Seeing without Pixels: Perception from Camera Trajectories

Supplementary Material

1. Supplementary Video

We invite readers to view the supplementary video available at https://sites.google.com/view/seeing-without-pixels for a visual demonstration of our work's overview and additional qualitative examples. The video animates the static trajectories presented in the paper (*e.g.*, Fig. 1), offering a clearer view of the motion signatures. It also visualizes the learned embedding space, demonstrating how CamFormer clusters semantically similar neighbors. Furthermore, we provide qualitative examples of successful text retrieval (in both egocentric and exocentric domains) and demonstrate CamFormer's emergent capabilities, such as repetitive action counting and text-based trajectory retrieval, alongside a visual analysis of failure modes.

2. Experimental Setup

2.1. Task Setup

2.1.1. Text Retrieval (5-way MCQ)

To evaluate cross-modal alignment between the camera trajectory and text modality, we formulate text retrieval as a 5-way MCQ task, mitigating the inherent ambiguity of open-ended retrieval. Given a trajectory query, the model must select the correct text description from five options based on feature similarity, using the pre-trained CamFormer as a frozen feature extractor.

Negative Sampling Strategy. To ensure a rigorous benchmark, we curate distractor options to have primary action verbs that do not overlap with the ground truth. Furthermore, we adopt a hard negative sampling strategy: distractor options are sourced from narrations within the same continuous video take (for egocentric datasets) or from captions with the same YouTube ID (for exocentric datasets), forcing the model to distinguish between temporally or thematically adjacent actions.

Evaluation Splits. This task serves as our primary testbed across Ego-Exo4D, Nymeria, and DynPose-100K. We further exploit dataset-specific annotations to dissect our model's strengths versus the visual modality. On Ego-Exo4D, we analyze performance across "in-view" and "out-of-view" splits. On Nymeria, we break down results by narration type (legs, focus, body, hands).Qualitative MCQ examples are provided in Fig. 4 (main paper) and Fig. 8.

2.1.2. Other Evaluation Tasks

Proficiency Estimation on Ego-Exo4D. We utilize the dataset's skill-level annotations for the rock climbing and music activities. We focus exclusively on these two scenarios as they are the only ones that retain sufficient samples and a balanced class distribution after filtering for camera pose availability. To further mitigate imbalance, we formulate the task as a binary classification problem (expert vs. non-expert). We evaluate using an end-to-end fine-tuning protocol, comparing our pre-trained CamFormer initialization against a model trained from scratch.

Keystep Recognition & Localization on Ego-Exo4D. We utilize the dataset's established benchmark, which defines 278 fine-grained keystep labels. We adopt a frozen feature evaluation protocol for both tasks: we train a linear SVM for recognition and a dedicated localization network [46] for localization, both on top of our fixed trajectory embeddings.

Activity Classification on Ego-Exo4D. We evaluate coarse-grained understanding using the dataset's 8 high-level activity labels. For this task, we adopt an end-to-end fine-tuning protocol, training the trajectory encoder jointly with a linear classification head. We compare initializing from our pre-trained CamFormer against a model trained from scratch to measure the benefit of pre-training.

Scene Attribute Classification on DynPose-100K. We enrich DynPose-100K with semantic labels to enable a new binary scene attribute classification task. To do this, we designed 10 questions covering a diverse range of attributes (e.g., temporal, environmental, and social context) and utilized Gemini-2.5-Pro [8] to automatically label the videos. The prompts used for annotation are as follows:

1. **Day / Night:** Is the video filmed during the day or at night?
2. **Animal:** Does the video contain any animals?
3. **Text:** Does the video contain any visible written text?
4. **Urban / Rural:** Is the scene in the video urban or rural?
5. **Male / Female:** What is the gender of the people in the video?
6. **Food:** Does the video feature any food?
7. **Sports:** Is the video related to sports activities?
8. **Indoor / Outdoor:** Decide if the video is filmed indoors, outdoors, or if it is unclear.
9. **More than 1 people:** How many people are visible in the video?

10. **Walking:** Does the video show people walking?

From these new labels, we create a balanced 3,000-sample dataset for each attribute (with equal positive and negative examples) and train a linear SVM on our frozen camera trajectory features using an 80:20 train/test split.

