistics are deferred to Appendix A.6.

**Procedural data.** We synthesize $N_{\mathrm{proc}}$ tuples by pairing randomized SOMA (Saito et al., 2026) motion sequences with camera programs sampled from our DSL grammar, decoded into ground-truth trajectories via $\Phi$ (Appendix A.2). Captions are produced from structured templates that verbalize the sampled motion and framing parameters. This split affords full distributional control and near-uniform coverage of all DSL axes, including rare multi-axis transitions.

**Real-world data.** We mine $N_{\mathrm{real}}$ tuples from CondensedMovies (Bain et al., 2020) by recovering metric-scale 3D human and camera trajectories with a TRAM-style estimator (Wang et al., 2024a), projecting them into our human-centric parameter space (Section 3.1), tagging scene-defining moments with the nearest DSL tokens to obtain $\mathcal{P}_{\mathrm{cam}}$, and captioning each clip with AuroraCap (Chai et al., 2024) (augmented by prompt rewriting (Gemini Team, 2024)).

**Why both.** The two splits are complementary: procedural data provides exact ground truth and broad axis coverage, while real-world data captures natural correlations, noise, and stylistic diversity of professional cinematography. As shown in Section 4, combining both is critical for generalization to in-the-wild captions; the synth-only ablation degrades sharply on paraphrased real-world prompts.

## 4 Experiments

We evaluate Auteur on both a controlled synthetic benchmark and existing camera-conditioned video datasets. Our evaluation focuses on two aspects: (i) adherence to cinematographic intent specified in text prompts, and (ii) robustness of camera–subject alignment during motion.

**Framing metrics.** Standard metrics such as *out-of-frame rate (out-rate)* and *Fréchet distance* on absolute camera do not capture whether a model correctly executes cinematographic instructions. The *out-rate* merely confirms whether a subject is visible, completely ignoring spatial composition; while Fréchet distance evaluates generic distribution similarity without capturing whether a model actually executed a specific stylistic command, such as a precise orientation or a low-angle close-up. We therefore introduce a set of framing-specific metrics that directly measure adherence to intended shot properties. Given a generated trajectory, we compute: (i) *F-Ori*: alignment between the camera orientation and desired actor-facing camera direction; (ii) *F-RoT*: adherence to the rule-of-thirds composition; (iii) *F-Scale*: conformity with the target shot scale; (iv) *F-Tilt*: accuracy of camera elevation angle; (v) *F-Roll*: accuracy of the Dutch angle.

Each metric is normalized to $[0, 1]$, where 1 indicates perfect adherence to the target framing. To evaluate not just static composition but the accurate execution of dynamic camera choreography over time, we compute the average of these five metrics for all the frames of the generated sequence. The mean yields the final *Auteur Score*, an aggregated metric where a score of 1 represents perfect intended cinematographic choreography.

**Controlled diagnostic benchmark.** Evaluating camera control models on in-the-wild datasets can mask their true capabilities due to strong dataset biases, such as the prevalence of centered, eye-level shots. To isolate and rigorously evaluate adherence to cinematographic instructions, we construct a *Controlled Diagnostic Benchmark*. The benchmark is designed to evaluate each framing attribute independently. For each metric (e.g., shot scale, orientation, or composition), we generate a set of test sequences where the target attribute is sampled from a uniform distribution over its valid range. This ensures coverage of diverse and challenging configurations, preventing models from exploiting dataset priors. The remaining camera parameters are randomly sampled to maintain variability and avoid trivial solutions. Each test sequence consists of a prompt specifying the desired camera behavior, along with a corresponding ground-truth framing configuration. This allows direct measurement of how well a model follows explicit cinematographic instructions.

Table 1: **Framing accuracy on the controlled benchmark.** Mean per-frame framing scores are reported. Rows 5-6 represent an ablation isolating the actor-relative representation by reconstructing ground-truth (GT) trajectories using a world-space DSL.

| | F-Ori | F-RoT | F-Scale | F-Tilt | F-Roll | CamF1 | Auteur-Score | Out-Rate |
|---|---|---|---|---|---|---|---|---|
| Baseline | 0.495 | 0.625 | 0.695 | 0.875 | 0.880 | 0 | - | 0 |
| LAMP | 0.451 | 0.257 | 0.729 | 0.832 | 0.745 | 0.112 | 0.603 | 41.0 |
| PulpMotion | <u>0.514</u> | 0.351 | 0.712 | <u>0.891</u> | <u>0.839</u> | 0.091 | <u>0.661</u> | 36.2 |
| Ours | **0.962** | **0.745** | **0.888** | **0.968** | **0.937** | **0.623** | **0.901** | **5.45** |
| World-space, 4-seg (GT recon.) | 0.499 | 0.371 | 0.730 | 0.675 | 0.311 | 0.262 | - | 0 |
| World-space, 8-seg (GT recon.) | 0.500 | <u>0.372</u> | <u>0.733</u> | 0.674 | 0.313 | <u>0.280</u> | - | 0 |

**Competing methods.** (i) **LAMP** (Kizil et al., 2025) models camera motion via structured 3D trajectory programs, providing strong geometric control but limited framing awareness. (ii) **PulpMotion** (Courant et al., 2026), in contrast, introduces a learned screen-space framing signal that improves subject coherence but lacks explicit, controllable cinematographic structure. All methods are evaluated under identical prompts and trajectory settings to ensure fair comparison. We train and test our method on a single NVIDIA A100 GPU. Training takes about 8 hours. (iii) Additionally, to contextualize performance on this benchmark, we include a naive static **Baseline** consisting of a stationary camera with a centered composition observing a non-moving subject. This Baseline approximates the dominant mode of the data distribution and therefore achieves moderate scores on several framing metrics without actively following the prompt. Importantly, this baseline highlights a key challenge in dynamic camera control: while dynamic methods aim to execute complex motion, they may introduce framing instability during movement, leading to subject drift or out-of-frame artifacts. In contrast, the static baseline trivially maintains full subject visibility, resulting in a 0% out-rate. We include this baseline as a reference point for interpreting the trade-off between motion expressiveness and framing stability.

