TraCE: Trajectory-Aware Concepts for Explainable Video Understanding

Wooil Lee[1*] **Junpyo Hong**[1*] **Jongseo Lee**[1]
Tuomas Oikarinen[2] **Tsui-Wei Weng**[2] **Jinwoo Choi**[1†]
[1]Kyung Hee University [2]University of California San Diego
{lwi2765,intpboy,jong980812,jinwoochoi}@khu.ac.kr
{toikarinen,lweng}@ucsd.edu

Abstract

Video understanding models often rely on motion, yet their explanations can obscure which motion sources support a prediction. We identify *motion-source under-specification* as a limitation of existing video XAI: region-centric explanations can be source-ambiguous, while motion vocabularies restricted to single-person pose sequences are source-incomplete. We propose TraCE—Trajectory-Aware Concepts for Explainable Video Understanding—a concept-based framework that discovers *source-flexible* motion concepts by clustering optical-flow trajectories from training videos. These concepts represent dynamic evidence from people, objects, and interactions, while object and scene concepts account for static context. The resulting typed trajectory/object/scene interface supports class prediction for action recognition and question-conditioned concept matching for video question answering (VQA). We evaluate TraCE across five random seeds on six benchmarks: NTU RGB+D Mutual, Chiral SSv2, UCF-101, KTH, TOMATO, and STAR. Relative to non-interpretable baselines, TraCE maintains competitive task performance, with an average accuracy drop of 1.0 points for action recognition and a nominal gain of 0.4 points for VQA. Motion perturbation and concept-removal tests provide complementary evidence of directional semantic alignment and decision faithfulness, while sample-level interventions illustrate targeted model debugging. Our project page is available at https://leewooil.github.io/TraCE_project_page/.

1 Introduction

Motion is a central source of evidence in video understanding. Many events that share similar objects and scenes can differ primarily in how entities move over time [20, 51, 55]. However, modern video understanding models [6, 8, 32, 60, 63] remain largely opaque: their predictions do not reveal whether they rely on dynamic motion, static context, or spurious correlations [2, 14]. Reliable deployment therefore requires explainable video understanding methods that expose the evidence behind each prediction, including whether and how motion contributes. This need is not tied to a particular output format: whether a model predicts an action label or answers a language question, explanations should reveal the dynamic and static evidence behind the decision.

When motion contributes, explaining it as a single factor is insufficient. Motion in videos is not monolithic: it can arise from actors, manipulated objects, or interactions among entities, each carrying different semantics [39, 61]. For example, in a soccer play, a player's leg swing, the ball trajectory, and a defender's movement correspond to distinct sources of motion. An explanation that fails to specify these sources suffers from *motion-source under-specification*: it may indicate that "motion" matters, but not which motion supports the prediction. This failure can take two forms, as

[*]Equal contribution.
[†]Corresponding author.

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

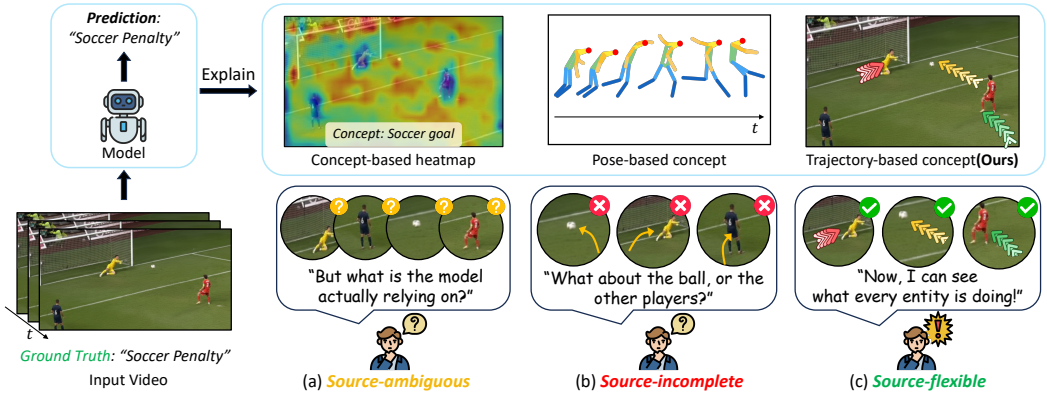


Figure 1: **Trajectory concepts make motion sources flexible.** Existing video XAI often underspecifies *which motion source* supports a prediction. (a) Heatmap-based explanations [7] can conflate actor motion, object motion, and static context (*source-ambiguous*). (b) Pose-based concepts [26] cover one person's joints but omit object motion and interactions (*source-incomplete*). (c) In contrast, TraCE provides *source-flexible motion explanations* by representing dynamic evidence from people, objects, and interactions as trajectory concepts.

shown in Figure 1. An explanation may conflate multiple motion sources into an ambiguous visual cue (*source-ambiguous*), or omit important sources altogether (*source-incomplete*). In both cases, users cannot determine whether a model relies on action-relevant body dynamics, object trajectories, interactions, or static contextual shortcuts. Thus, explainable video understanding requires motion explanations that specify dynamic evidence by source and present it distinctly.