Event Classification on FineGym. We evaluate on the 4 gymnasium event labels: floor exercise, balance beam, uneven bars, and vault. We adopt an end-to-end fine-tuning protocol, training the encoder jointly with a linear classification head.

Action Recognition on UCF101-Dynamic. To ensure a meaningful, motion-based evaluation, we curated this custom benchmark by quantitatively analyzing camera dynamics in UCF101. We selected the two most dynamic classes from each of four major action types, resulting in 8 classes: SkiJewandb sync ./anonymized/runs –project test-privacyt, SkateBoarding, Knitting, MoppingFloor, WalkingWithDog, Lunges, MilitaryParade, and SoccerPenalty. We evaluate using the same end-to-end fine-tuning protocol. For completeness, we also report results on the full 101-class UCF101 dataset (cf. Table 8).

2.2. Datasets

Pretraining Data. For the egocentric domain, we use Ego-Exo4D [21]. Adhering to the official splits, we obtain 159,186 training and 69,073 validation (trajectory, text) pairs, where text is human-annotated narrations provided by the dataset. Trajectory boundaries for these long, untrimmed videos are defined following [39, 52]. There are two camera trajectory sources: the original Aria glasses data (which we downsample to 20 FPS) and video-only estimations we obtain by running π^3 (5 FPS); a comparison is provided in Table 10. For the exocentric domain, we use DynPose-100K [54]. As no official split is available, we randomly split the dataset into 88,151 training and 10,452 validation (trajectory, text) pairs, where the text is video captions from Panda70M [7]. Unlike the egocentric data, these are short clips with fixed boundaries. There are two trajectory sources, both estimated from videos: the original dataset's provided ones (12 FPS) and the ViPE-provided [26] ones (30 FPS).

Downstream Data. For the egocentric domain, we evaluate on Ego-Exo4D and Nymeria. On Ego-Exo4D, for our designed text retrieval task, we draw samples from the official validation split; for all other tasks, we follow the dataset's official benchmark splits. For Nymeria, which shares the same Aria hardware as Ego-Exo4D, we use the entire set of data with available motion narrations as a zero-shot test set for text retrieval; the camera trajectory is also downsampled to 20FPS. For the exocentric domain, we use the original action labels from FineGym and UCF101. Since these datasets lack trajectories, we generate them using our pose estimation pipeline at 5 FPS. We generate three versions for UCF101 (using MegaSaM [38], ViPE [26], and π^3 [71]) and one version for FineGym (using π^3).

2.3. Implementation

When using hardware-estimated camera poses (*i.e.*, from Aria glasses [16]), we utilize the gravity direction information. To be specific, we compute the 3D gravity vector in our chosen relative reference frame and project it to d_{in} via a learned linear projection layer. This additional token is subsequently prepended to the input sequence before processing by the Transformer. This step is omitted when processing poses estimated from monocular videos. We train CamFormer using an AdamW optimizer with a learning rate of 1×10^{-4}, weight decay of 1×10^{-3}, and a batch size of 1024. CLIP loss temperature τ is 0.07. Training is conducted on 8 NVIDIA A100 GPUs.

Camera Pose Extraction. Given the computational expense of running multiple pose estimators, all pose estimations are performed at 5 FPS. For the particularly long, untrimmed videos in the Ego-Exo4D activity classification task (which can be several minutes), we further limit pose extraction to the center 4-second clip. For the egocentric setting, an alignment step is required. Our model is pre-trained on the Aria pose coordinate frame (x left, y up, z forward), so we apply a rigid transformation to convert all estimated poses from the standard OpenCV frame (x right, y down, z forward) before they are fed into our model.

Analysis of Contextualized Trajectory Encoding. We detail the experimental setup for the four settings in Fig. 6 of the main paper.