**Main results.** Table 1 reports the quantitative results on the controlled benchmark. Our method significantly outperforms prior approaches across all framing metrics, demonstrating that modeling camera behavior in a human-centric framing space leads to more accurate execution of cinematographic intent. In particular, the large improvement in F-Ori and F-RoT indicates better alignment between camera orientation and subject composition. Our method also reduces out-of-frame errors (Out-Rate) by a large margin, showing improved stability during motion.

**Isolating the representation.** To isolate the benefit of our actor-relative formulation independently of the model architecture, we compare it against a world-space domain-specific language (DSL) baseline. Because prior world-space DSLs are not directly invertible, we evaluate this by fitting our ground-truth trajectories with a free-form world-space segment DSL. Even when granted significant advantages—specifically, reconstructing from ground truth rather than predicting from language, and receiving perfect first-frame alignment—the world-space representation struggles. Increasing its segment budget from 4 to 8 yields only a 0.280 CamF1 score, compared to 0.623 for our representation. The largest performance gaps occur on the framing axes, demonstrating that world-space segments cannot inherently anchor the camera to a moving actor and are not suitable for sophisticated camera movements.

**Generalization on existing benchmarks.** We further evaluate Auteur on the PulpMotion benchmark under *pure* and *mixed* settings (see PulpMotion for details). Results are shown in Table 2. On the *pure* split, our method achieves the highest Camera CLaTr score (a contrastive language-trajectory alignment metric introduced by (Courant et al., 2024)) of 71.1, while reducing the out-of-frame rate to 1.26%, compared to 7.00% for LAMP and over 18% for PulpMotion variants. This indicates that modeling camera behavior in a human-centric framing space leads to substantially improved subject alignment and stability during motion. While LAMP attains the highest Camera F1 score (a multi-class F1 metric for camera movement translation), this reflects its strength in reproducing geometric trajectories rather than maintaining consistent subject-centric framing, as evidenced by its higher out-of-frame rate.

On the *mixed* split, which introduces more diverse and challenging scenarios, our method maintains an almost zero out-of-frame rate (0.01%), demonstrating strong robustness across data distributions.

Table 2: **PulpMotion benchmark (pure and mixed).** Best/second-best results are in bold/underlined.

| | *Pure* | | | | *Mixed* | | | |
|---|---|---|---|---|---|---|---|---|
| | Cam F1 | CLaTr | Out-rate | $FD_{frame}$ | Cam F1 | CLaTr | Out-rate | $FD_{frame}$ |
| LAMP | **99.7** | <u>69.5</u> | <u>7.00</u> | 35.5 | **71.3** | **59.2** | <u>7.02</u> | 33.57 |
| PulpMotion (DiT) | 52.9 | 39.7 | 18.5 | **4.19** | 34.8 | 31.0 | 24.9 | **5.36** |
| PulpMotion (MAR) | 64.8 | 49.3 | 29.1 | <u>6.80</u> | 38.7 | 39.5 | 39.5 | 8.63 |
| Ours | <u>83.6</u> | **71.1** | **1.26** | 8.08 | <u>54.0</u> | <u>48.5</u> | **0.01** | <u>8.03</u> |

In contrast, both PulpMotion variants exhibit significant degradation in out-rate, exceeding 24%, suggesting instability in maintaining subject visibility under more complex conditions. Although LAMP achieves higher F1 and CLaTr scores, its performance remains limited by its reliance on world-space trajectory modeling, which does not explicitly enforce subject-centric framing constraints.

These results highlight a key distinction between approaches. LAMP models camera motion as a geometric planning problem in world coordinates, which leads to accurate trajectories but weaker control over subject framing. PulpMotion introduces an implicit framing signal through learned latents, improving perceptual quality but lacking explicit controllability. In contrast, Auteur explicitly models camera behavior as a function of the subject, enabling more reliable framing and significantly improved robustness in maintaining subject visibility. Overall, the results demonstrate that a framing-centric formulation provides a more effective interface for controllable video generation, particularly in scenarios where maintaining consistent subject composition is critical. See supplemental webpage.

**Ablation studies.** We evaluate the effect of combining real and synthetic training data. While synthetic data offers perfect ground truth, its captions are linguistically rigid. To simulate in-the-wild queries, we test on a challenging split with LLM-augmented and paraphrased descriptions. Although this augmentation naturally introduces some linguistic ambiguity, it accurately reflects the complexity of real user intent. Table 3 compares the full model with a synthetic-only baseline. The 'Translation' metric denotes strict accuracy across all three spatial directions $(t_x, t_y, t_z)$.

Results show that real-world data is critical for grounding diverse narrative intent, preventing severe performance drops in complex parameters like Shot Scale (SS), and making Auteur robust to natural language. Importantly, training with real augmented captions substantially improves performance on real-world paraphrased prompts while introducing only minimal degradation on the synthetic test set, indicating strong generalization across both domains. Both models achieve zero format errors.

**Integration across generation paradigms.** To demonstrate the generality of Auteur, we integrate it with multiple video generation frameworks spanning different conditioning modalities:

(i) **Image-to-Video (I2V).** We interface with VerseCrafter (Zheng et al., 2026), which accepts explicit camera parameters. The predicted 6-DoF camera trajectories from Auteur are directly used to guide the generation process, enabling controlled viewpoint changes consistent with the actor's motion.

(ii) **Joint human motion and camera generation.** We integrate with Kimodo (Rempe et al., 2026), which refines coarse motion trajectories into articulated human motion. By combining Kimodo with our predicted camera trajectories, we enable synchronized generation of human motion and camera behavior, allowing rendering from arbitrary, dynamically controlled viewpoints.

(iii) **Text-to-Video (T2V).** For models without explicit camera interfaces, such as VACE (Jiang et al., 2025), we convert predicted trajectories into rendered control sequences (e.g., projected actor geometry under the predicted camera). These sequences serve as conditioning signals, enabling implicit control over framing and perspective without architectural modification.

