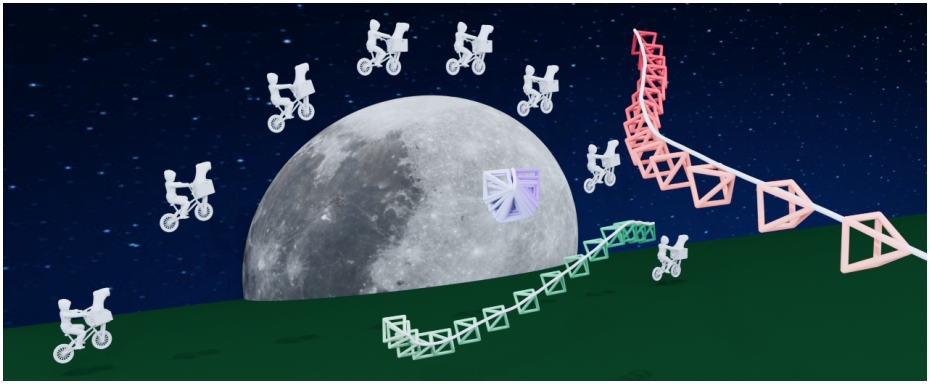
E.T. the Exceptional Trajectories: Text-to-camera-trajectory generation with character awareness

Robin Courant[1] Nicolas Dufour[1,2] Xi Wang[1] Marc Christie[3] Vicky Kalogeiton[1]

[1] LIX, Ecole Polytechnique, IP Paris
[2] LIGM, Ecole des Ponts, CNRS, UGE
[3] Inria, IRISA, CNRS, Univ. Rennes



Prompt: The camera trucks right to follow the character.

Prompt: The camera performs a push-in to get closer to the character.

Prompt: The camera stays static the entire shot.

Fig. 1: Different results generated by our camera trajectory diffusion system. Project page https://www.lix.polytechnique.fr/vista/projects/2024_et_courant.

Abstract. Stories and emotions in movies emerge through the effect of well-thought-out directing decisions, in particular camera placement and movement over time. Crafting compelling camera trajectories remains a complex iterative process, even for skilful artists. To tackle this, in this paper, we propose a dataset called the Exceptional Trajectories (E.T.) with camera trajectories along with character information and textual captions encompassing descriptions of both camera and character. To our knowledge, this is the first dataset of its kind. To show the potential applications of the E.T. dataset, we propose a diffusion-based approach, named DIRECTOR, which generates complex camera trajectories from textual captions that describe the relation and synchronisation between the camera and characters. To ensure robust and accurate evaluations, we train on the E.T. dataset CLaTr, a Contrastive Language-Trajectory embedding for evaluation metrics. We posit that our proposed dataset and method significantly advance the democratization of cinematography, making it more accessible to common users.

1 Introduction

Cinematography is a collaborative and complex crafting process that mixes technical, artistic and storytelling skills. The ultimate objective is to communicate a distinct message to the audience, at a cognitive (e.g., revealing facts), emotional and aesthetic level, through tasks such as laying out the scene (*mise-en-scène*), setting up the lighting and making decisions to place and move the camera in relation to the characters, their actions or the overall scene content. In this context, the camera is the only window into this staged world and therefore plays a critical role in conveying the director's intention. Through more than a hundred years of practice, cinematography has forged a common language for directors – the *film grammar* – that prescribes how to place and move the camera to achieve intended effects. Yet mastering camera placements and motions remains challenging, especially for novice users confronted with hundreds of possibilities and little insights into how to generate the best ones.

To lower the barriers in handling camera placement and camera motion, researchers have introduced a variety of methods. These include purely geometric approaches [4, 30], optimization- and control-based strategies [11, 12], as well as deep learning-grounded methodologies [5, 11, 20, 23] to interactively or automatically compute the parameters of camera trajectories. Typically, these methods address cinematographic tasks as either cinematic-rule-based control [5, 12, 20] or example-based imitation [22, 23, 45], conceptually resembling discriminative and regression models or registration and adaptation methods, respectively. Such techniques, however, suffer from the need to either design the underlying geometric model for each type of motion, or to design carefully crafted cost functions for each motion, and are often limited in their capacity to combine mixed motions creatively.

Recent advances in video generation [46, 52] enable users to explore more creative possibilities by capturing and reproducing camera motion in their generated videos. Jiang *et al.* [24] followed this path and addressed camera trajectory generation using diffusion models, which incorporate a high degree of controllability. Yet, this work displayed two main drawbacks: first, it relied on a character-centric coordinate system to simplify the problem, thus limiting its generation capabilities, and second its evaluation metrics relied on camera trajectory features with oversimplified assumptions.

In other domains, the generative techniques often rely on the availability of large datasets enriched with textual descriptions, such as language-motion obtained via motion capture (mocap) [14, 36] or language-vision [29, 40] datasets. Yet in cinematography, there is no movie datasets where crucial cinematic information such as camera and character trajectories are available. Most recent approaches build on synthetic data [22–24], or general videos from streaming platforms (see [20] for drone trajectory generation, or [53] for dedicated real-estate videos) without the cinematic features that conform to the film grammar. Some example-based approaches address cinematic transfer tasks from real film clips [25, 45], these approaches only retarget and adapt the camera trajectory

with little control or variability in the results and do not encode cinematographic knowledge.

In this work, we propose a new camera trajectory dataset extracted from real movie clips, called *E.T. the Exceptional Trajectories*. It comprises camera trajectories together with textual descriptions of both camera and character trajectory over time (see Figure 2). E.T. contains more than 11M frames with the corresponding camera and character trajectories, as well as two types of captions: camera-only and camera-character, describing the trajectory of the camera with respect to the trajectory of the character. To our knowledge, E.T. is the first extensive dataset with geometric information on both camera and character trajectories accompanied by textual descriptions.

To exploit this dataset, we also propose DIRECTOR (DiffusIon tRansformEr Camera TrajectORy), a diffusion-based model that generates camera trajectories by leveraging text descriptions and character information, as shown in Figure 1. This allows us to better encode the correlation between character and camera trajectories. Moreover, unlike previous methods [24] that use a constrained character-relative coordinate system, we propose to use a global coordinate system. DIRECTOR relies on a classical diffusion framework with three distinct architectures for conditioning: in-context, AdaLN and cross-attention settings. Furthermore, we propose a language-trajectory embedding: CLaTr (Contrastive Language-Trajectory), trained at scale using the E.T. dataset. CLaTr serves as a foundation for computing default generative metrics similar to Frechet-Inception-Distance (FID) [16] for generated trajectories. Our experiments show that all three architectures of DIRECTOR successfully leverage the combination of input captions and character trajectories as conditions. Overall, DIRECTOR sets the new state-of-the-art on the camera trajectory generation task.

Our contributions are: (1) We introduce the E.T. camera trajectory dataset extracted from real movie clips. We complement camera trajectories with character trajectories and captions for both camera and character. (2) We present DIRECTOR, a camera trajectory diffusion model that exploits both character trajectories and textual descriptions. It offers higher controllability and granularity for users than existing approaches [24] and achieves state-of-the-art performances. (3) We propose CLaTr, a robust and accurate language-trajectory embedding, which facilitates the evaluation of camera trajectory generation models.