- *Global Label Tasks*. We fine-tune CamFormer end-to-end for these tasks. Our model accepts flexible input trajectory lengths (both in pre-training and finetuning), which allows us to systematically vary the input length during inference. (1) For Ego-Exo4D activity classification, we vary the input trajectory length from 1 to 16 seconds. (2) For FineGym event classification, since events have fixed segments, we vary the input ratio from 20% to 100% of the full event duration.
- *Localized Label Tasks*. For these tasks, we apply the pre-trained CamFormer as a frozen feature extractor and investigate extending the context outside the given segment window $[t_1, t_2]$. (3) For Ego-Exo4D text retrieval, we extend the atomic window $[t_1, t_2]$ by a total duration of 0, 2, 4, 6, or 8 seconds. We apply this symmetrically; for instance, a 2-second extension results in the input window $[t_1 - 1, t_2 + 1]$. (4) For Ego-Exo4D keystep recogni-

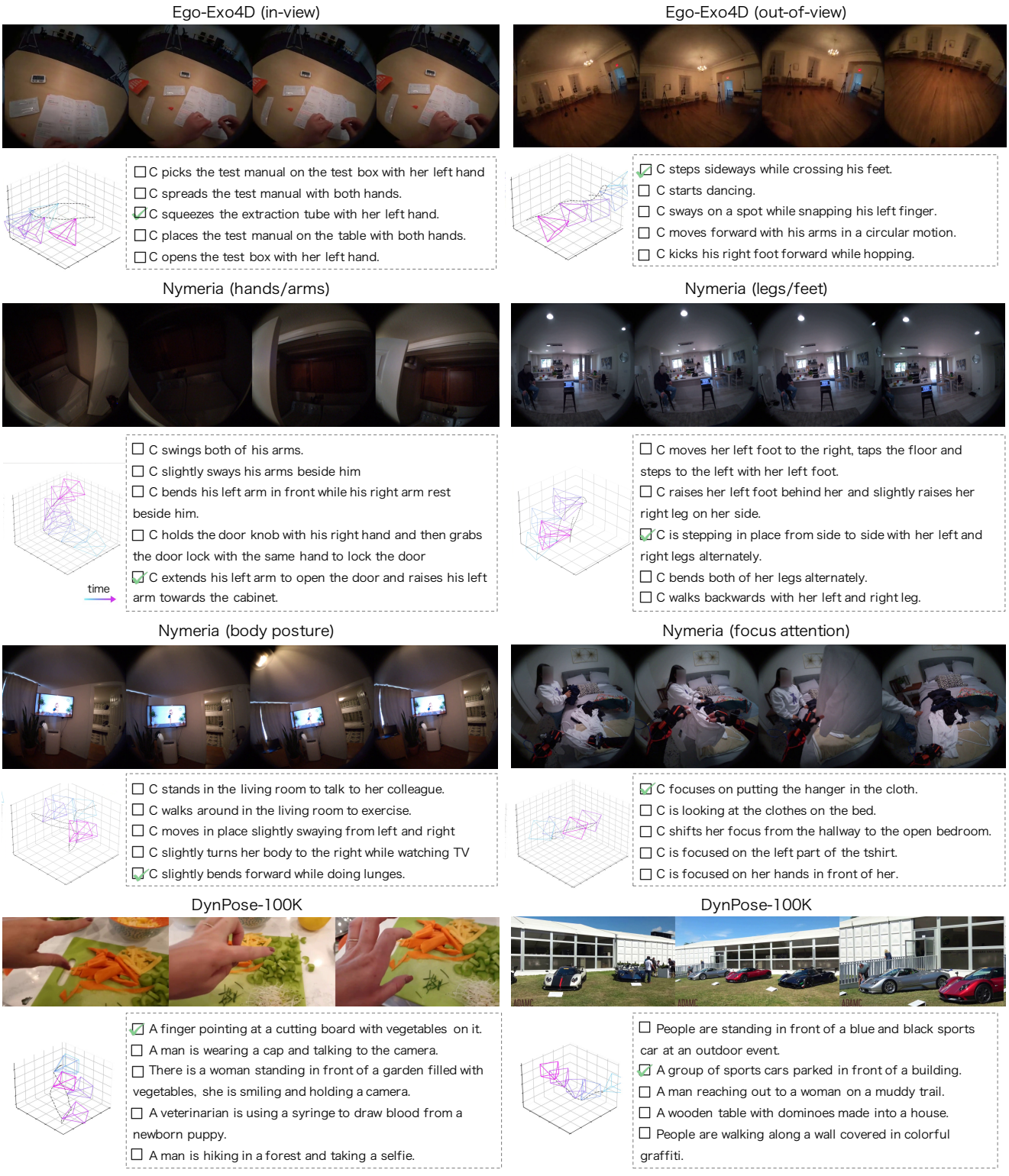

Figure 8. Qualitative Text Retrieval Results on Ego-Exo4D (top row), Nymeria (middle two rows) and DynPose-100K (bottom row). Note that CamFormer takes only the camera trajectory as input; corresponding video frames are shown solely for illustration. These examples further demonstrate that our model effectively captures the trajectory-semantic link across both egocentric and exocentric domains.