Despite this need, existing video XAI methods suffer from *motion-source under-specification*. Attribution-based and region-centric methods [19, 31, 59] explain predictions by producing spatio-temporal heatmaps, tubelets, or other localized visual evidence over input videos. While these outputs can indicate where influential evidence appears, they do not specify what type or source of evidence the model uses. A highlighted region may correspond to static appearance, object or scene context, actor motion, object motion, or their mixture. Thus, such explanations are *source-ambiguous*: they do not explicitly represent motion as interpretable concepts, let alone separate motion concepts according to their sources. For example, as shown in Figure 1 (a), highlighting a player and a ball cannot tell whether the model relies on the player's kicking motion, the ball trajectory, or static contextual cues. This forces users to infer motion evidence from frame-to-frame changes in the visualization, making the explanation underspecified and ambiguous.

Concept-level explanations can reduce this ambiguity by exposing prediction evidence as interpretable concepts rather than raw spatio-temporal regions. However, in video understanding, their reliability critically depends on how motion concepts are defined. Language-derived concepts are effective for objects and scenes, but fine-grained motion is often *tacit* and difficult to express in words [45]. Recent pose-centric concept methods represent motion with human pose or skeleton sequences [26, 27], making single-person body motion explicit and interpretable. This pose-centric formulation, however, is *source-incomplete*: its motion vocabulary is defined over visible human joints and cannot directly represent dynamic evidence beyond single-person pose, such as object trajectories or multi-agent interactions. As a result, motion from interacting agents or manipulated objects may be omitted, or only indirectly reflected through human-pose or static-context concepts. This calls for motion concepts that are not tied to predefined human joints, but can instead trace dynamic evidence across diverse moving entities.

To address motion-source under-specification, we propose Trajectory-Aware Concepts for Explainable Video Understanding (TraCE), a concept-based framework that treats visual trajectories as first-class explanatory concepts. Derived from optical-flow trajectories, these concepts form a *source-flexible vocabulary for dynamic evidence* from people, manipulated objects, and interactions. As illustrated in Figure 1, TraCE explains predictions through these trajectory-aware motion concepts, while using object and scene concepts to account for static context. Because these concepts are defined at the level of video evidence rather than task-specific labels, the same typed trajectory/object/scene interface can support both class prediction and question-conditioned answer scoring. This shifts the explanatory bottleneck from describing single-person body poses to representing dynamic visual evidence at the trajectory level. To our knowledge, TraCE is the first

concept-based video explanation framework that uses optical-flow trajectory concepts as a unified interface for explaining dynamic evidence across action recognition and VQA.

We instantiate TraCE on two representative video understanding tasks: action recognition and video question answering (VQA). Across six benchmarks—NTU RGB+D Mutual [52], Chiral SSv2 [5, 18], UCF-101 [57], KTH [49], TOMATO [53], and STAR [67]—we evaluate single-person body motion, multi-person interactions, object-centric dynamics, temporally directional cues, and language-grounded reasoning. For VQA, we use native VQA benchmarks together with multiple-choice VQA reformulations of action-recognition datasets, allowing us to test whether the same typed interface supports both class prediction and question-conditioned answer scoring. Spatial grounding analyses and user studies support sample-level alignment with motion sources and human-perceived interpretability, respectively. TraCE maintains competitive task performance, while its trajectory-only variant retains predictive value under object-category shift and shows modest cross-domain transfer. Motion perturbation tests and concept removal provide complementary evidence of directional semantic alignment and decision faithfulness, while sample-level interventions illustrate targeted model debugging.

To summarize, we make the following three contributions.

- We identify *motion-source under-specification* as a limitation of existing video explanations and introduce *trajectory concepts* as a source-flexible representation of dynamic evidence from people, objects, and interactions.
- We propose TraCE, a concept-based framework that automatically discovers motion concepts by clustering optical-flow trajectories and complements them with object and scene concepts, without requiring manual concept annotation. The resulting typed interface supports sparse classification for action recognition and question-conditioned concept matching for VQA.
- Five-seed evaluations across six benchmarks show competitive task performance, with an average accuracy drop of 1.0 points for action recognition and a nominal gain of 0.4 points for VQA relative to non-interpretable baselines. Spatial grounding analyses and user studies support sample-level spatial alignment and interpretability, while perturbation and intervention tests provide complementary evidence of directional semantic alignment and decision faithfulness.