**Cross-renderer evaluation.** To quantitatively evaluate the efficacy of our trajectories when transferred to downstream generators, we measure prompt-adherence using VQAScore (the $p(\text{yes})$ formulation introduced in CHAI (Lin et al., 2026b)), which acts as an automatic stand-in for human judgment on whether the rendered video follows the requested camera and framing instructions. Crucially, we compare against a strong baseline—Wan2.2 direct TI2V (Wan et al., 2025) 5B model using the same prompt but without explicit trajectory guidance—isolating the contribution of Auteur's trajectory itself.

Table 3: **Ablation on Training Data Composition.** We evaluate prediction accuracy on both real and synthetic test sets. Training on real data prevents severe performance drops on complex parameters (e.g., Shot Scale) and robustifies the model to linguistically augmented and paraphrased captions.

| | **Real Test Set** | | **Synthetic Test Set** | |
|---|---|---|---|---|
| **DSL Component** | **Synth-Only** | **Ours** | **Synth-Only** | **Ours** |
| Translation | 96.4% | **97.8%** | 99.5% | **99.8%** |
| Human Yaw ($\Psi$) | 93.7% | **97.5%** | **98.8%** | 98.5% |
| Orientation Angle (OA) | 84.6% | **85.1%** | 98.6% | 98.6% |
| Shot Scale (SS) | 60.1% | **85.8%** | 95.0% | **95.2%** |
| Camera Level (CL) | 92.8% | **93.2%** | **97.1%** | 96.7% |
| Look-at Level (LL) | 90.2% | **90.5%** | **96.8%** | 96.3% |
| Framing Offset (FO) | 86.1% | **88.6%** | **99.1%** | 97.9% |
| Dutch Angle (DA) | 98.9% | **99.3%** | **100.0%** | 99.8% |

Table 4: **Cross-Renderer Prompt Adherence.** VQAScore ($p(\text{yes})$) evaluating how faithfully the rendered video executes the requested camera/framing instruction.

| **Configuration** | **VQAScore** ($p(\text{yes})$) |
|---|---|
| Baseline (Wan2.2) | 0.39 |
| Ours + VerseCrafter | **0.50** |
| Ours + ReCamMaster | <u>0.49</u> |
| Ours + EPiC | 0.47 |

As shown in Table 4, every generator driven by Auteur's trajectory improves prompt-adherence substantially over the Wan2.2 baseline (achieving $0.47$–$0.50$ vs. $0.39$). This directly demonstrates two key properties: (i) independent renderers beyond VerseCrafter—including EPiC (Wang et al., 2025b) and ReCamMaster(Bai et al., 2025)—faithfully execute Auteur's predicted trajectories while preserving output quality, and (ii) the framing improvements are not an artifact of a single renderer, as prompt-adherence consistently rises over the unguided baseline across all architectures. We provide side-by-side video comparisons of these integrations on the supplementary website.

**User study.** We conducted a blind A/B test to validate perceptual effectiveness. Across 10 scenarios, paired videos used identical prompts, images, and rendering settings—differing solely in the predicted camera trajectory. From 225 pairwise judgments (24 participants), users chose which video looked more professional (quality) and better followed instructions (adherence). Auteur achieved overwhelming win rates against prior works: **82.7%** (quality) and **80.4%** (adherence) against LAMP (Kizil et al., 2025); and **81.3%** (quality) and **83.1%** (adherence) against PulpMotion (Courant et al., 2026). This confirms our trajectory-level gains yield perceptually superior videos. We provide an additional automatic perceptual evaluation in the (Section A.4) which further supports findings.

## 5 Conclusion

We presented Auteur, a language-driven framework for cinematographic camera framing in human-centric video generation. Auteur grounds camera control in actor-relative framing rather than world-space trajectories, and expresses it through a deterministically decodable DSL generated by a fine-tuned LLM. By formulating camera behavior as a conditional response to human motion, our approach enables more consistent subject alignment, interpretable control, and improved robustness across diverse scenarios. Extensive experiments demonstrate that this framing-centric formulation leads to significant gains in both controllability and stability compared to prior trajectory-based methods. Future work includes extending the framework to multi-actor interactions, richer interaction dynamics, and integrating learned policies more tightly with video generation architectures.

**Limitations.** Our DSL keyframe representation assumes that cinematographic intent can be expressed as a sparse sequence of discrete states; highly continuous or reactive shot styles (e.g., handheld verité, continuous reframing during action) may be underrepresented in the vocabulary. We only model actor trajectories without fine articulation, augmenting Auteur with advanced human motion synthesis capabilities is an interesting future direction. Finally, while Auteur reliably controls framing, the perceptual quality of the final video remains largely determined by the downstream generator.

**Broader Impact.** Auteur lowers the barrier to professional-quality cinematography for creators without formal filmmaking training, with potential applications in accessibility, independent production, and education. At the same time, the same capability could be misused to produce more convincing synthetic media; we encourage downstream deployments to pair Auteur with provenance and watermarking tools.

## 6 Acknowledgments

This research was supported in part by the KUIS AI Center Research Awards awarded to M. Burak Kizil and Enes Sanli. Aykut Erdem and Erkut Erdem acknowledge support from the TÜBİTAK 2247-A Program under Awards 123C550 and 123C542, respectively. Niloy Mitra acknowledges partial support from the UCL AI Centre.

## 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.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, and Di Zhang. ReCamMaster: Camera-controlled generative rendering from a single video. In *ICCV*, 2025.

Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed movies: Story based retrieval with contextual embeddings. *CoRR*, abs/2005.04208, 2020.

David Bordwell and Kristin Thompson. *Film Art: An Introduction*. McGraw-Hill Education, 12th edition, 2020.

Chenjie Cao, Jingkai Zhou, shikai Li, Jingyun Liang, Chaohui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. Uni3c: Unifying precisely 3d-enhanced camera and human motion controls for video generation. *arXiv preprint arXiv:2504.14899*, 2025.

Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jeng-Neng Hwang, Saining Xie, and Christopher D. Manning. Auroracap: Efficient, performant video detailed captioning and a new benchmark. *arXiv preprint arXiv:2410.03051*, 2024.

Boyuan Cheng, Shang Ni, Jian Jun Zhang, and Xiaosong Yang. Automating visual narratives: Learning cinematic camera perspectives from 3d human interaction. *SSRN Electronic Journal*, 2025.

Soon Yau Cheong, Duygu Ceylan, Armin Mustafa, Andrew Gilbert, and Chun-Hao Paul Huang. Boosting camera motion control for video diffusion transformers. In *36th British Machine Vision Conference 2025, BMVC 2025, Sheffield, UK, November 24-27, 2025*. BMVA, 2025.

Robin Courant, Nicolas Dufour, Xi Wang, Marc Christie, and Vicky Kalogeiton. E.T. the exceptional trajectories: Text-to-camera-trajectory generation with character awareness. In *ECCV*, 2024.

Robin Courant, David Loiseaux, Xi Wang, Marc Christie, and Vicky Kalogeiton. Pulp motion: Framing-aware multimodal camera and human motion generation. 2026.

Wanquan Feng, Tianhao Qi, Jiawei Liu, Mingzhen Sun, Pengqi Tu, Tianxiang Ma, Fei Dai, Songtao Zhao, Siyu Zhou, and Qian He. I2VControl: Disentangled and unified video motion synthesis control. In *ICCV*, 2025.

Quentin Galvane, Christophe Lino, Marc Christie, Julien Fleureau, Fabien Servant, François-Louis Tariolle, and Philippe Guillotel. Directing cinematographic drones. *ACM TOG*, 37(3), 2018.

Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Michael Gleicher and Andrew Witkin. Through-the-lens camera control. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, page 331–340, New York, NY, USA, 1992. Association for Computing Machinery.

Ahmet Berke Gokmen, Yigit Ekin, Bahri Batuhan Bilecen, and Aysegul Dundar. RoPECraft: Training-free motion transfer with trajectory-guided rope optimization on diffusion transformers. In *Adv. Neural Inform. Process. Syst.*, 2025.

Google DeepMind. Veo-3. `https://blog.google/technology/ai/generative-media-models-io-2025/`, 2025. Accessed April 17, 2026.

Hao He, Ceyuan Yang, Shanchuan Lin, Yinghao Xu, Meng Wei, Liangke Gui, Qi Zhao, Gordon Wetzstein, Lu Jiang, and Hongsheng Li. CameraCtrl II: Dynamic scene exploration via camera-controlled video diffusion models. *ArXiv preprint arXiv:2503.10592*, 2025.

Chen Hou and Zhibo Chen. Training-free camera control for video generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models, 2023.

Hongda Jiang, Xi Wang, Marc Christie, Libin Liu, and Baoquan Chen. Cinematographic camera diffusion model. In *EG*, 2024.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. VACE: All-in-one video creation and editing. In *ICCV*, 2025.

Wonjoon Jin, Qi Dai, Chong Luo, Seung-Hwan Baek, and Sunghyun Cho. FloVD: Optical flow meets video diffusion model for enhanced camera-controlled video synthesis. In *CVPR*, 2025.

Steven D. Katz. *Film Directing: Shot by Shot – Visualizing from Concept to Screen*. Michael Wiese Productions, 1991.

Muhammed Burak Kizil, Enes Sanli, Niloy J. Mitra, Erkut Erdem, Aykut Erdem, and Duygu Ceylan. Lamp: Language-assisted motion planning for controllable video generation, 2025.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Mengtian Li, Yuwei Lu, Feifei Li, Chenqi Gan, Zhifeng Xie, and Xi Wang. VERTIGO: Visual preference optimization for cinematic camera trajectory generation, 2026.

Teng Li, Guangcong Zheng, Rui Jiang, Shuigen Zhan, Tao Wu, Yehao Lu, Yining Lin, Chuanyun Deng, Yepan Xiong, Min Chen, Lin Cheng, and Xi Li. RealCam-I2V: Real-world image-to-video generation with interactive complex camera control. In *ICCV*, 2025.

Xinyang Li, Zhangyu Lai, Linning Xu, Yansong Qu, Liujuan Cao, Shengchuan Zhang, Bo Dai, and Rongrong Ji. Director3d: Real-world camera trajectory and 3d scene generation from text. In *NeurIPS*, 2024.

Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. VideoDirectorGPT: Consistent multi-scene video generation via llm-guided planning. In *COLM*, 2024.

Kuan Heng Lin, Zhizheng Liu, Pablo Salamanca, Yash Kant, Ryan Burgert, Koichi Namekata, Yiwei Zhao, Bolei Zhou, Micah Goldblum, Paul Debevec, and Ning Yu. Vista4D: Video reshooting with 4d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a.

Zhiqiu Lin, Chancharik Mitra, Siyuan Cen, Isaac Li, Yuhan Huang, Yu Tong Tiffany Ling, Hewei Wang, Irene Pi, Shihang Zhu, Ryan Rao, George Liu, Jiaxi Li, Ruojin Li, Yili Han, Yilun Du, and Deva Ramanan. Building a precise video language with human-ai oversight, 2026b.

Christophe Lino and Marc Christie. Intuitive and efficient camera control with the toric space. *ACM Trans. Graph.*, 34(4), 2015.

Joseph V Mascelli. *The Five C's of Cinematography: Motion Picture Filming Techniques*. Cine/Graphic Publications, 1965.

Qwen Team. Qwen2.5-vl, 2025.

Gustavo Mercado. *The Filmmaker's Eye: Learning (and Breaking) the Rules of Cinematic Composition*. Focal Press, 2010.

OpenAI. Sora: Generating videos from text, 2025. Accessed October 7, 2025.

Zirui Pan, Xin Wang, Yipeng Zhang, Hong Chen, Kwan Man Cheng, Yaofei Wu, and Wenwu Zhu. Modular-Cam: Modular dynamic camera-view video generation with llm. In *AAAI*, 2025.