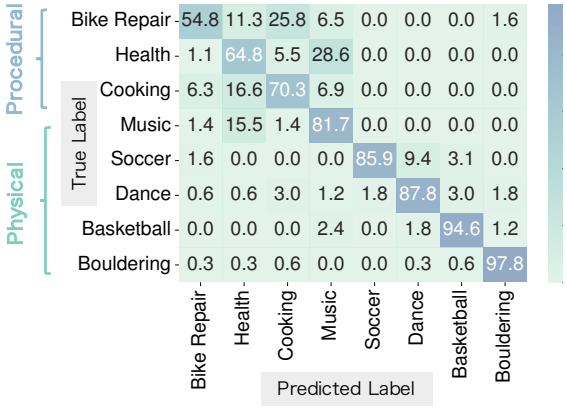

Figure 9. Confusion Matrix for CamFormer on Ego-Exo4D Activity Classification. CamFormer performs strongly on dynamic physical activities (*e.g.*, near-perfect for "bouldering"), while the main confusion occurs between the three procedural activities, which involve more subtle motion cues.

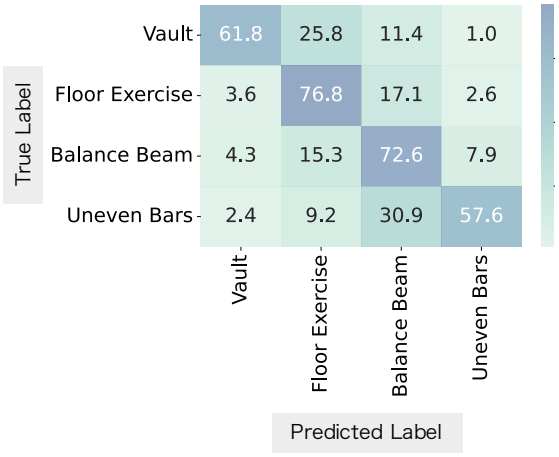

Figure 10. Confusion Matrix for CamFormer on FineGym Event Classification. The matrix details our model's performance on the 4 gymnasium activities.

tion, we expand the input trajectory window proportionally (100%-400% of the original duration).

2.4. Baselines

For the Gemini row in Table 2 of the main paper, we input 8 uniformly sampled video frames directly. The five candidate texts are randomly shuffled and assigned labels (A-E). Gemini-2.5-Pro [8] is queried with the prompt: *Which of the following descriptions best matches the video?*
For exocentric text retrieval (Table 4 of the main paper), we consider the following baselines:

- **Two-stage camera description baselines.** First, we prompt specialized LMMs (Qwen-VL-7B [17] or ShotVL-7B [40]) to *Describe the camera motion in this video.* Note that these models are trained to generate textual description of the camera motion in the associated video (*e.g.*, "zoom", "pan") and do not aim to describe the content of the video. Second, we feed this generated description into another LLM (we use Gemini-2.5-Flash [8]) and prompt it to answer the MCQ given the camera motion described in text form. The prompt is: *The following describes the motion and focus of a camera while filming a scene:[Generated Description]. Which of the following events or scene descriptions is most likely being filmed with this camera movement? [Option A-E].*
- **Zero-shot LMM baselines.** We evaluate the zero-shot capabilities of two strong LMMs for this task: Qwen3-VL-32B-Instruct [4] (open-source) and Gemini-3-Pro [8] (proprietary). To make the trajectory data compatible with these models, we render the 3D camera trajectories into video clips, consistent with our human-readable visualizations. We provide these trajectory videos as input alongside the following prompt: *The video features the trajectory of a camera. Answer the question about the video content based on the camera trajectory. Which of the following events or scene descriptions is most likely being filmed with this camera movement? [Option A-E].*
- **SFT baselines.** To assess the impact of domain-specific training, we implement two supervised fine-tuning (SFT) baselines using Qwen3-4B-Instruct [74] (an LLM) and Qwen3-VL-4B-Instruct [4] (an LMM). We perform SFT on the DynPose-100K training set to predict video descriptions. For the LMM, we input the rendered trajectory videos and the zero-shot prompt above. For the text-only LLM, we format the trajectory as a text list of 9D pose sequences with the following prompt: *You are given the relative camera trajectory of a video as a sequence of pose vectors. Each pose vector is 9D: 3D translation + 6D rotation. Camera trajectory: [value]. Task: Which of the following events or scene descriptions is most likely being filmed with this camera movement? [Option A-E].*