2 Related Work

Video understanding and dynamic evidence. Modern video understanding models span spatio-temporal action recognizers [8, 10, 17, 25, 28, 55, 60] and video-language models for language-grounded reasoning [6, 32, 63]. Yet their predictions can depend on entangled dynamic motion and static context [29]. Recent action-scene or spatial-temporal disentanglement methods address this issue for robustness or hallucination reduction [2, 3, 4], but do not expose dynamic evidence as concepts or specify which motion source supports an individual prediction. TraCE instead explains dynamic evidence with trajectory concepts and static context with object and scene concepts.

Region-centric and post-hoc concept explanations in video XAI. A first form of motion-source under-specification arises in region-centric video XAI. Saliency, perturbation, tube, and supervoxel methods localize where evidence appears [19, 31, 59], but remain *source-ambiguous*: a highlighted region may mix actor motion, object motion, static appearance, and scene context. Post-hoc concept methods [21, 24, 46] discover or probe recurring spatio-temporal concepts, but their concepts remain *untyped* rather than reusable motion concepts. TraCE explains dynamic evidence at the trajectory level, avoiding source-ambiguous regions and generic spatio-temporal concepts.

Concept bottlenecks and pose-centric motion concepts. Concept bottleneck models explain predictions through human-interpretable concepts [23, 40, 44, 58, 69, 70, 71], but video CBMs depend critically on how motion concepts are defined. Language-derived concepts struggle with fine-grained motion, while pose-centric methods [26, 27] restrict motion concepts to human pose or skeleton sequences. Such vocabularies are *source-incomplete*: they can explain single-person body motion but cannot directly represent object trajectories or multi-agent interactions. TraCE addresses this limitation with optical-flow trajectory concepts, providing a source-flexible motion vocabulary.

Explainable video-language understanding. Beyond action recognition, video-language works improve interpretability through temporal or spatial grounding [1, 15, 30, 33, 68] and textual or structural reasoning traces [11, 12, 34, 47]. These signals localize evidence or organize reasoning, but do not provide concept-level explanations of what dynamic evidence the model uses. In

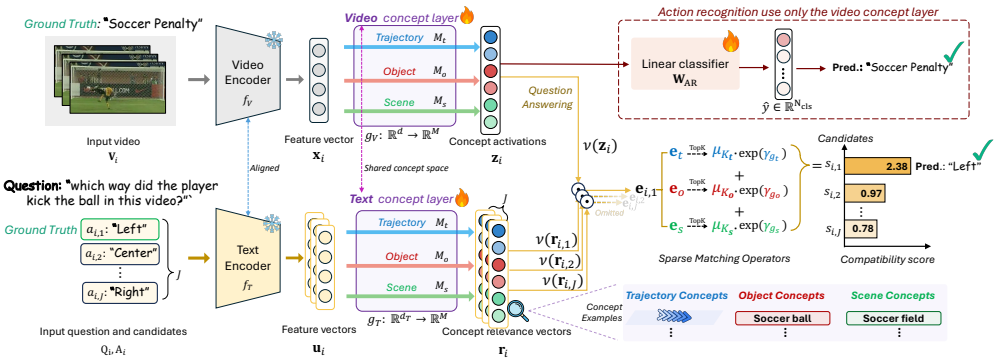


Figure 2: **Overview of TraCE.** TraCE represents videos with a typed trajectory/object/scene concept interface: trajectory concepts capture dynamic evidence, while object and scene concepts capture static context. Action recognition predicts directly from video-side concept activations, whereas VQA projects each question–answer pair into concept relevance and matches it with the video-side concepts. Thus, the same typed interface supports trajectory-based explanations for both class prediction and language-grounded reasoning.

particular, they do not expose reusable motion concepts or distinguish the motion source behind a prediction. TraCE represents dynamic evidence with trajectory concepts, yielding a task-general explanation interface for action recognition and VQA.

3 TraCE: A Trajectory-Aware Concept Interface

3.1 Design Overview

TraCE explains video understanding models through the trajectory-aware concept interface in Figure 2. The interface separates dynamic evidence from static context and exposes predictions through three typed concept groups: for a video $\mathbf{V}_i$, $\mathbf{z}_i = [\mathbf{z}_i^t; \mathbf{z}_i^o; \mathbf{z}_i^s]$, where $\mathbf{z}_i^t$ denotes trajectory concept activations, and $\mathbf{z}_i^o, \mathbf{z}_i^s$ denote object and scene concept activations. The design follows three principles.