Davis Rempe, Mathis Petrovich, Ye Yuan, Haotian Zhang, Xue Bin Peng, Yifeng Jiang, Tingwu Wang, Umar Iqbal, David Minor, Michael de Ruyter, Jiefeng Li, Chen Tessler, Edy Lim, Eugene Jeong, Sam Wu, Ehsan Hassani, Michael Huang, Jin-Bey Yu, Chaeyeon Chung, Lina Song, Olivier Dionne, Jan Kautz, Simon Yuen, and Sanja Fidler. Kimodo: Scaling controllable human motion generation, 2026.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. GEN3C: 3d-informed world-consistent video generation with precise camera control. In *CVPR*, 2025.

Jun Saito, Jiefeng Li, Michael de Ruyter, Miguel Guerrero, Edy Lim, Ehsan Hassani, Roger Blanco Ribera, Hyejin Moon, Magdalena Dadela, Marco Di Lucca, Qiao Wang, Xueting Li, Jan Kautz, Simon Yuen, and Umar Iqbal. Soma: Unifying parametric human body models. *arXiv preprint arXiv:2603.16858*, 2026.

Kunpeng Song, Tingbo Hou, Zecheng He, Haoyu Ma, Jialiang Wang, Animesh Sinha, Sam Tsai, Yaqiao Luo, Xiaoliang Dai, Li Chen, Xide Xia, Peizhao Zhang, Peter Vajda, Ahmed Elgammal, and Felix Juefei-Xu. Llama learns to direct: Directorllm for human-centric video generation, 2025.

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.

Qinghe Wang, Yawen Luo, Xiaoyu Shi, Xu Jia, Huchuan Lu, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. CineMaster: A 3d-aware and controllable framework for cinematic text-to-video generation. In *SIGGRAPH*, 2025a.

Yufu Wang, Ziyun Wang, Lingjie Liu, and Kostas Daniilidis. Tram: Global trajectory and motion of 3d humans from in-the-wild videos. In *ECCV*, page 467–487, 2024a.

Zhenzhi Wang, Yixuan Li, Yanhong Zeng, Youqing Fang, Yuwei Guo, Wenran Liu, Jing Tan, Kai Chen, Tianfan Xue, Bo Dai, and Dahua Lin. Humanvid: Demystifying training data for camera-controllable human image animation. In *NeurIPS*, 2024b.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, 2024c.

Zun Wang, Jaemin Cho, Jialu Li, Han Lin, Jaehong Yoon, Yue Zhang, and Mohit Bansal. EPiC: Efficient video camera control learning with precise anchor-video guidance. *ArXiv preprint arXiv:2505.21876*, 2025b.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. TrajectoryCrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *ICCV*, 2025.

Mengchen Zhang, Tong Wu, Jing Tan, Ziwei Liu, Gordon Wetzstein, and Dahua Lin. GenDoP: Auto-regressive camera trajectory generation as a director of photography. In *ICCV*, 2025.

Sixiao Zheng, Zimian Peng, Yanpeng Zhou, Yi Zhu, Hang Xu, Xiangru Huang, and Yanwei Fu. VidCRAFT3: Camera, object, and lighting control for image-to-video generation. *ArXiv preprint arXiv:2502.07531*, 2025.

Sixiao Zheng, Minghao Yin, Wenbo Hu, Xiaoyu Li, Ying Shan, and Yanwei Fu. Versecrafter: Dynamic realistic video world model with 4d geometric control. *arXiv preprint arXiv:2601.05138*, 2026.

# A Appendix

## A.1 Cinematography DSL

We define a discrete DSL as a quantized, human-readable version of the human centric camera parameter space. As shown in Table A.1, each axis is equipped with a finite vocabulary of cinematographically motivated tokens.

Table A.1: **Camera axes: continuous domains, discrete DSL vocabularies, and token-to-scalar mappings.** Each axis $a \in \mathcal{A}$ has a continuous domain $\mathcal{K}_a$, a finite vocabulary $\mathcal{V}_a$, and a deterministic embedding $\iota_a : \mathcal{V}_a \hookrightarrow \mathcal{K}_a$. Scalar values are expressed in meters (CL, LL), unitless fractions (SS, FO), or degrees (OA, DA). Shot-scale scalars denote the fraction of frame height occupied by the actor's body height $h_t$.

| Axis | Symbol | $\mathcal{K}_a$ | Geometry | $\mathcal{V}_a$ (token $\to$ scalar via $\iota_a$) |
|---|---|---|---|---|
| Camera Level | CL | $\mathbb{R}_{\geq 0}$ | $\mathbb{R}$ | GROUND $\to 0.1\,\text{m}$, LOW $\to 0.6\,\text{m}$, EYE $\to 1.6\,\text{m}$, HIGH $\to 2.5\,\text{m}$, OVERHEAD $\to 4.0\,\text{m}$ |
| Look-at Level | LL | $[0,\, h_t]$ | $\mathbb{R}$ | FEET $\to 0.0$, KNEES $\to 0.25\,h_t$, WAIST $\to 0.50\,h_t$, CHEST $\to 0.75\,h_t$, EYES $\to 0.95\,h_t$ |
| Shot Scale | SS | $(0,\, 1]$ | $\mathbb{R}$ | ECU $\to 0.10$, CU $\to 0.20$, MCU $\to 0.35$, MS $\to 0.50$, MLS $\to 0.65$, LS $\to 0.85$, EWS $\to 1.00$ |
| Framing Offset | FO | $[-\frac{1}{2},\, +\frac{1}{2}]$ | $\mathbb{R}$ | FARLEFT $\to -0.35$, LEFT $\to -0.17$, CENTER $\to 0.0$, RIGHT $\to +0.17$, FARRIGHT $\to +0.35$ |
| Orientation Angle | OA | $\mathbb{S}^1$ | $\mathbb{S}^1$ | FRONT $\to 0$, FRONTL $\to -45$, SIDEL $\to -90$, BACKL $\to -135$, BACK $\to \pm 180$, BACKR $\to +135$, SIDER $\to +90$, FRONTR $\to +45$ |
| Dutch Angle | DA | $\mathbb{S}^1$ | $\mathbb{S}^1$ | NONE $\to 0$, SLIGHTL $\to -10$, SLIGHTR $\to +10$, STRONGL $\to -25$, STRONGR $\to +25$ |

By changing the camera axes properties at different keyframes, our DSL captures common camera behaviour. For example, the orientation determines whether a particular tracking motion is *tail, lead,* or *side*. A change in shot scale results in a *dolly in* or *out*. The camera and look-at levels together determine tilt behaviour. A change in roll results in the classic *duth angle* effect.