3. Results

3.1. Additional Results

Text Retrieval. Supplementing Table 2 of the main paper, Table 5 provides a detailed activity-level breakdown on Ego-Exo4D text retrieval, comparing our CamFormer embeddings with leading video encoder features (EgoVLPv2 [52]). The results allow us to clearly delineate the strengths of the two modalities. For the procedural activities (where visual cues are more critical), the video baseline maintains its lead. For the five physical activities, the camera trajectory modality is demonstrably stronger on its own. This performance is particularly effective in out-of-view (oov) settings, highlighting trajectory's unique value in scenarios where the visual signal is occluded or ambiguous. Finally, our CamFormer achieves the best overall re-

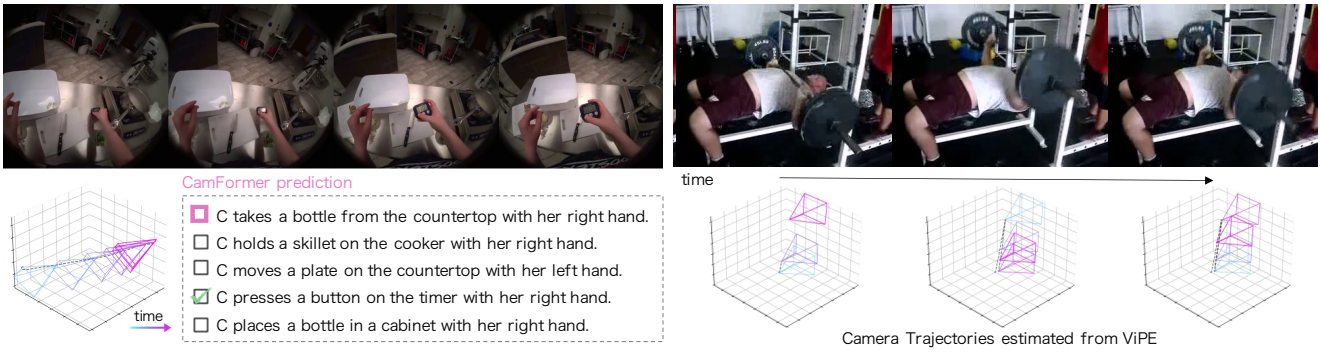

Figure 11. Failure Cases. Left: CamFormer struggles to distinguish actions with subtle motion patterns and often cannot capture specific noun semantics. Right: The pose source fails when the ViPE estimator mistakenly tracks object motion as camera motion (best viewed in Supp. video), highlighting a necessary area for development in pose estimation algorithms.

Table 5. Per-Activity Breakdown of Ego-Exo4D Text Retrieval Results (MCQ Accuracy). We compare the performance of CamFormer (trajectory features) and the best video baseline, EgoVLPv2 [52] (Ego-Exo4D), across all 8 activity scenarios. This detailed breakdown allows us to identify the relative strengths and weaknesses of the camera trajectory versus the video modality.

	Bike		Health		Cooking		Music		Soccer		Dance		Basketball		Bouldering		All
	iv	oov	iv	oov	iv	oov	iv	oov	iv	oov	iv	oov	iv	oov	iv	oov	
# Queries	500	205	500	416	500	500	500	176	500	500	282	500	500	500	500	500	7079
Video	**37.20**	**40.98**	**52.60**	**47.12**	**61.60**	**45.80**	29.40	21.02	24.40	21.20	25.89	15.40	50.80	38.00	59.40	29.60	38.40
Ours	28.80	28.78	24.40	26.44	49.80	39.40	**32.00**	**35.23**	**52.00**	**26.60**	**39.36**	**50.60**	**67.40**	**57.00**	**62.60**	**55.40**	**44.80**

sult, demonstrating the promising value of the camera trajectory in action understanding.