D1. Visual trajectories as motion concepts. TraCE treats *visual trajectories as first-class explanatory concepts*. Unlike pose-centric motion concepts tied to predefined human joints, trajectory concepts are derived from optical-flow trajectories and capture visible entity-level dynamics from actors, manipulated objects, and interactions among multiple entities. This yields a source-flexible motion vocabulary without manual motion annotation or natural-language motion descriptions.

D2. A typed interface for dynamic and static evidence. As illustrated in Figure 2, trajectory concepts explain dynamic evidence, while object and scene concepts account for static context. The exact concept vocabularies are discovered separately from each downstream dataset's training videos, but the typed interface—trajectory, object, and scene—remains fixed. This allows TraCE to adapt its concept vocabulary to each dataset while preserving a consistent explanation structure.

D3. Task-general consumption of the concept interface. The typed video-side concept interface supports different video understanding tasks. For action recognition, a sparse classifier predicts the action label directly from $\mathbf{z}_i$. For VQA, each question–answer pair is projected into the same concept space as a question-conditioned concept relevance vector and matched with $\mathbf{z}_i$. Thus, VQA prediction becomes concept-level: the model answers by matching concepts present in the video with concepts required by the candidate answer. Importantly, *trajectory concepts do not require language names*. This is crucial for VQA: although trajectory concepts are visual motion clusters, answer supervision can still teach the text branch which concepts are needed for a candidate answer. Object and scene relevance, in contrast, can be supervised using language concept names.

In the following subsections, we present task-agnostic concept discovery (Section 3.2), concept bottleneck learning (Section 3.3), and task instantiations (Section 3.4).

3.2 Task-Agnostic Concept Vocabulary Discovery

For each downstream dataset, TraCE discovers concept vocabularies from its training videos, independently of whether the final task is action recognition or VQA. These vocabularies consist of trajectory concepts $\mathbb{C}^t$ for entity-level dynamic evidence, and object and scene concepts $\mathbb{C}^o, \mathbb{C}^s$ for static context. All concepts are discovered automatically without manual concept annotation.

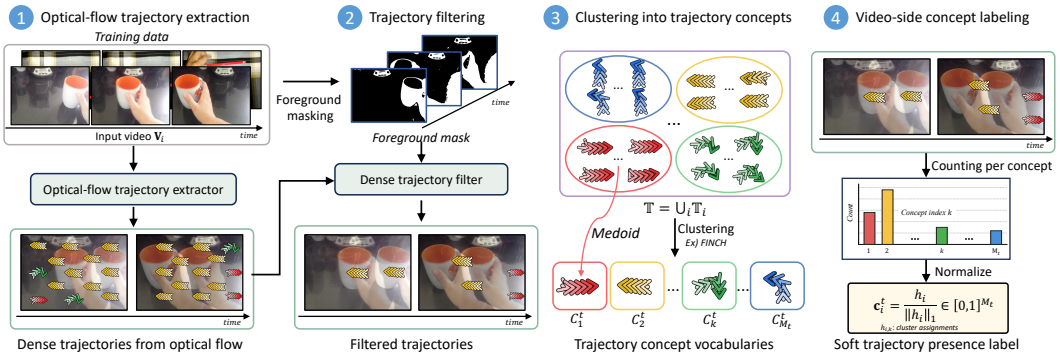


Figure 3: **Trajectory concept discovery and labeling.** TraCE extracts optical-flow trajectories from training videos, filters weak or noisy motion, and clusters the remaining trajectories into recurring motion patterns. Each cluster defines a trajectory concept, and each video is labeled by the normalized frequency of its trajectory assignments. This yields a *source-flexible motion vocabulary* for people, objects, and interactions without pose annotations or language descriptions of motion.

3.2.1 Trajectory Concepts

Trajectory generation. Given a video $\mathbf{V}_i$, we estimate dense optical flow between consecutive frames using a pretrained flow estimator [64]. We divide the video into N_s temporal segments and extract an L-frame clip around each segment center. For one segment, let $\Delta\mathbf{p}_r$ denote the optical flow from frame r to $r+1$. Starting from a seed pixel $\mathbf{p}_0 \in \mathbb{R}^2$, we trace a visual trajectory by accumulating flow as illustrated in Figure 3,

$$\Delta\mathbf{p}_r = \mathbf{p}_{r+1} - \mathbf{p}_r, \qquad \tau(\mathbf{p}_0) = [\Delta\mathbf{p}_0, \ldots, \Delta\mathbf{p}_{L-2}] \in \mathbb{R}^{2(L-1)}. \tag{1}$$

Applying this to all segments yields raw trajectories. We retain foreground trajectories with large motion magnitude, producing a filtered trajectory set $\mathbb{T}_i$ for video $\mathbf{V}_i$. Details of interpolation, foreground masking, and trajectory filtering are provided in Appendix A.5.1.