2 Related work

Camera control. Over the past twenty years, there have been several paradigm shifts in camera planning and control. Initial studies [4] predominantly focused on geometric modeling [30] and rule-based trajectory controls [11] to direct and create camera trajectories that comply with either hand-crafted cinematic rules or image-based criteria. With the progress of deep learning, [23] introduced a method to synthesize camera trajectory for 3D animations in two stages: (i) capturing cinematic styles from a reference clip using a Mixture-of-Experts model, and (ii) generating trajectories based on 3D character animations autoregres-

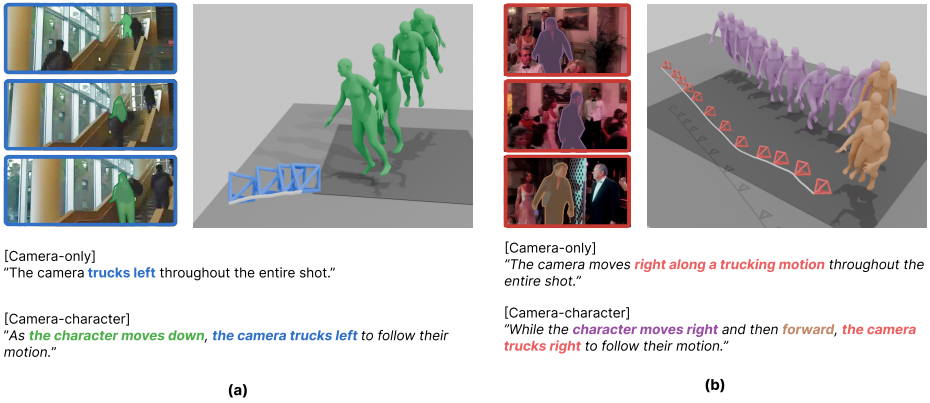


Fig. 2: Examples E.T. samples. Each subfigure presents frames from the original movie shot on the left, while the right side depicts the extracted and processed camera and character trajectories. Additionally, the bottom part showcases the generated camera trajectory caption with or without the character trajectory.

sively. Subsequent research [22], building on this, incorporates keyframing to provide extra control such as positional and velocity constraints. More recently, JAWS [45] pioneered the direction for example-based camera retargeting within a Neural Radiance Field (NeRF) [31] setting, by optimising camera trajectory directly given the 2D reference clip in a 3D NeRF. All these example-based methods share a common limitation: they struggle with generalization because they require carefully selected reference videos to ensure high quality.

Unlike example-based methods, many cinematic-rule-based methods readily integrate with Deep Reinforcement Learning (DRL) and Imitation Learning (IL) techniques, particularly in the drone cinematography domain: [20] exploit optical flow and human poses to guide drone controls via an IL framework. Similarly, [5] use DRL to control drone actions for multiple rewards, including obstacle avoidance, target tracking, shooting style etc. Recently, GAIT [48] employs an aesthetic score-based RL method instead of handcrafted rewards to control the camera in the virtual 3D environment. However, these RL-based camera control approaches also have limitations: (1) they need environment-specific training; (2) they inherently restrict the diversity of results, often leading to collapsed trajectory styles. Instead, we leverage the generalization capabilities of generative models to address the camera control task.

Camera diffusion. Generative models have recently gained much progress and attention in domains such as textual-conditioned image generation [33, 37, 39], video synthesis [3, 41] and human motion generation [7, 42, 51]. Among these, diffusion models stand out for their strong ability to produce high-fidelity and diverse generative samples [10, 47], making them particularly well-suited for camera trajectory generation tasks.

The first application of diffusion models in camera control is the Cinematographic Camera Diffusion (CCD) [24], which relies on the MDM architecture

(human Motion Diffusion Model) [42] and is trained on synthetic data. However, CCD simplifies the task by expressing all the camera trajectories in character-centric relative coordinates. Its small-scale synthetic training dataset also limits the broader application of the method (e.g., only 48-size vocabulary is used during training), thus making it unable to generate camera trajectories from real datasets and, in turn, impractical for common users. In contrast, in our proposed E.T. dataset, we represent camera trajectories in a global coordinate system, distinct from character trajectories. This approach allows for more diverse correlations between character and camera movements. Additionally, E.T. offers a rich vocabulary ($\sim$ 5.4k) and extensive camera trajectory data.

Recent literature also includes several text-to-video generation techniques that can handle different categories of camera motions [46, 52]. These methods, however, assume access to 3D camera trajectories, whereas our approach generates them. Furthermore, they typically overlook the camera's primary targets (i.e., the characters), which are essential for defining camera trajectories. In contrast, our dataset contains character information, and we leverage it to generate camera trajectories that focus on a specific target character.

Camera trajectory datasets. Many modern generative methods leverage large multimodal datasets. For instance, in text-to-image generation, the default dataset is LAION [40] with around 400 million image-text pairs. Similarly, in human motion synthesis, the large-scale KIT [36] and HumanML3D [14] datasets offer detailed textual captions that enhance comprehension of human motion. Yet, for camera control, only a few datasets are available [24, 53]. This is largely due to the intricacies involved in extracting camera poses from real-world videos, especially in cinematic contexts due to the presence of stylistic elements (*e.g.* motion blur or depth-of-field). Zhou et al. [53] applied Structure-from-Motion (SfM) methods to YouTube real-estate videos, creating the RealEstate10K dataset. This dataset, designed primarily for 3D reconstruction, comprises solely smooth camera movements and limited scene variation, lacking the nuanced complexity of cinematic camera motion and human presence. More recently Jiang et al. [24] introduced a synthetic cinematic camera trajectory dataset, aiming to circumvent extraction challenges. However, this dataset oversimplifies the intricate cinematic dynamics present in real-world movies.

A recent breakthrough in 3D human pose estimation for videos, termed SLAHMR [13], offers a compelling trade-off between robustness and accuracy by jointly optimizing camera and character trajectory estimations. Motivated by the lack of camera trajectory datasets, the capabilities of SLAHMR and the recent advances in other domains, we propose a new multi-modal camera trajectory dataset E.T. extracted from cinematic content, which we enhance with automatically generated captions for camera and character trajectories.

3 Exceptional Trajectories (E.T.)

We introduce a camera trajectory dataset called *Exceptional Trajectories* (E.T.), extracted from real movies. E.T. is built upon the Condensed Movies Dataset

Dataset	#Samples	#Frames	#Hours	Domain	Character		Camera		#Vocabulary
					Traj	#Captions	Traj	#Captions	
KIT Motion-Language [36]	4K	0.8M	11.23	Mocap	✓	6K		-	1,623
HumanML3D [14]	14K	2M	28.59	Mocap	✓	45K		-	**5,371**
RealEstate10k [53]	79K	**11M**	**121**	Youtube		-	✓	-	-
CCD [24]	25K	4.5M	50	Synthetic		-	✓	25K	48
E.T. (Ours)	**115K**	**11M**	120	**Movie**	✓	**115K**	✓	**230K**	1,790

Table 1: Dataset comparison. We compare the E.T. dataset to (i) two human motion datasets KIT [36] and HumanML3D [14]; and (ii) camera trajectory datasets RealEstate10K [53] and CCD [24]. Here the notion of sample is common across all datasets and corresponds to data associated with a continuous temporal sequence.

(CMD) [1]. Each *sample* in E.T. represents a camera trajectory at the shot level together with a character trajectory and two types of textual captions: a camera-only caption, which describes the camera motion; and a joint camera-character trajectory caption, which describes the motion of the camera according to the motion of the character (see Figure 2). Below, we describe the key properties and statistics of E.T. (Section 3.1) followed by the creation pipeline (Section 3.2).

3.1 E.T. properties and statistics

The key properties of E.T. are as follows:

Cinematic content. The camera trajectories in E.T. are both realistic and cinematic, since they are extracted from real-world movies (Table 1). This dual nature allows for effective modelling of various visual styles, in contrast to RealEstate10k's [53] focus on shots characterized by smooth camera trajectories and limited scene variation. Furthermore, by extracting data from real-world movies, E.T. sets itself apart from CCD [24], which only relies on synthetic camera trajectories.