Qualitative Results. Supplementing Fig. 4 of the main paper, Fig. 8 presents additional qualitative results for the text retrieval task. These examples demonstrate that across diverse datasets and varying text descriptions, CamFormer successfully decodes the semantic information embedded in camera trajectories and accurately matches it with the corresponding text.

Proficiency Estimation. For physical activities, we find that camera trajectory is a powerful standalone signal for assessing skill levels (beginner/expert). Table 6 presents proficiency estimation results for the two physical activities. Our lightweight model successfully captures the motion signatures of expertise and outperforms the video baseline. This performance is further boosted by our pre-training strategy, which surpasses the train-from-scratch counterpart.

Keystep Recognition & Localization. For procedural activities, where motion patterns can be ambiguous, camera trajectory provides valuable complementary information. For the keystep recognition and localization tasks on these scenarios (Table 7), fusing trajectory with video features provides a consistent performance boost over the video-only baseline.

Table 6. Proficiency Estimation Accuracy (%) on Ego-Exo4D. The camera trajectory is a particularly strong modality for this physical task, outperforming the video baseline. Moreover, our pretraining is crucial, as initializing from CamFormer provides a boost over training from scratch.

Method	Modality	Pretrain?	Bouldering	Dancing
Majority	-	-	55.97	59.30
TimeSformer [5]	video	-	55.35	69.92
CamFormer	trajectory	✗	63.52	66.67
CamFormer	trajectory	✓	**65.41**	**70.73**

Table 7. Keystep Recognition and Localization Results on Ego-Exo4D. For these procedural tasks, where vision is a strong baseline, fusing trajectory and video features (denoted by *) consistently outperforms the video-only model, proving that camera trajectory provides an essential, non-redundant signal.

Method	Modality	Rec. Acc. (%)	Loc. Rank@1 IoU @0.3	Loc. Rank@1 IoU @0.5	Loc. Rank@5 IoU @0.3	Loc. Rank@5 IoU @0.5
Majority	-	3.52	-	-	-	-
EgoVLPv2 [52]	video	29.17	31.81	26.28	62.90	52.69
CamFormer	trajectory	14.07	20.29	15.67	47.23	38.09
CamFormer*	video+trajectory	**32.37**	**34.68**	**29.06**	**66.65**	**57.29**

Activity & Event Classification. The confusion matrix (Fig. 9) on Ego-Exo4D activity classification provides a detailed breakdown of the per-class accuracy results in Fig. 3 (c). CamFormer excels at recognizing dynamic physical ac-

Table 8. Comparing action recognition results on UCF101-Dynamic (left) and UCF101 (right) with various estimated camera poses. Echoing our egocentric analysis, the results confirm the benefits of our pre-training strategy. Across all pose estimators, the model initialized with our checkpoint pre-trained on DynPose-100K (✓) consistently outperforms its counterpart trained from scratch (✗).

Pose Source	UCF101-Dynamic			UCF101		
	Pretrain ✗	Pretrain ✓	Δ	Pretrain ✗	Pretrain ✓	Δ
MegaSaM [38]	66.67	69.15	+2.48	16.02	17.91	+1.89
ViPE [26]	64.18	68.16	+3.98	16.54	19.62	+3.08
π^3 [71]	61.69	64.18	+2.49	17.53	19.25	+1.72

Table 9. Ablation Study of Input Camera Trajectory Representation on Ego-Exo4D Text Retrieval. We compare various formulations, including the use of absolute vs. relative poses, the rotation format (no vs. 4D quaternion vs. 6D continuous), the specific reference frame used for calculating relative poses, and whether to include gravity direction information.

	Dim. (Tsl. + Rot.)	Acc. (%)
Absolute	3D	32.84
Absolute	3D + 4D	34.74
Absolute	3D + 6D	37.82
Relative (prev.)	3D + 4D	43.66
Relative (mid.)	3D + 4D	44.02
Relative (any)	3D + 4D	44.00
Relative (mid.)	3D + 6D	44.12
+ Gravity direction	3D + 6D	**44.81**

tivities, while confusion is heavily concentrated among the three procedural activities (where camera motion is subtle). For the exocentric domain, on FineGym event classification (Fig. 10), the model performs strongly on recognizing 'floor exercise', and the main confusion occurs between 'uneven bars' and 'balance beam'.