Trajectory concept discovery and labeling. We aggregate trajectories over the training set, $\mathbb{T} = \bigcup_i \mathbb{T}_i$, and cluster them with a clustering algorithm. TraCE is clustering-agnostic; we use FINCH [48] in our experiments. Let $\pi : \mathbb{T} \to \{1, \ldots, M_t\}$ denote the cluster assignment. The k-th trajectory concept is

$$C_k^t = \{\tau \in \mathbb{T} \mid \pi(\tau) = k\}, \qquad \mathbb{C}^t = \{C_k^t\}_{k=1}^{M_t}. \tag{2}$$

Each concept captures a recurring trajectory-level motion pattern shared across videos. For video $\mathbf{V}_i$, we count cluster assignments $h_{i,k} = \sum_{\tau \in \mathbb{T}_i} \mathbb{1}[\pi(\tau) = k]$ and form a soft trajectory presence label $\mathbf{c}_i^t = \frac{\mathbf{h}_i}{\|\mathbf{h}_i\|_1} \in [0,1]^{M_t}$. Thus, $c_{i,k}^t$ indicates the relative frequency of trajectory concept k in $\mathbf{V}_i$.

3.2.2 Object and Scene Concepts

To account for static context, we use object and scene concepts. Unlike prior class-name prompting [26], we query training videos with a Video-LLM [6], so the vocabularies reflect visible evidence rather than action-category priors. After filtering long or redundant phrases, we obtain $\mathbb{C}^o$ and $\mathbb{C}^s$. We assign soft pseudo-labels by video-text similarity using a pretrained vision-language encoder [63]. Additional details on prompts, filtering, and score normalization are provided in Appendix A.5.2.

3.3 Concept Bottleneck Learning

Given a video $\mathbf{V}_i$, a frozen video encoder f_V extracts $\mathbf{x}_i = f_V(\mathbf{V}_i) \in \mathbb{R}^d$. We define a video-side concept layer $g_V : \mathbb{R}^d \to \mathbb{R}^M$, where $M = M_t + M_o + M_s$ is the total number of trajectory, object, and scene concepts. Then we compute raw concept scores $\mathbf{a}_i = g_V(\mathbf{x}_i) = [\mathbf{a}_i^t; \mathbf{a}_i^o; \mathbf{a}_i^s]$ and activations $\mathbf{z}_i = [\sigma(\mathbf{a}_i^t); \mathbf{a}_i^o; \mathbf{a}_i^s]$. The sigmoid is applied only to trajectory concepts, which are trained with soft-label binary cross-entropy L_t using the soft trajectory label $\mathbf{c}_i^t$. Object and scene branches are trained with cosine-cubed losses L_o, L_s using pseudo-labels $\mathbf{c}_i^o, \mathbf{c}_i^s$, following label-free concept bottleneck models [26, 40]. The video-side concept learning objective is

$$L_{\text{concept}} = \lambda_t L_t + \lambda_o L_o + \lambda_s L_s. \tag{3}$$

This stage initializes the video-side concept interface. For action recognition, we freeze the learned video-side concept layer and train only the sparse classifier. For VQA, we freeze the video encoder but continue updating the video-side concept layer under concept supervision, while learning the text-side concept projector and answer-scoring parameters.

3.4 Task Instantiations

For each downstream dataset, `TraCE` discovers its own trajectory, object, and scene concept vocabularies from training videos. Thus, the exact concepts are task-specific, while the typed interface $\mathbf{z}_i = [\mathbf{z}_i^t; \mathbf{z}_i^o; \mathbf{z}_i^s]$ is shared, as shown in Figure 2. Action recognition consumes this interface directly with a sparse classifier, while VQA matches it with question-conditioned concept relevance.

3.4.1 Action Recognition

For action recognition, each sample is $(\mathbf{V}_i, y_i)$, where $y_i \in \{1, \ldots, N_{\text{cls}}\}$ is the action label. We freeze the video-side concept layer and train a sparse linear classifier on $\mathbf{z}_i$: $\hat{\mathbf{y}}_i = \text{softmax}(\mathbf{W}_{\text{AR}}\mathbf{z}_i)$. We optimize cross-entropy with elastic-net regularization [66] on $\mathbf{W}_{\text{AR}}$, encouraging each prediction to rely on a compact set of trajectory, object, and scene concepts.