Scale. E.T. is built upon 16,210 different scenes from CMD [1]. It comprises 115K samples spanning 11M frames and totalling 120 hours of footage, offering extensive and diverse camera and character (human) trajectories based on real movies. In contrast, existing human motion datasets are much smaller, with only 11.23 hours for KIT [36] and 28.59 hours for HumanMl3D [14] (see Table 1). When compared against datasets with camera trajectories, it far exceeds CCD [24] in terms of hours, frames and samples. Although its scale is comparable to RealEstate10k [53], it provides additional character trajectories and captions referring to real movies as opposed to RealEstate10k, which focuses only on camera trajectories in another domain.

Controllability. E.T. stands out by comprising not only camera and character trajectories but also camera-only and camera-character captions (see Figure 2). Incorporating caption information into the model offers multiple advantages: (1) it democratizes the input format for general users; and (2) it adds complementary semantic information to the trajectory data. In comparison, RealEstate lacks captions entirely. CCD's captions are limited by a small vocabulary size and

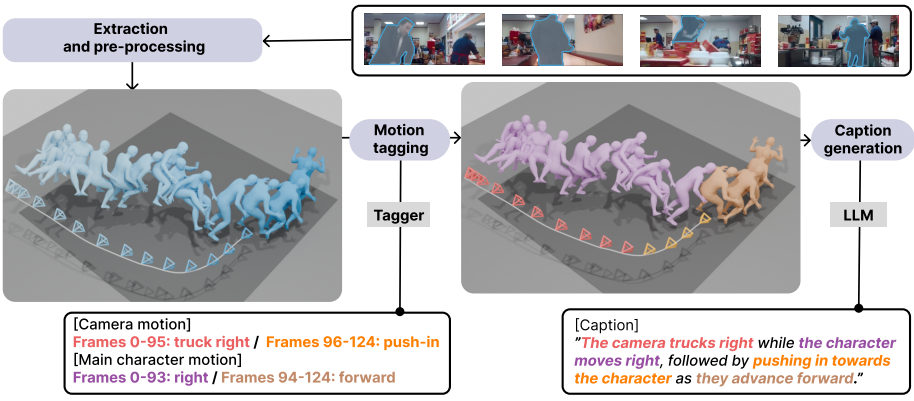


Fig. 3: Dataset creation pipeline. Given RGB frames from a video, we first extract and pre-process camera and character poses, then tag resulting camera and character trajectories (sequence of poses) to obtain rough independent descriptions (middle part). Finally, we translate these descriptions into rich textual captions, aligning the camera trajectory with that of the character (right part).

focus only on camera while lacking character information[4]. The richness and complexity of E.T.'s captions are on par in terms of vocabulary size –above a thousand– with human motion datasets such as KIT and HumanML3D, which provide detailed, hand-crafted human motion descriptions[5].

Statistics. Figures 4a- 4b display the statistics of the E.T. dataset, confirming the diversity and all six degrees of freedom coverage of both camera and character trajectories (see more in Appendix B.1.)

3.2 Dataset creation pipeline

E.T. is constructed by a three-step process (see Figure 3). First, we extract the 3D coordinates of cameras and characters over time, which we further refine to form uniform trajectories. Second, we perform *motion tagging*, i.e. partition each trajectory into segments with each segment comprising a pure camera motion that we label (tag). Third, we generate captions that describe both the camera and the character trajectory over time. We detail each step below.

Data extraction and pre-processing. To extract camera and character poses, we apply on each shot the joint camera and 3D human poses estimator SLAHMR [50]. Given the complexity of estimating 3D poses from 2D data, the raw outputs

[4] Note that CCD indirectly comprises camera trajectories through the character-relative coordinate system.

[5] Note that E.T. has no overlap with human motion datasets. E.T.'s extracted 3D poses (see Section 3.2) are less accurate than the ones in motion capture, while its captions describe camera trajectory relative to character trajectory, as opposed to describing exact human motions targeted by these datasets.

tend to be noisy. To address this, we perform various pre-processing steps such as alignment, filtering, smoothing and cropping to a maximum length of 300 frames as in [14]. Refer to the Appendix B.2 for further details.

Motion tagging. Our objective is to partition camera or character trajectories into segments of pure motion: tags. Besides static, we consider the six fundamental motions across three degrees of freedom. They include lateral movements left, and right; vertical movements up and down; and depth movements forward and backwards. Each trajectory is partitioned into motion tags with one, two, or three pure camera motions, totalling 27 combinations (see Figure 4a).

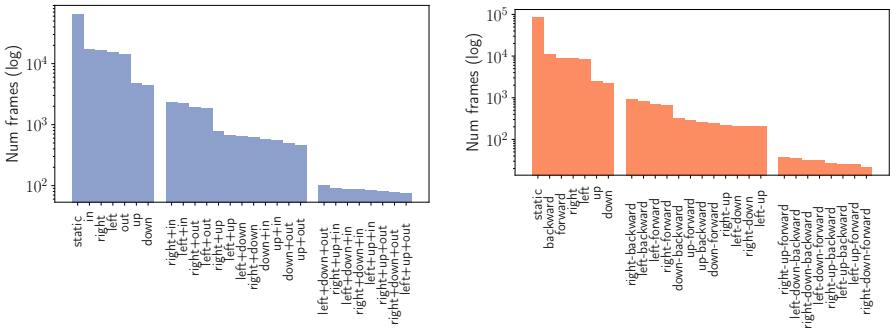
We propose a thresholding-based method that uses trajectory velocity for motion tagging: This method consists of two stages: (i) for each dimension (XYZ), we use an initial threshold on velocity to detect whether the camera or character remains static along the dimension; (ii) when multiple dimensions are non-static, we calculate pairwise velocity rates and use a threshold to pinpoint dominant velocities. A dimension is classified as static if its velocity is outmatched. The tag of motion between two points is then determined by the combination of non-static dimensions. Finally, we apply smoothing to avoid noisy and sparse tags and hence enhance the overall trajectory-level tagging.

For *camera trajectory tagging*, we use the rigid body velocity $\in SE(3)$ – derived from rotation and translation– to account for the camera's facing direction. this enables us to differentiate between similar motions, such as 'trucking', where the camera moves along an axis with a perpendicular facing direction, and 'depth', where the facing direction aligns with the movement axis. For *character trajectory tagging*, we assume that characters face the direction of their movement. Hence, we represent character trajectory using only the linear velocity, as derived from the translation of their hip centres.

These result in a coarse description of both camera and character trajectories over time as shown in Figure 3 (left).

Caption generation. Our objective is to provide rich textual descriptions of the extracted camera trajectories according to the character trajectory. In movie, cameras typically move relative to the subject being filmed, i.e., the main character. Therefore, for each shot, we first identify the main character following [43][6] based on the temporal and spatial coverage of their bounding boxes within the shot. Then, for both camera and main character trajectories, we generate captions for each motion tag, as shown in the center of Figure 3. Then, inspired by [9], our goal is to convert the descriptions obtained via motion tagging for camera and character trajectories into detailed textual annotations. For this, we prompt an LLM –Mistral-7B [21]– to generate camera trajectory captions by referencing the main character's trajectory as anchor points. Our prompt formulation follows a structured approach with context, instruction, constraint, and example. Further details can be found in the Appendix B.3.

[6] Hitchcock's rule: '*the size of an object in the frame should equal its importance in the story at the moment*' [43].



(a) Camera segment distribution

(b) Character segment distribution

Fig. 4: E.T. statistics.

This step results in a rich description of both camera and character trajectories over time as shown in Figure 3 (right).