## A.2 From DSL to 6-DoF Camera Trajectories

We now describe how a sparse keyframe sequence in $\mathcal{K}$ becomes a dense $\mathrm{SE}(3)$ trajectory.

**Geometric decoder $\Phi$.** Given a dense state $\mathbf{k}_t \in \mathcal{K}$, an actor state $a_t$, and fixed intrinsics Field-of-View and Aspect Ratio $(\mathrm{FOV}, \mathrm{AR})$, the decoder $\Phi(\mathbf{k}_t; a_t, \mathrm{FOV}, \mathrm{AR}) \in \mathrm{SE}(3)$ is defined as follows. The image-plane span occupied by the actor's body of height $h_t$ at shot scale SS is

$$H \;=\; h_t/\mathrm{SS}, \qquad W \;=\; \mathrm{AR} \cdot H, \tag{9}$$

giving an actor-camera distance

$$d \;=\; \frac{H}{2\tan(\mathrm{FOV}/2)}, \tag{10}$$

and a lateral framing shift $S_f = \mathrm{FO} \cdot W$ in the world. The global azimuth in the ground plane is $\theta = \mathrm{OA} + \psi_t$. The camera position $\mathbf{T}_t = (T_t^x, T_t^y, T_t^z) \in \mathbb{R}^3$ is

$$T_t^x = \mathbf{p}_t^x \;+\; d\cos\theta \;-\; S_f \sin\theta, \tag{11}$$

$$T_t^y = \mathbf{p}_t^y \;+\; d\sin\theta \;+\; S_f \cos\theta, \tag{12}$$

$$T_t^z = \mathrm{CL}. \tag{13}$$

The orientation $\mathbf{R}_t \in \mathrm{SO}(3)$ is determined by the convention that the optical axis points from $\mathbf{T}_t$ toward the look-at target $\mathbf{q}_t = (\mathbf{p}_t^x, \mathbf{p}_t^y, \mathrm{LL})$, modulo a Dutch-angle roll:

$$\mathbf{R}_t \;=\; \mathrm{LookAt}\big(\mathbf{T}_t \to \mathbf{q}_t,\, \hat{\mathbf{z}}\big) \cdot \mathbf{R}_z(\mathrm{DA}), \tag{14}$$

where $\mathrm{LookAt}$ produces the rotation aligning the camera's $-\hat{\mathbf{z}}_{\text{cam}}$ with $\mathbf{q}_t - \mathbf{T}_t$ and using $\hat{\mathbf{z}}$ as up reference. Equivalently, in Euler-angle form (yaw–pitch–roll, intrinsic),

$$\mathrm{Yaw} = \theta + \pi, \qquad \mathrm{Pitch} = \mathrm{atan2}(\mathrm{LL} - \mathrm{CL},\; d), \qquad \mathrm{Roll} = \mathrm{DA}. \tag{15}$$

**Domain restrictions.** $\Phi$ is well-defined on the open subset $\mathcal{K}^\circ \subset \mathcal{K}$ where $\mathrm{SS} \in (\epsilon, 1]$ for some $\epsilon > 0$ (bounding $d$ from above) and where $(\mathrm{LL}, \mathrm{CL}, d)$ are not simultaneously zero (avoiding pitch degeneracy). Both DSL vocabularies and dataset construction enforce $\mathcal{K}^\circ$.

**Axis-aware interpolation in $\mathcal{K}$.** Note that we deliberately interpolate in $\mathcal{K}$ rather than $\mathrm{SE}(3)$: cinematographically meaningful axes vary smoothly under their natural metrics (e.g., a linear ramp in SS produces a perceptually uniform zoom), whereas $\mathrm{SE}(3)$ interpolation entangles rotation and translation in ways that violate framing intent.

### A.3 Ablation on Actor-First Factorization

As formulated in the main text, Auteur predicts the actor's motion before the camera's response. This actor-first factorization is a deliberate design choice driven by causality and utility rather than pure predictive accuracy.

As shown in Table A.2, alternative autoregressive orderings—such as generating the camera first or generating both jointly—perform comparably across all axes on the synthetic test set, with no single formulation emerging as a strict winner.

We adopt the actor-first formulation for two fundamental reasons. First, the ordering is strictly dictated by the actor-relative pipeline: because the camera is parameterized relative to the actor, generating camera tokens first is inherently ill-posed. Without an established actor trajectory, there is nothing to anchor the camera to, meaning a standalone camera program cannot be decoded into a valid world-space trajectory. Second, this factorization enables practical downstream control. It allows the camera generation module to be conditioned on an externally provided or user-defined actor sequence, cleanly separating subject motion from camera response in a way that matches real-world cinematography. The empirical results in Table A.2 confirm that we incur no performance penalty by adopting this necessary, causally grounded factorization.