3.4.2 Video Question Answering via Concept Matching

For VQA, each sample is $(\mathbf{V}_i, Q_i, A_i, y_i)$, where Q_i is a question, $A_i = \{a_{i1}, \ldots, a_{iJ}\}$ is the set of answer candidates, and y_i is the index of the correct answer.

Our VQA instantiation builds on the concept-space matching design of prior work [54], but adapts it to video understanding through *a typed trajectory/object/scene interface* and *a name-free trajectory branch*.

Text-side trajectory relevance without motion names. The key challenge is that trajectory concepts are visual motion clusters *without natural-language names*. Object and scene relevance can be supervised by concept-name similarity, but trajectory relevance cannot. We therefore learn text-side trajectory relevance through answer-supervised alignment with video-side trajectory evidence, rather than through text-concept similarity.

Question–answer projection. Let f_T be a pretrained text encoder and $g_T : \mathbb{R}^{d_T} \to \mathbb{R}^M$ a text-side concept projector. For each candidate answer, we compute question-conditioned concept relevance:

$$\mathbf{r}_{ij} = g_T(f_T([Q_i; a_{ij}])) = [\mathbf{r}_{ij}^t; \mathbf{r}_{ij}^o; \mathbf{r}_{ij}^s]. \tag{4}$$

Here, $\mathbf{z}_i$ represents video-side concepts, while $\mathbf{r}_{ij}$ represents concepts needed to support answer a_{ij}.

Concept-level answer scoring. We measure a concept-compatibility score of each answer as:

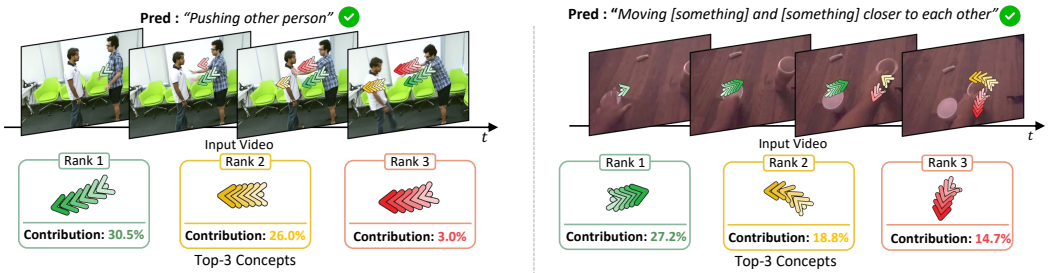
$$s_{ij} = \sum_{g \in \{t,o,s\}} \exp(\gamma_g)\mu_{K_g}(\mathbf{z}_i^g, \mathbf{r}_{ij}^g), \tag{5}$$

where μ_{K_g} sums the top-K_g entries of $\nu(\mathbf{z}) \odot \nu(\mathbf{r})$ ($\odot$ denotes element-wise multiplication), following prior works [54, 58], $\nu(\mathbf{x}) = \text{ReLU}(\mathbf{x})/(\|\text{ReLU}(\mathbf{x})\|_2 + \varepsilon)$, and γ_g is a learnable weight of the concept group g. The predicted answer is $a_{ij_i^{\text{pred}}}$, where $j_i^{\text{pred}} = \arg\max_j s_{ij}$.

VQA training objective. To preserve concept fidelity, we retain the video-side concept loss L_{concept}, computed over videos in each mini-batch. The answer loss is the cross-entropy over candidate scores, denoted L_{QA}. For object and scene concepts, we supervise text-side relevance with a candidate-averaged cosine loss L_{text}^g, $g \in \{o, s\}$, using text-concept similarity labels. For trajectory concepts, text-concept similarity is unavailable, so we align question-conditioned trajectory relevance with video-side trajectory evidence using an InfoNCE loss [41] L_{align}^t, where the correct answer is the positive and other candidates are negatives. The overall VQA objective is