Action Recognition. Table 8 compares action recognition performance using our pre-trained CamFormer against the train-from-scratch baseline on UCF-Dynamic (our curated 8-class subset) and the full UCF101 dataset. The results show that initializing from CamFormer yields better performance than the train-from-scratch baseline across all three pose sources and on both settings. The largest performance gain is observed when using ViPE poses, as CamFormer was pre-trained with ViPE camera trajectories on DynPose-100K. Even with the other pose sources, the consistent gains observed across datasets demonstrate the robust generalization capability of our pre-training strategy.

Semantic Disentanglement in Embedding Space. To confirm that CamFormer learns generalized semantic representations rather than relying on geometric similarity, we analyze 5000 validation samples from Ego-Exo4D. We identify the top 1% most similar trajectory pairs in the raw input space, as measured by Euclidean distance after trajectory alignment. When mapped to the CamFormer embedding space, only 1.61% of these pairs remain in the top 1% of similarity, demonstrating that the model actively separates geometrically similar inputs. Crucially, this disentanglement is semantically driven: the pairs that remained close in our embedding space exhibited a 65.8% agreement in their activity labels, compared to only a 7.6% agreement for pairs that the model separated.

Table 10. Ablation Study of Pretraining Choices on Ego-Exo4D text Retrieval. We compare CamFormer performance using two different camera pose sources (high-fidelity Aria [16] vs. video-estimated π^3 [71]) and text encoder training modes (frozen vs. finetuned).

	Pose Source	Text Encoder	Acc. (%)
(a)	π^3 [71]	frozen	43.86
(b)	Aria [16]	finetune	45.42
(c)	Aria [16]	frozen	44.81

3.2. Ablation Study

Table 9 presents the ablation study on Ego-Exo4D text retrieval, where we compare various ways to represent the input camera trajectory. The results demonstrate that relative pose sequences are critical and greatly outperform absolute pose sequences, with the sequence midpoint being the optimal reference frame. Furthermore, the 6D continuous rotation representation [84] is preferred over the 4D quaternion, and encoding gravity direction provides a further performance boost.

Table 10 investigates our pretraining choices. First, regarding pose source: replacing high-fidelity Aria poses with video-estimated π^3 ones still yields comparable performance (43.86% vs. 44.81%). This is a promising result, indicating significant potential to scale up pre-training data using poses estimated from large collections of in-the-wild videos. Second, regarding the text encoder: while fine-tuning the CLIP encoder yields a marginal performance gain, it comes with a substantial computational cost. We therefore adopt the frozen text encoder for our final model to prioritize efficiency, though we posit that end-to-end fine-tuning may become more beneficial as data scale increases in the future.

3.3. Limitations

We acknowledge that obtaining high-quality camera poses initially incurs a computational cost, whether through multi-sensor hardware or video estimation algorithms. We view this, however, as a one-time, amortized process. Concurrent advances in hardware and the development of efficient algorithms are actively enriching existing video datasets with camera trajectories. This growing repository of pose-annotated data, like [85], provides the reusable, large-scale

foundation that our method can directly leverage.

Due to the inherent differences between egocentric and exocentric motion, we currently train a separate CamFormer for each domain under our unified framework. A promising avenue for future work is to build a single, unified trajectory encoder. This could be achieved by introducing an explicit conditional domain token that allows the unified encoder to effectively distinguish and interpret the recorder's intent across both camera perspectives.

The utility of the camera trajectory signal intrinsically depends on the correlation between camera motion and semantic content. While this signal is highly informative for physical activities, it is naturally weaker for fine-grained procedural tasks characterized by subtle, localized motions. Consequently, CamFormer can struggle to distinguish actions with kinematically similar patterns and inherently lacks the capacity to encode specific noun semantics. We present two failure modes of our investigation, as shown in Fig. 11. The left panel reveals that CamFormer struggles with subtle motion patterns ("press a button" in this case), confuses it with the adjacent action of "taking something from the countertop", and inherently fails to encode specific noun semantics. The right panel highlights an issue with the pose source: we observe cases where the estimator (*e.g.*, ViPE [26]) mistakenly correlates object motion with camera motion. This failure suggests that further algorithmic development in camera pose estimation is necessary to ensure robust semantic analysis.

Lastly, our investigation is scoped to everyday human activity videos; settings where camera motion is decoupled from content (*e.g.*, fixed cameras) fall outside scope.