Table A.2: **Effect of Factorization Ordering.** Prediction accuracy (%) on the synthetic test set across different autoregressive generation orders. The required actor-first factorization performs on par with alternative orderings.

| DSL Component | Camera-First | Joint | Ours (Actor-first) |
|---|---|---|---|
| Translation | 99.6 | 99.6 | **99.8** |
| Human Yaw ($\Psi$) | **99.3** | **99.3** | 98.5 |
| Orientation Angle (OA) | **98.7** | 98.6 | 98.6 |
| Shot Scale (SS) | 94.1 | 93.0 | **95.2** |
| Camera Level (CL) | **98.7** | 98.2 | 96.7 |
| Look-at Level (LL) | **98.5** | 98.2 | 96.3 |
| Framing Offset (FO) | 93.3 | 93.0 | **97.9** |
| Dutch Angle (DA) | 96.8 | 96.7 | **99.8** |

### A.4 Automatic Perceptual Evaluation

To complement the human user study with an automatic video-level measure, we evaluated the generated videos using VBench (Huang et al., 2023). The evaluation was conducted on the same set of scenarios used in our human trials, ensuring the downstream generator and rendering settings remained strictly identical across all methods.

As reported in Table A.3, Auteur leads across most evaluated dimensions. Notably, our method shows strong improvements in Subject Consistency, Dynamic Degree, and overall Aesthetic Quality. These

quantitative video-level results align closely with the user study, confirming that the framing and trajectory improvements of Auteur directly yield higher-quality final video outputs.

Table A.3: **Automatic Perceptual Evaluation (VBench).** Video-level quality metrics (%) for generated videos. Auteur achieves the highest scores across the majority of perceptual dimensions.

| Metric | PulpMotion | LAMP | Ours |
|---|---|---|---|
| Subject Consistency | 87.72 | 89.50 | **91.23** |
| Dynamic Degree | 81.82 | 54.55 | **90.91** |
| Background Consistency | 90.86 | 91.37 | **93.01** |
| Motion Smoothness | 98.32 | **99.08** | 98.99 |
| Aesthetic Quality | 59.88 | 59.35 | **63.11** |
| Image Quality | 69.36 | **69.65** | 68.71 |

### A.5 Training details

We fine-tune the `Qwen2.5-VL-7B-Instruct` model using parameter-efficient LoRA adaptation with rank 128 and $\alpha = 128$. Training is performed for a single epoch over approximately 300K image-text samples. We use a global batch size of 512 with gradient accumulation, BF16 mixed-precision training, cosine learning rate scheduling, and a peak learning rate of $1 \times 10^{-4}$. Following prior efficient multimodal fine-tuning practices, the vision encoder and LLM backbone are frozen while only the multimodal merger layers and LoRA adapters are optimized.

### A.6 Dataset

This appendix expands on the brief description in Section 3.4. We provide the full procedural-synthesis pipeline, the four-stage real-world annotation pipeline, dataset statistics, and a comparison to existing camera-trajectory datasets.

#### A.6.1 Procedural Synthesis

The procedural split consists of $N_{\text{proc}}$ tuples $(\ell, \mathcal{P}_{\text{act}}, \mathcal{P}_{\text{cam}})$ synthesized in three stages.

**Motion sampling.** We sample human motion clips from the SOMA library (Saito et al., 2026), covering common locomotion (walking, running, turning), gestural actions (pointing, reaching, sitting), and stationary poses. Each clip is a sequence $\{a_t\}_{t=1}^{T}$ of pelvis position $\mathbf{p}_t$, yaw $\psi_t$, and a clip-level body height $h_t$ drawn from a realistic human-height distribution.

**Camera program sampling.** For each motion clip we independently sample a camera DSL program $\mathcal{P}_{\text{cam}}$ from our grammar with axis-stratified priors. Concretely, we (i) sample the number of keyframes $K \sim \text{Uniform}\{1, \ldots, K_{\max}\}$, (ii) sample keyframe indices $\tau_1 < \cdots < \tau_K$ uniformly within $[1, T]$, and (iii) for each $\tau_k$ sample a partial assignment $\delta_k$ by first drawing the cardinality $|\text{dom}(\delta_k)|$ from a distribution that explicitly upweights multi-axis transitions, then drawing token values uniformly from each axis vocabulary. This sampling strategy ensures the LLM sees sufficient examples of rare combinations such as simultaneous five- or six-axis changes, which are vanishingly rare in real-world footage (Table A.4, left).

**Decoding and caption generation.** Each $(\mathcal{P}_{\text{act}}, \mathcal{P}_{\text{cam}})$ pair is decoded via the deterministic pipeline of Section 3.1 and Appendix A.2: carry-forward decoding, embedding via $\varphi$, axis-aware spline interpolation, and the geometric decoder $\Phi$ produce a ground-truth dense actor state $\mathbf{a}_{1:T}$ and 6-DoF camera trajectory $\{(\mathbf{R}_t, \mathbf{t}_t)\}_{t=1}^{T}$. Captions are generated by structured templates that verbalize the sampled motion and framing parameters: e.g., a clip with SS=MS, OA=FRONT, CL=EYE over a forward-walking actor yields the caption *"A person walks forward; medium shot, frontal, eye-level."* While these captions are linguistically rigid, they provide unambiguous supervision for the LLM to associate cinematic vocabulary with DSL outputs; we ablate the importance of augmenting this with real-world captions in Section 4.

### A.6.2 Real-World Pipeline

The real-world split contains $N_{\text{real}}$ tuples mined from CondensedMovies (Bain et al., 2020). Each clip is processed through a four-stage pipeline.

**(i) 3D reconstruction.** Following TRAM (Wang et al., 2024a), a joint human-and-camera estimator recovers metric-scale global camera extrinsics $\{(\mathbf{R}_t, \mathbf{t}_t)\}_{t=1}^{T}$ and per-frame SOMA body parameters from monocular video. Clips for which TRAM fails to converge or returns low-confidence estimates are discarded.

**(ii) Projection to $\mathcal{K}_{\text{cam}}$.** The recovered camera and actor trajectories are projected into the human-centric parameter space defined in Section 3.1: for each frame $t$, we compute the six camera axis values $(\text{CL}, \text{LL}, \text{SS}, \text{FO}, \text{OA}, \text{DA})_t$ from the world-space pose relative to the body frame $\mathcal{F}_t$. This yields a dense continuous trajectory in $\mathcal{K}_{\text{cam}}$ for each clip.