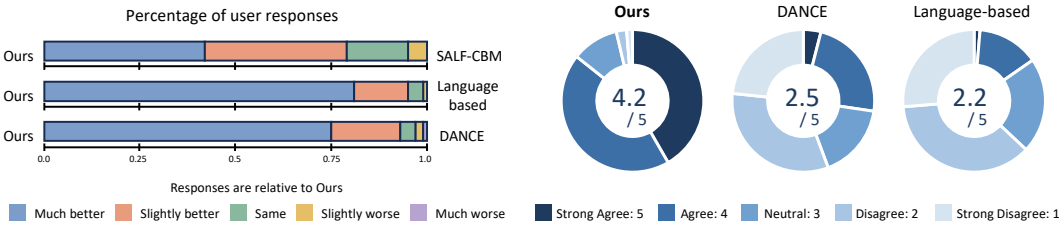
$$L_{\text{VQA}} = \lambda_{\text{QA}}L_{\text{QA}} + \lambda_{\text{concept}}L_{\text{concept}} + \lambda_{\text{text}}(L_{\text{text}}^o + L_{\text{text}}^s) + \lambda_{\text{align}}L_{\text{align}}^t. \tag{6}$$

In summary, L_{concept} preserves the video-side concept semantics during VQA training, L_{text}^g trains text-side relevance for *named* object and scene concepts, and L_{align}^t connects *unnamed* trajectory concepts to question-conditioned answer relevance.



(a) Multi-agent Interaction (b) Object-centric Dynamics

Figure 4: **Sample-level explanation.** We visualize the top-3 contributing concepts by overlaying them on top-activated video regions. (a) Interaction dynamics across agents. (b) Object-centric motion with partial human cues. Trajectory concepts capture motion from multiple sources.



(a) Interpretability of TraCE (b) Interpretability of the trajectory concept

Figure 5: **Interpretability of `TraCE`.** (a) Pairwise preference study comparing `TraCE` explanations with SALF-CBM [7], a language-based motion-concept CBM, and DANCE [26]. Bars show responses relative to `TraCE`. (b) Concept-level study comparing trajectory concepts (`TraCE`), pose-sequence concepts (DANCE), and language-derived motion concepts. Participants rate how well each concept matches the motion shared by its top-activating video clips on a five-point Likert scale.

4 Experimental Results

We assess explanation interpretability and sample-level grounding (Section 4.1), task performance and generalization (Section 4.2), and concept semantics and decision faithfulness (Section 4.3). Dataset and implementation details are provided in Appendix C and Appendix B, respectively.

4.1 Source-Flexible Explanations with Trajectory Concepts

Concept contribution and visualization. For action recognition, the contribution of concept m to the predicted class $\hat{c}_i = \arg\max_c \hat{\mathbf{y}}_i[c]$ is $z_{i,m}\mathbf{W}_{\text{AR}}[\hat{c}_i, m]$, the product of its activation and classifier weight. For VQA, we use the compatibility score s_{ij}: each concept contributes through top-K matching between video-side evidence $z_i^g[m]$ and question-conditioned relevance $r_{ij}^g[m]$. We visualize trajectory concepts using representative cluster trajectories overlaid on top-activated clips with Grad-CAM [50], used *only to localize responses for visualization*.

`TraCE` provides source-flexible motion explanations. We quantify sample-level spatial grounding on object-disjoint Chiral SSv2 using object bounding box annotations from Something-Else dataset [38]. For this post-hoc analysis, we associate the top-contributing concepts with input trajectories through cluster assignment. As shown in Table 1, `TraCE` achieves superior coverage of annotated hand and manipulated-object boxes than DANCE. These results support spatial alignment with both entity types without fixed entity labels for trajectory concepts. Protocol details and additional concept-selection results are provided in Appendix G.1.

Table 1: **Motion-source grounding.** Top-10 concept support within annotated boxes (%).

Method	Hand	Object
DANCE [26]	42.3	7.6
`TraCE`	**64.5**	**63.5**

Complementary visualizations in Figure 4 show the top-3 contributing trajectory concepts for motion beyond single-person pose dynamics. For *pushing other person*, the selected concepts correspond to both the pushing motion of one person and the displacement of the other. For *moving [something] and [something] closer to each other*, they correspond to the objects' converging movements even though no full human body is visible.

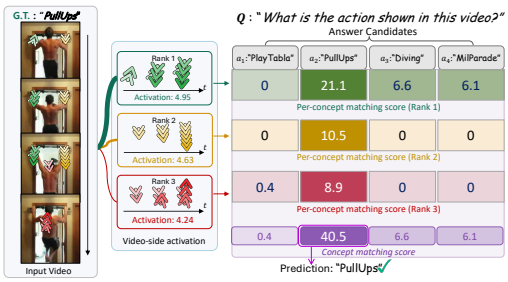


Figure 6: **Sample-level VQA explanation.** Rows show the top-3 video-side concepts and columns show answer candidates, with cells indicating concept–answer compatibility.