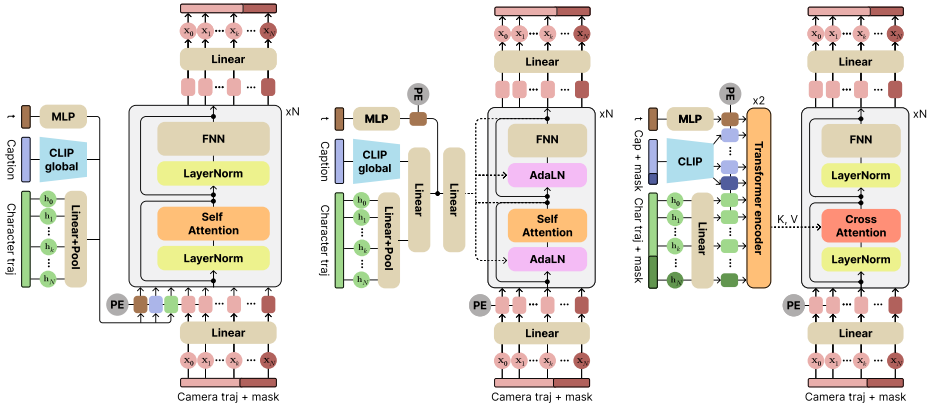
4 Method

Here, we introduce our proposed *DiffusIon tRansformEr Camera TrajectORy* (DIRECTOR) method for camera trajectory generation (Section 4.1). DIRECTOR takes as input the character trajectory with the camera-character caption and generates a camera trajectory. Additionally, we present the *Contrastive Language-Trajectory* embedding (CLaTR) that serves as a basis for creating a common space between text and trajectories (Section 4.2), enabling the computation of evaluation metrics.

4.1 Camera trajectory diffusion

Problem formulation. We consider a camera trajectory $\mathbf{x}_{1:N}$ as a sequence of N consecutive camera poses. Each camera pose $\mathbf{x} = [\mathbf{R}|\mathbf{t}]$ comprises a rotation $\mathbf{R}$ representing the camera's orientation and a translation $\mathbf{t}$ indicating its position. We aim at generating camera trajectories under two conditions: (i) a target character trajectory $\mathbf{h}_{1:N}$ capturing the 3D positions of the main character; and (ii) a textual description c specifying the desired camera movement relative to the character movement.

Diffusion framework. We follow the general diffusion paradigm established in EDM [26]. In essence, diffusion models consist of randomly sampling $\mathbf{x}^0 \sim \mathcal{N}(\mathbf{0}, \sigma^2_{max}\mathbf{I})$, and progressively denoising it to reach the endpoint $\mathbf{x}^K$ of this process, distributed according to the initial data distribution. During the training stage, we perturb an initial data distribution with standard deviation σ_{data}, with i.i.d. Gaussian noise with standard deviation σ. When $\sigma_{\max} \gg \sigma_{\text{data}}$, the noise distribution equivalent to a normal distribution $\mathcal{N}(\mathbf{0}, \sigma^2_{max}\mathbf{I})$. We use these



(a) Director A **(b)** Director B **(c)** Director C

Fig. 5: ***DiffusIon tRansformEr Camera TrajectORy*** **(Director).** We display 3 variants of our diffusion model Director. Director A incorporates the conditioning as in-context tokens. Director B leverages AdaLN modulation of the transformer block to add the conditioning. Director C uses the full text and character trajectory sequences by relying on cross-attention.

modified versions of the initial data distribution to train a denoiser module D, which takes as input a sample $\mathbf{x}$ to denoise, the two conditions (character trajectory $\mathbf{h}$ and the caption c), and the corresponding standard deviation σ. Then, D is trained using the denoising score matching loss:

$$\mathcal{L}_{\text{score}} = \big(D(\mathbf{x}, \mathbf{h}, c;\ \sigma) - \mathbf{x}\big)/\sigma^2. \tag{1}$$

During the sampling phase, we apply the 2nd order deterministic sampling introduced in EDM [26] with classifier-free guidance [19].

Director architecture. Director (*DiffusIon tRansformEr Camera TrajectORy*) takes as input the character trajectory and the caption and generates a camera trajectory. Its architecture is illustrated in Figure 5. The base of Director is a pre-norm Transformer [44,49]. We condition the transformer on the diffusion timestep, the character trajectory, and a textual description that describes the relative movement between the camera and character trajectories (see Figure 2). The timestep is tokenized using a sinusoidal positional embedding [44] and then mapped with an MLP.

Inspired by the DiT architecture variants [34], we explore three distinct ways to include the conditioning in the denoising process (Figure 5).

Director A (Figure 5a). The conditioning is added to the ***context*** of the transformer input. We only use the global clip token for the text, and we do a linear embedding of the character trajectories, which in turn gets averaged pooled into a single token.

Director B (Figure 5b). Both conditionings (character trajectory and caption) are concatenated into a single token which gets mapped at each layer into 6

vectors, $\gamma_1, \beta_1,\ \lambda_1, \gamma_2, \beta_2,\ \lambda_2$. Then, the layer-norm of the transformer is replaced by the following ***AdaLN*** operation:

$$\text{ADALN}(\gamma, \beta, x) = (1 + \gamma)\text{LN}(X) + \beta \quad , \tag{2}$$

where LN refers to the Layer Normalization, γ, β are the scale and bias, respectively. The AdaLN operation is performed before each self-attention and feed-forward layer in the transformer. The output of each self-attention and cross-attention is rescaled by λ. Following [34], we initialize the modulation such that the output is zero.

DIRECTOR C (Figure 5c). We leverage the full sequence length of the conditioning. We retrieve the CLIP-embedded text sequence and the linearly projected trajectory and concatenate them into a single sequence. We then use 2 layers of transformer encoders to pre-process this sequence, which is then incorporated into the DIRECTOR transformer with a ***cross-attention*** block.

4.2 Contrastive Language-Trajectory embedding (CLaTr)

Given the scarcity of relevant camera trajectory methods and datasets, the community has not introduced adequate metrics for this task. In the concurrent cinematic camera trajectory diffusion work [24], the authors evaluate their model with metrics from the human motion community. For this, they train a dedicated camera trajectory classifier to extract features. However, their classifier is trained on a simplistic task, comprising only six basic camera motion classes on synthetic data, which fails to capture the true complexity of camera trajectories.

To address this lack of proper evaluation metrics, in this section, we propose to extend existing metrics from text-image-based and text-motion-based generation (which rely on feature embeddings to measure the generation quality) to text-trajectory generation. The main obstacle is that no commonly accepted text-trajectory feature embedding exists. Therefore, we propose to learn a general text-trajectory embedding in a contrastive CLIP-like manner to acquire an accurate and robust feature representation, which can serve as a foundation for computing camera trajectory evaluation metrics.

We introduce *Contrastive Language-Trajectory* embedding (CLaTr) by capitalizing our multi-modal dataset E.T. with a CLIP-like approach [38]. Our language-trajectory embedding follows the methodology outlined in [35], originally designed for human motion. CLaTr consists of a VAE [27] framework with trajectory and text encoders and a shared feature decoder. CLaTr is trained with three losses: (a) a reconstruction loss $\mathcal{L}_R$, quantifying trajectory reconstruction of both trajectory and text features; (b) four KL loss terms $\mathcal{L}_{KL}$, which regularize each modality distribution and also enforce inter-modality similarity; and (c) a cross-modal embedding similarity loss $\mathcal{L}_E$, ensuring alignment between text and trajectory features. See Appendix C for more details.

Set	Methods	ω	Camera trajectory quality					Text-camera coherence			
			FD_{CLaTr} ↓	P ↑	R ↑	D ↑	C ↑	CS ↑	C-P ↑	C-R ↑	C-F1 ↑
E.T. pure trajectories	CCD [24]	5.5	31.33	0.79	0.55	0.83	0.72	3.21	0.53	0.28	0.27
	MDM [42]	1.8	6.10	0.77	0.68	0.89	0.80	21.26	0.81	0.75	0.76
	DIRECTOR A	1.6	5.16	0.82	0.67	1.00	0.86	21.88	0.84	0.78	0.80
	DIRECTOR B	1.8	6.61	0.80	0.72	0.92	0.82	23.10	0.85	0.80	0.86
	DIRECTOR C	1.6	4.57	0.83	0.65	1.00	0.87	21.49	0.83	0.78	0.80
E.T. mixed trajectories	CCD [24]	6.0	35.81	0.73	0.55	0.75	0.67	6.26	0.37	0.20	0.17
	MDM [42]	2.0	6.79	0.78	0.65	0.85	0.76	18.32	0.36	0.36	0.34
	DIRECTOR A	1.4	3.88	0.82	0.68	0.98	0.85	20.76	0.43	0.43	0.42
	DIRECTOR B	1.6	6.10	0.78	0.74	0.85	0.78	20.78	0.41	0.40	0.39
	DIRECTOR C	1.4	3.76	0.83	0.67	1.00	0.86	21.95	0.49	0.49	0.48

Table 2: Quantitative Results. Comparison of DIRECTOR and concurrent methods on E.T. pure and mixed subsets, evaluating trajectory quality (left) and caption coherence (right). First best and second best.