**(iii) Motion tagging.** We extract a sparse keyframe program from the dense trajectory using a per-axis change-point detector. For each axis $a$, we identify frames at which the axis value undergoes a transition larger than a threshold $\Delta_a$ (calibrated per axis to reflect cinematographic significance, e.g., a change of one shot-scale category for SS, or a $30^\circ$ azimuth change for OA). At each detected change point, the new axis value is snapped to the nearest token in $\mathcal{V}_a$ via the inverse of the per-axis lookup $\varphi_a$. The resulting keyframes form $\mathcal{P}_{\text{cam}}$. The same procedure applied to the actor trajectory yields $\mathcal{P}_{\text{act}}$.

**(iv) Captioning.** We obtain a free-form natural-language caption $\ell$ for each clip using Aurora-Cap (Chai et al., 2024). To improve linguistic diversity and robustness to in-the-wild user prompts, we additionally generate paraphrased variants by prompt rewriting with Gemini (Gemini Team, 2024). Each clip is associated with multiple caption variants during training, sampled uniformly per epoch.

Each successfully processed clip contributes one tuple $(\ell, \mathcal{P}_{\text{act}}, \mathcal{P}_{\text{cam}})$ to the real-world split.

### A.6.3 Statistics

**Real-world split.** Table A.4 reports the distribution of camera-and actor-field changes per sequence in the single human subset of real-world split. Most sequences are static or involve one-to-two camera-field changes per clip; multi-axis transitions (three or more axes changing simultaneously) form a long but well-represented tail. Among individual axes, orientation and shot-scale changes are the most frequent camera events, consistent with standard cinematographic practice that uses repositioning and re-scaling as the primary shot-transition mechanisms (Mascelli, 1965; Bordwell and Thompson, 2020). Dutch-angle changes are rare in mainstream cinema, which is reflected in the low count for that axis; the procedural split compensates by upweighting Dutch-angle samples during synthesis.

Table A.4: **Real-world split statistics.** *Left:* number of camera-field changes per sequence; most sequences are static or involve one–two field changes. *Center:* per-field change frequency; orientation and shot-scale (depth) changes dominate, consistent with standard cinematographic practice (Mascelli, 1965; Bordwell and Thompson, 2020). *Right:* human-field changes per segment.

| # Changes | Count |
|---|---|
| Static (0) | 14,639 |
| 1-field | 1,910 |
| 2-field | 1,534 |
| 3-field | 917 |
| 4-field | 440 |
| 5-field | 158 |
| 6-field | 8 |

| Camera Field | Changed |
|---|---|
| Orientation | 2,527 |
| Shot scale | 2,403 |
| Framing offset | 2,281 |
| Look-at level | 1,820 |
| Camera level | 1,184 |
| Dutch angle | 112 |

| # Changes | Count |
|---|---|
| Static (0) | 25,682 |
| 1-field | 5,438 |
| 2-field | 4,085 |
| 3-field | 3,171 |
| 4-field | 836 |

**Comparison to existing datasets.** Table A.5 compares the Auteur dataset to prior camera-trajectory datasets along several axes: support for camera and human annotation, multi-human scenes, frame and sample counts, and average caption length. While prior datasets such as DataDoP (Zhang

et al., 2025), E.T. (Courant et al., 2024), and PulpMotion (Courant et al., 2026) are larger in raw scale, Auteur is the first to combine (i) explicit human annotation, (ii) multi-human support, and (iii) DSL-aligned camera framing labels in a single corpus. Our contribution emphasizes annotation quality and structural alignment with cinematographic conventions rather than raw scale; this design choice is validated by the strong cross-dataset generalization reported in Section 4.

Table A.5: **Dataset comparison.** Comparison of various datasets with a focus on camera and human attributes. Our proposed dataset is shown at the bottom.

| | | | | | | *Vocabulary* | | *Avg. Cap. Len.* | |
|---|---|---|---|---|---|---|---|---|---|
| Dataset | Camera | Human | Multi-Human | #Frames | #Samples | Cam. | Hum. | Cam. | Hum. |
| DataDoP (Zhang et al., 2025) | ✓ | × | × | 11M | 29K | 8698 | × | 86.2 | × |
| E.T. (Courant et al., 2024) | ✓ | ✓ | × | 11M | 115K | 906 | × | 11.58 | × |
| PulpMotion (Courant et al., 2026) | ✓ | ✓ | × | 22M | 193K | 1002 | 4007 | 11.72 | 13.92 |
| Ours | ✓ | ✓ | ✓ | 2.7M | 34K | 3264 | 5667 | 53.8 | 120.1 |

**Combined corpus.** The full training set comprises $N_{\text{proc}}$ procedural and $N_{\text{real}}$ real-world tuples for a total of $34{,}000$ aligned samples. The two splits are deliberately complementary: procedural data ensures broad axis coverage and exact ground truth for supervision of rare combinations, while real-world data exposes the model to the correlations, noise, and stylistic diversity of professional cinematography. The ablation in Section 4 (Table 3) confirms that combining the two is critical. A synth-only model degrades sharply on paraphrased real-world prompts, particularly on linguistically-mediated axes such as shot scale where natural captions are most ambiguous.

Overall, we view Auteur as a step toward more transparent and controllable generative systems, where structured representations can facilitate both creative empowerment and responsible oversight.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are accurate. The introduction ends with a bullet-point summary of the main contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The conclusion ends with a paragraph that is explicitly named as **Limitations** and discusses the shortcomings of our work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The main paper and the appendix section provide sufficient details for reproducing our results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The submission does not provide open access to data and code in its current form. We plan to release them upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The training data is explained in in detail Section 3.4 whereas the testing datasets are discussed in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not provide this due to limited compute resources, we avoid generating results with multiple seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The relevant information is provided at the beginning of Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include a paragraph that discusses the broaded impact in the conclusion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We discuss potential ways for safeguarding in the conclusion.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the datasets used in our training and testing.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The videos generated by our method are provided in the supplementary webpage and documented explicitly.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not conduct subjective experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No subjective experiments were conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Our method finetunes an LLM to predict a cinematographic DSL clearly explained by the paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.