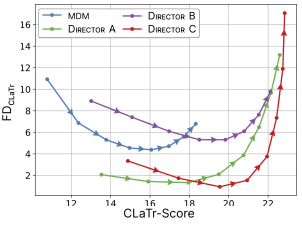


Fig. 6: FD_{CLaTr} vs CLaTr-Score. Guidance range between 0.6 and 2.2 on E.T. mixed subset.

5 Experiments

Implementation details. We train DIRECTOR with a batch size of 128 using the AdamW optimizer with a learning rate of 1e-4, (β_1, β_2)=(0.9, 0.95) and a weight decay of 0.1. We use a cosine decay learning rate scheduler with 5k steps of warmup for a total of 170k steps in bfloat16 mixed precision. The model has 8 layers with a hidden dim of 512 and 16 attention heads. We use dropout and stochastic depth of 0.1. We set the default temporal input size to 300 to match the E.T. sample size (see Section 3.2) and use masking to handle inputs with fewer than 300 frames. For the camera trajectory, we use the 6D continuous representation for rotation [54] combined with the 3D translation component. For the character trajectory, we use the 3D position of the character's hip center.

5.1 Quantitative results

Metrics. We use two sets of metrics.
First, we assess **camera trajectory quality**, specifically how well the generated camera trajectories match the distribution of the ground truth camera trajectories. For this, we use the CLaTr-based metrics described in Section 4.2: the Frechet CLaTr Distance (FD_{CLaTr}) similar to FID [17]), Precision (R), Recall (R), Density (D) and Coverage (C) [32]. As the validation set comprises only a few samples and these metrics need a critical amount of samples (10k+), we compare to the train set as it is common practice in small dataset generative models (e.g. CIFAR image generation [18, 28]).
Second, we use **text-camera coherence** metrics, which measure the coherence between the given caption (text) and the generated camera trajectory. For this, we use the CLaTr-Score (CS) (see Section 4.2), similar to CLIP-Score [15]. Additionally, we derive Classifier Precision (C-P), Classifier Recall (C-R) and Classifier F1-Score (C-F1) by performing motion tagging (described in Section 3.2) on generated camera trajectories and compare them to the ground truth.

Dataset. In our experiments, we train and evaluate our model on two different subsets of the E.T. dataset. First, the *pure camera trajectory subset*, where we

only keep the samples having a single camera motion trajectory (e.g. "*the camera trucks right*"). Second, the *mixed camera trajectories subset*, which excludes some static-only camera trajectories to create a balanced subset. In this way, we can both correctly compare against methods suited for simple, pure trajectories and emphasize the difficulty of the mixed compositional camera trajectories. We compare in Table 2 DIRECTOR with concurrent methods on the pure subset (top) and mixed subset (bottom).

Comparison to the state of the art. We report in Table 2 and Figure 6 quantitative results of the different DIRECTOR architectures against the previous state-of-the-art CCD [24], and MDM [42], a default modern method in human motion. We observe that overall we outperform both works on all metrics and both subsets. Particularly, in the mixed trajectory subset (bottom of Table 2), we demonstrate superior camera trajectory quality metrics (left section of Table 2) with a margin of -3.0 FD_{CLaTr} against MDM and -32.1 against CCD. Additionally, our method excels in text-camera coherence (right section of Table 2) within the same subset, achieving a substantial improvement of $+3.6$ ClaTR-Score against MDM and $+15.7$ against CCD.

Additionally, we show in Figure 6 the trade-off between FD_{CLaTr} (trajectory quality) and CLaTr-Score (conditioning coherence) for varying guidance weights. The optimal point is at the bottom right, where FD_{CLaTr} is lowest and CLaTr-Score is highest. We observe that the MDM curve (blue) consistently lies above DIRECTOR's curves, indicating that MDM performs worse.

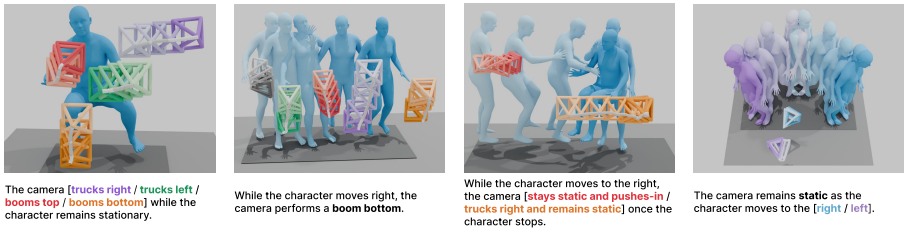
These results reveal the effectiveness of our method both in generating high-quality camera trajectories and in handling the input caption conditioning.

Ablation of DIRECTOR architectures. We observe in Table 2 and Figure 6 that DIRECTOR C outperforms other variants, followed closely by DIRECTOR A. The cross-attention mechanism in DIRECTOR C enables effective incorporation of conditioning into the model, leading to its superior performance. DIRECTOR A offers a compelling balance of efficiency and performance: it exhibits comparable results to DIRECTOR C with a simpler concept and fewer parameters. In contrast, DIRECTOR B excels in text-camera coherence on the pure trajectory subset (top-right of Table 2) but struggles on the mixed trajectory subset (bottom-right of Table 2). We attribute this to the AdaLN's ability to condition the model in simple setups, but its failure to capture sequential complexity in harder scenarios.

5.2 Qualitative results

Figure 7 shows generated camera trajectories from DIRECTOR (architecture C). Each sub-figure displays the trajectories with pyramid markers for keyframes, along with character meshes and corresponding captions. The output trajectories are smooth and consistent with the input conditions. We highlight four key strengths of our method:

Controllability (Figure 7a). DIRECTOR offers high controllability: by modifying only two words in the caption, the user can generate all kinds of camera trajectories, e.g. "*trucks right*", "*trucks left*", "*booms top*" and "*booms bottom*".



(a) Controllability **(b)** Diversity **(c)** Complexity **(d)** Character-aware

Fig. 7: Qualitative results. Generated camera trajectories with corresponding prompts and character trajectories, highlighting (a) controllability, (b) diversity, (c) complexity, and (d) character awareness. Darker shades indicate later frames.

Diversity (Figure 7b). Given the same input conditions (i.e. character trajectory and caption), DIRECTOR generates diverse camera trajectories, allowing users to explore a wide range of creative and unique outputs.
Complexity (Figure 7c). DIRECTOR can handle complex input conditions, including character trajectories (e.g., "*moves right*" then "*stops*") and camera trajectories descriptions (e.g., "*stays static and pushes-in*" and "*trucks right and remains static*").
Character-awareness (Figure 7d). DIRECTOR effectively considers the character, generating camera trajectories that follow the character's movement when the prompt and character trajectory are mirrored.

6 Conclusion

We designed and implemented E.T., a dataset of camera and character trajectories extracted from movie sequences that we believe will be very beneficial to the community. In addition to their trajectories, E.T. comes with text captions that describe the camera and character trajectories over time. We showed how E.T. can be exploited to train a diffusion-based approach to generate complex camera trajectories from high-level textual descriptions which correlate the trajectory of the camera with the trajectory of the characters. For this, we propose the diffusion-based method DIRECTOR, which sets the new state of the art on camera trajectory generation. In the future, we plan to address the expressiveness of the trajectory captions, by including more information about modifiers and the exact position on the screen where the characters should be located.

Acknowledgements

This work was supported by ANR-22-CE23-0007, ANR-22-CE39-0016, Hi!Paris grant and fellowship, and was granted access to the HPC resources of IDRIS under the allocation 2023-AD011013951 made by GENCI. We would like to thank Hongda Jiang, Mathis Petrovich, Pierre Vassal and the anonymous reviewers for their insightful comments and suggestions.

References

1. Bain, M., Nagrani, A., Brown, A., Zisserman, A.: Condensed movies: Story based retrieval with contextual embeddings. In: ACCV (2020) 6, 18
2. Björck, Å.: Least squares methods. Handbook of numerical analysis (1990) 20
3. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023) 4
4. Blinn, J.: Where am I? what am I looking at? (cinematography). IEEE Computer Graphics and Applications (1988) 2, 3
5. Bonatti, R., Wang, W., Ho, C., Ahuja, A., Gschwindt, M., Camci, E., Kayacan, E., Choudhury, S., Scherer, S.: Autonomous aerial cinematography in unstructured environments with learned artistic decision-making. J. Field Robotics. (2020) 2, 4
6. Castellano, B.: Pyscenedetect. https://github.com/Breakthrough/PySceneDetect (2014) 18
7. Chen, X., Jiang, B., Liu, W., Huang, Z., Fu, B., Chen, T., Yu, G.: Executing your commands via motion diffusion in latent space. In: CVPR (2023) 4
8. Courant, R., Lino, C., Christie, M., Kalogeiton, V.: High-level features for movie style understanding. In: ICCV-W (2021) 21
9. Delmas, G., Weinzaepfel, P., Lucas, T., Moreno-Noguer, F., Rogez, G.: Posescript: 3d human poses from natural language. In: ECCV (2022) 8
10. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. In: NeurIPS (2021) 4
11. Drucker, S.M., Galyean, T.A., Zeltzer, D.: Cinema: A system for procedural camera movements. In: Symposium on Interactive 3D graphics (1992) 2, 3
12. Galvane, Q., Christie, M., Lino, C., Ronfard, R.: Camera-on-rails: automated computation of constrained camera paths. In: ACM Motion In Games (2015) 2
13. Goel, S., Pavlakos, G., Rajasegaran, J., Kanazawa*, A., Malik*, J.: Humans in 4D: Reconstructing and tracking humans with transformers. In: ICCV (2023) 5
14. Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., Cheng, L.: Generating diverse and natural 3d human motions from text. In: CVPR (2022) 2, 5, 6, 8, 23
15. Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., Choi, Y.: CLIPScore: A reference-free evaluation metric for image captioning. In: EMNLP (2021) 12
16. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS (2017) 3
17. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017) 12
18. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS (2020) 12
19. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS-W (2021) 10
20. Huang, C., Lin, C., Yang, Z., Kong, Y., Chen, P., Yang, X., Cheng, K.: Learning to film from professional human motion videos. In: CVPR (2019) 2, 4
21. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.d.l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al.: Mistral 7B. arXiv preprint arXiv:2310.06825 (2023) 8
22. Jiang, H., Christie, M., Wang, X., Liu, L., Wang, B., Chen, B.: Camera keyframing with style and control. ACM TOG (2021) 2, 4

23. Jiang, H., Wang, B., Wang, X., Christie, M., Chen, B.: Example-driven virtual cinematography by learning camera behaviors. ACM TOG (2020) 2, 3
24. Jiang, H., Wang, X., Christie, M., Liu, L., Chen, B.: Cinematographic camera diffusion model. Computer Graphics Forum (2024) 2, 3, 4, 5, 6, 11, 12, 13
25. Jiang, X., Rao, A., Wang, J., Lin, D., Dai, B.: Cinematic behavior transfer via nerf-based differentiable filming. arXiv preprint arXiv:2311.17754 (2023) 2
26. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. NeurIPS (2022) 9, 10
27. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. stat (2014) 11
28. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images. Toronto, ON, Canada (2009) 12
29. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014) 2
30. Lino, C., Christie, M.: Intuitive and efficient camera control with the toric space. ACM TOG (2015) 2, 3
31. Mildenhall, B., Srinivasan, P., Tancik, M., Barron, J., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) 4
32. Naeem, M.F., Oh, S.J., Uh, Y., Choi, Y., Yoo, J.: Reliable fidelity and diversity metrics for generative models. In: ICML (2020) 12
33. Nichol, A.Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., Mcgrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In: ICML (2022) 4
34. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV (2023) 10, 11
35. Petrovich, M., Black, M.J., Varol, G.: TMR: Text-to-motion retrieval using contrastive 3d human motion synthesis. In: ICCV (2023) 11, 23
36. Plappert, M., Mandery, C., Asfour, T.: The KIT motion-language dataset. Big Data (2016) 2, 5, 6
37. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: SDXL: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023) 4
38. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021) 11
39. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022) 4
40. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: LAION-400M: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114 (2021) 2, 5
41. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792 (2022) 4
42. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A.H.: Human motion diffusion model. In: ICLR (2023) 4, 5, 12, 13
43. Truffaut, F., Scott, H.: Hitchcock/truffaut. revised edition. Simon and Schuster (1985) 8
44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017) 10

45. Wang, X., Courant, R., Shi, J., Marchand, E., Christie, M.: JAWS: Just A Wild Shot for cinematic transfer in neural radiance fields. In: CVPR (2023) 2, 4
46. Wang, Z., Yuan, Z., Wang, X., Chen, T., Xia, M., Luo, P., Shan, Y.: Motionctrl: A unified and flexible motion controller for video generation. arXiv preprint arXiv:2312.03641 (2023) 2, 5
47. Xiao, Z., Kreis, K., Vahdat, A.: Tackling the generative learning trilemma with denoising diffusion GANs. In: ICLR (2021) 4
48. Xie, D., Hu, P., Sun, X., Pirk, S., Zhang, J., Mech, R., Kaufman, A.E.: GAIT: Generating aesthetic indoor tours with deep reinforcement learning. In: ICCV (2023) 4
49. Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., Liu, T.: On layer normalization in the transformer architecture. In: ICML (2020) 10
50. Ye, V., Pavlakos, G., Malik, J., Kanazawa, A.: Decoupling human and camera motion from videos in the wild. In: CVPR (2023) 7, 18, 20, 21
51. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: MotionDiffuse: Text-driven human motion generation with diffusion model. IEEE TPAMI (2024) 4
52. Zhao, R., Gu, Y., Wu, J.Z., Zhang, D.J., Liu, J., Wu, W., Keppo, J., Shou, M.Z.: Motiondirector: Motion customization of text-to-video diffusion models. arXiv preprint arXiv:2310.08465 (2023) 2, 5
53. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. ACM TOG (2018) 2, 5, 6
54. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: CVPR (2019) 12, 23

Appendix

A Ethical discussion

We discuss the ethical impact of our method across several aspects:

- *Creative Integrity:* It is a fine line between using AI tool to enhance the human creativity and allowing it to deprive human creative process. Under misusage, the proposed method could diminish the artistic expression instead of support it.
- *Intellectual Property*: The use of AI-generated content raises questions about ownership and copyright. The Intellectual Property ownership of the generated content can be debatable.
- *Job Displacement or Creation*: The automation of certain aspects of filmmaking could lead to concerns about job displacement within the industry, or under proper usage, may also help to create new types of jobs in the domain.

B Exceptional Trajectories dataset (E.T.)

B.1 Additional statistics

We build our E.T. dataset the Condensed Movies Dataset [1] (CMD), encompassing over 30,000 scenes from 3,000 diverse movies, totaling more than 1,000 hours of video. We segment each movie scene into continuous shots by leveraging changes in color and intensity between frames [6].

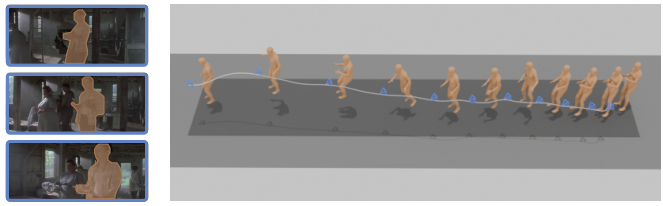
We show additional statistics of E.T. in Figure 9. We observe that for both camera and character, the majority of trajectories are smaller than 20 meters, i.e. corresponding to a velocity of 20 meters/(300 frames/25 fps) $= 1.67 m.s^{-1}$.

Additionally, in Figure 8, we show extensive examples of E.T. samples.

B.2 Data pre-processing

Chunk alignment. A limitation of SLAHMR [50] is its inability to handle long videos (exceeding 100 frames). Consequently, we divide each shot into chunks of 100 frames and process them independently. However, it produces non-consitant outputs: it exhibits translational bias/offset and different scales, as shown in Figure 10a.

To address this issue, we propose the following alignment method: dividing shots into overlapping chunks, where consecutive chunks share frames, and performing alignment on these overlapping frames. A chunk contains camera trajectories with $SE(3)$ poses represented as $[\mathbf{R}|\mathbf{t}]$ (where $\mathbf{R}$ denotes rotation and $\mathbf{t}$ translation), and 3D human poses described by $\mathbf{V}$ (vertices of a 3D mesh).



[Camera-only] "*The camera moves laterally to the left (trucking) throughout the entire shot.*"

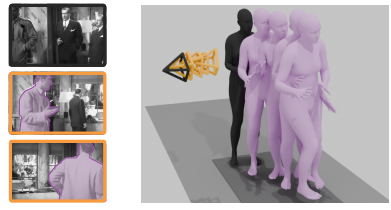
[Camera-character] "*As the character moves backward, the camera trucks left to maintain their position in the shot.*"

(a)

[Camera-only]
"*The camera **starts static**, then trucks left, and finally trucks right.*"

[Camera-character]
"*As t**he character stands still, the camera remains stationary**, then trucks left, followed by trucking right. Meanwhile, the character begins moving forward towards the end of the scene.*"

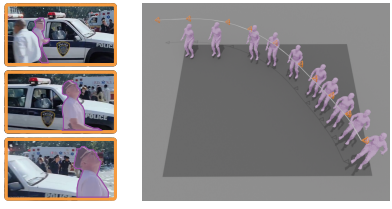
(b)



[Camera-only]
"*The camera begins with a static position and then moves inwards until the end of the shot.*"

[Camera-character]
The camera stays still as the character stands stationary, then moves in closer while the character moves to the right.

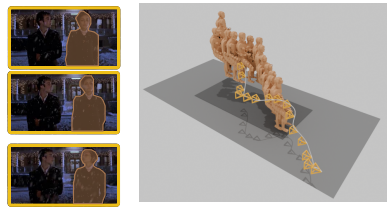
(c)



[Camera-only] "*The camera performs a trucking right-push-in motion throughout the shot.*"

[Camera-character] "*As the character moves right, the camera trucks right and pushes in.*"

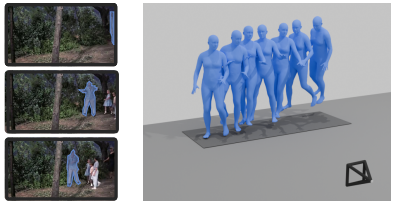
(d)



[Camera-only] "*The camera pulls out throughout the entire shot.*"

[Camera-character] "*As the character moves forward, the camera pulls out.*"

(e)



[Camera-only] "***The camera remains static** during the entire shot.*"

[Camera-character] "***The camera remains stationary** as the character moves to the left.*"

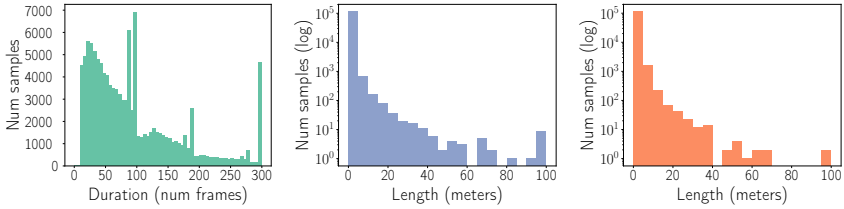
(f)

[Camera-only] "***The camera is static** during the entire shot.*"

[Camera-character] "***The camera remains static** as **the main character stays in place** throughout the entire sequence.*"

(g)

Fig. 8: Examples E.T. samples. Each subfigure presents frames from the original movie shot (left), and processed camera and character trajectories (right). Additionally, the bottom part showcases the generated camera trajectory caption with or without the character trajectory caption.

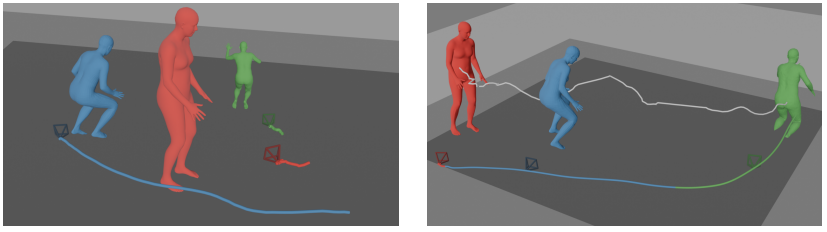


(a) Trajectory length (in #frames)

(b) Camera length (in meters)

(c) Character length (in meters)

Fig. 9: E.T. statistics.



(a) Before alignment.

(b) After alignment.

Fig. 10: Raw chunk alignment. We show in (a) the raw independent chunks just after the SLAHMR [50] extraction. In (b) we display the result of the chunk alignment process. Each color (red, blue, green) corresponds to a different chunk.

Given two consecutive chunks at k and $k+1$, we initially align the cameras. The alignment involves determining a scale parameter s and a $SE(3)$ rigid transformation $[\mathbf{B} \mid \mathbf{b}]$:

$$[\mathbf{R}_k \mid \mathbf{t}_k] = [\mathbf{B}_k \mid \mathbf{b}_k]\,[\mathbf{R}_{k+1} \mid s_k\,\mathbf{t}_{k+1}], \tag{3}$$

$$[\mathbf{R}_k \mid \mathbf{t}_k] = [\mathbf{B}_k\,\mathbf{R}_{k+1} \mid s_k\,\mathbf{B}_k\,\mathbf{t}_{k+1} + \mathbf{b}_k], \tag{4}$$

which simplifies to:

$$(a) \quad \mathbf{R}_k = \mathbf{B}_k\,\mathbf{R}_{k+1}, \tag{5}$$

$$(b) \quad \mathbf{t}_k = s_k\,\mathbf{B}_k\,\mathbf{t}_{k+1} + \mathbf{b}_k. \tag{6}$$

Notably, the rotation estimated by SLAHMR remains consistent across chunks, implying $\mathbf{B}_k = \mathbf{I}$, and simplifying Equations 5 and 6 :

$$(a) \quad \mathbf{R}_k = \mathbf{R}_{k+1}, \tag{7}$$

$$(b) \quad \mathbf{t}_k = s_k\,\mathbf{t}_{k+1} + \mathbf{b}_k. \tag{8}$$

Subsequently, alignment entails determining the scaling factor s and translational bias $\mathbf{b}$. These parameters can be accurately estimated using the least-square method [2], as represented by:

$$\begin{bmatrix}\mathbf{t}_k & \mathbf{I}\end{bmatrix} \begin{bmatrix} s_k \\ \mathbf{b_k} \end{bmatrix} = \mathbf{t}_{k+1}, \tag{9}$$

which can be further expressed as:

$$\begin{bmatrix} t_k^x & 1 & 0 & 0 \\ t_k^y & 0 & 1 & 0 \\ t_k^z & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_k \\ b_k^x \\ b_k^y \\ b_k^z \end{bmatrix} = \begin{bmatrix} t_{k+1}^x \\ t_{k+1}^y \\ t_{k+1}^z \end{bmatrix}. \tag{10}$$

We also seek the alignment transform Δ_b, such that:

$$[\mathbf{R}_{k+1} \mid s_k\, \mathbf{t}_{k+1} + \mathbf{b}_k]\, \boldsymbol{\Delta}_b = [\mathbf{R}_{k+1} \mid \mathbf{t}_{k+1}], \tag{11}$$

resulting in:

$$\boldsymbol{\Delta}_b = [\mathbf{R}_{k+1} \mid s_k\, \mathbf{t}_{k+1} + \mathbf{b}_k]^{-1}\, [\mathbf{R}_{k+1} \mid \mathbf{t}_{k+1}]. \tag{12}$$

Considering the inverse of a 4x4 transformation matrix representing a rigid transformation:

$$\begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{13}$$

we obtain from Eq. 12:

$$\boldsymbol{\Delta}_b = \begin{bmatrix} \mathbf{R}_{k+1}^T & -\mathbf{R}_{k+1}^T(s\mathbf{t}_{k+1} + \mathbf{b}_k) \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{k+1} & \mathbf{t}_{k+1} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{14}$$

$$\boldsymbol{\Delta}_b = \begin{bmatrix} \mathbf{I} & \mathbf{R}_{k+1}^T(\mathbf{t}_{k+1} - (s\mathbf{t}_{k+1} + \mathbf{b}_k)) \\ \mathbf{0} & 1 \end{bmatrix}. \tag{15}$$

Ultimately, to align the 3D human poses based on their vertices V:

$$\begin{bmatrix} \mathbf{V}_k^T \\ 1 \end{bmatrix} = \boldsymbol{\Delta}_b \begin{bmatrix} \mathbf{V}_{k+1}^T \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{k+1}^T + \mathbf{R}_{k+1}^T(\mathbf{t}_{k+1} - (s_k\mathbf{t}_{k+1} + \mathbf{b}_k)) \\ 1 \end{bmatrix}, \tag{16}$$

$$\mathbf{V}_k = \mathbf{V}_{k+1} + (\mathbf{t}_{k+1} - (s_k\mathbf{t}_{k+1} + \mathbf{b}_k))^T\mathbf{R}_{k+1}. \tag{17}$$

The alignment process outcome is illustrated in Figure 10b.

Data cleaning. The extracted trajectories have limitations from the data extraction method [50], including discontinuities, ruptures and jerky motions. To address this, we first clean the data by removing outliers (i.e., discontinuous segments), with a velocity threshold. Specifically, we eliminate trajectory points holding velocities greater than the 95th percentile of the overall trajectory velocity multiplied by a scaling factor. Subsequently, the trajectory is partitioned into sub-trajectories without outliers. Finally, we use Kalman filter on each chunk to reduce residual jerkiness and enhance overall smoothness.

B.3 Dataset creation pipeline

Motion tagging. We tune the parameters of our motion tagging method using the dataset introduced in [8]. This small dataset of 75 short clips includes annotated sequences of pure camera motion. For the character trajectory tagging, we extended this dataset by annotating human trajectories. We select parameters (i.e. mainly threshold values) that corresponds to the best classification metrics described in Section 5 of the main manuscript.

Caption generation. We show the prompt used for caption generation (see Section 3.2 of the main manuscript):

```
You act as a camera operator writing a technical script for camera
motion descriptions.

Given a rough outline of the camera motion and main character motion,
write the camera motion description according to the main character
motion.

The sentence should be short, and factual. Do not mention frame
indices.

# Examples
Outline: Total frames 209.
    [Camera motion] Between frames 0 and 154: boom top, Between
    frames 155 and 209: static.
    [Main character motion] Between frames 0 and 146: move up,
    Between frames 147 and 209: static.
Description: While the character climbs up, the camera follows them
with a boom top, and as soon as the character stops, it remains
static.
# End of examples

Outline: Total frames {CURRENT_NUM_FRAME}.
    [Camera motion] {CURRENT_CAMERA_DESCRIPTION}.
    [Main character motion] {CURRENT_CAMERA_DESCRIPTION}.
Description:
```

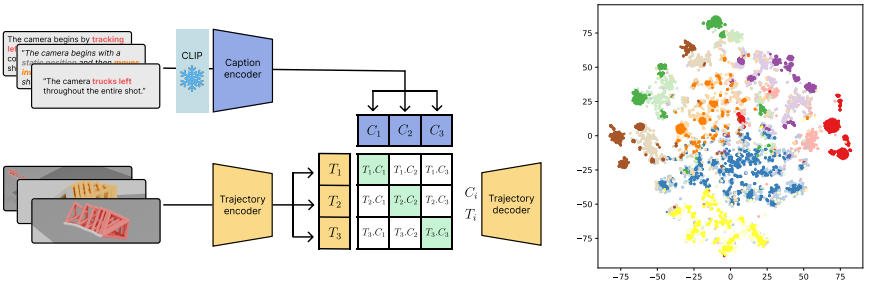
C Contrastive Language-Trajectory embedding (CLaTr)

Text-trajectory retrieval						Trajectory-text retrieval					
R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10 ↑	MedR ↓	R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10↑	MedR ↓
19.73	31.67	40.8	52.08	64.69	5.0	11.15	17.25	20.91	26.5	34.66	28.0

Table 3: CLaTr evaluation. We report the retrieval scores of CLaTr on the E.T. dataset.

We show in Figure 11a the overview of the CLaTr framework as described in Section 4.2 of the main manuscript.

Implementation details. We train CLaTr with a batch size of 32 using the AdamW optimizer with a learning rate of $1e-5$. The set the weight of the reconstruction loss at 1.0, of the latent loss at $1.0e-5$, of the KL loss at $1.0e-5$,



(a) Overview of CLaTr framework. CLaTr projects both text and camera trajectories into a common latent space using encoders. Self-similarity is then computed, and a shared-weight decoder decodes both text and camera trajectory features back into a camera trajectory.

(b) t-SNE visualization of CLaTr embedding of text (vivid colors) and trajectory (pastel colors). Each color corresponds to a K-Mean cluster of the text embedding.

and of the contrastive loss at 0.1. The model has 6 layers with a hidden dim of 256 and 4 attention heads. We use dropout of 0.1. Similar to DIRECTOR, we set the default temporal input size to 300 and use masking to handle inputs with fewer than 300 frames. We represent the camera trajectory with the 6D continuous representation for rotation [54] combined with the 3D translation component.

CLaTr Evaluation. Table 3 presents standard retrieval performance measures from [14,35]. Recall at rank k (R@k) indicates the percentage of times the correct caption is within the top k results (higher is better). Median rank (MedR) is also reported, where lower values are better.

As shown in Table 3, text-to-trajectory metrics outperform trajectory-to-text metrics. This may be because text descriptions are more ambiguous and varied in describing trajectories, making it easier to match a text description to a unique trajectory than to match a trajectory to a specific description among many possibilities.

CLaTr embedding. We show in Figure 11b a t-SNE visualization of CLaTr text (vivid colors) and trajectory (pastel colors) embeddings. We applied K-Means clustering to the text embeddings and visualized the corresponding clusters on the trajectory embeddings to assess the consistency of the joint embedding. Notably, we find that text clusters are preserved in the trajectory space, with vivid and pastel clusters overlapping, indicating a robust alignment between text and trajectory representations.