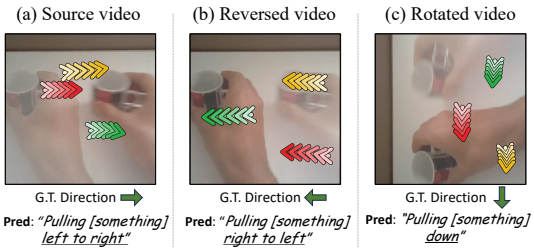


Figure 7: **Motion perturbation analysis.** Activated trajectory concepts track directional changes under reversal and rotation. Each panel overlays the first frame at low opacity with the last frame at full opacity.

Table 2: **Performance comparison on action recognition and VQA benchmarks.** We report mean ± sample standard deviation over five random seeds for Top-1 accuracy (%) using the same frozen backbone within each task, except for the single-run SpLiCE adaptation. A dash (–) indicates that the method is not applicable to the task. Task averages are computed over the corresponding dataset means.

Method	NTU RGB+D Mutual		Chiral SSv2		UCF-101		KTH	TOMATO	STAR	Avg.	
	AR	VQA	AR	VQA	AR	VQA	AR	VQA	VQA	AR	VQA
Baseline w/o interpretability	99.7±0.2	97.4±0.1	86.6±0.2	82.6±0.2	89.8±0.1	99.8±0.0	91.1±0.3	35.9±2.2	54.5±0.4	91.8	74.0
LF-CBM [40] w/ language concepts	96.9±0.3	–	77.6±0.5	–	88.3±0.2	–	89.1±0.7	–	–	88.0	–
DANCE [26]	97.1±0.3	–	81.7±0.3	–	89.0±0.2	–	91.3±0.4	–	–	89.8	–
SALF-CBM [7]	99.7±0.0	–	80.4±0.4	–	87.5±0.2	–	87.9±0.4	–	–	88.9	–
SpLiCE [9] adaptation	–	53.0	–	57.3	–	86.7	–	22.4	49.2	–	53.7
TraCE	97.4±0.1	97.1±0.3	84.9±0.4	82.6±0.6	89.0±0.3	99.7±0.1	91.8±0.8	37.0±1.7	55.4±1.2	90.8	74.4

Trajectory concepts and explanations are interpretable. We conduct two user studies on motion-focused subsets of NTU RGB+D [52] and Chiral SSv2 [5, 18] (Appendix E). First, participants compare TraCE explanations against DANCE [26], a language-based motion-concept CBM, and SALF-CBM [7]. As shown in Figure 5 (a), at least 93% of responses prefer TraCE over DANCE and the language-based baseline, and about 78% prefer TraCE over SALF-CBM. Second, participants rate how well a concept matches the motion shared by its top-4 activating clips. As shown in Figure 5 (b), trajectory concepts achieve the highest score (4.2/5), compared to DANCE pose concepts (2.5/5) and language-based concepts (2.2/5). These results indicate that users find trajectory concepts coherent and better aligned with the motion patterns in the clips.

Concept matching explains language-grounded predictions. Figure 6 shows how TraCE explains a VQA prediction by matching video-side concepts with question-conditioned concept relevance. The correct answer *PullUps* receives high scores across the top-3 trajectory concepts, which capture repeated vertical motion, while competing answers receive lower scores. To assess whether these concept-based explanations help users understand the model's answers, we conduct a blinded pairwise study with 50 participants. We control for answer correctness by including only examples for which both compared models predict the same correct answer. Participants prefer TraCE explanations over those based on language-derived motion concepts and pose-sequence concepts in 70.4% and 85.6% of comparisons, respectively, indicating greater perceived usefulness in the VQA setting. The full protocol and response distributions are provided in Appendix E.5.

4.2 Performance and Generalization of the Trajectory-Aware Interface

We next assess the predictive performance of the typed trajectory/object/scene interface across action recognition and VQA, including transfer under object-category and domain shifts.

TraCE preserves performance across video understanding tasks. In Table 2, we compare TraCE with non-interpretable baselines using the same frozen backbone within each task: VideoMAE [60] for action recognition and InternVideo2 [63] for VQA. Across five random seeds, TraCE remains close to the baseline on average, with a 1.0-point drop for action recognition and a nominal 0.4-point gain for VQA; given the reported variability, we do not interpret the VQA difference as a significant improvement. We also adapt SpLiCE [9] to question-conditioned answer scoring under the same frozen backbones. This single-run adaptation performs below the five-seed mean of TraCE on all five VQA benchmarks. Overall, the concept interface retains competitive task performance while providing concept-based explanations.

Trajectory concepts remain predictive across diverse motion sources. We compare TraCE with LF-CBM [40] using language-derived motion concepts, DANCE using pose-sequence con-

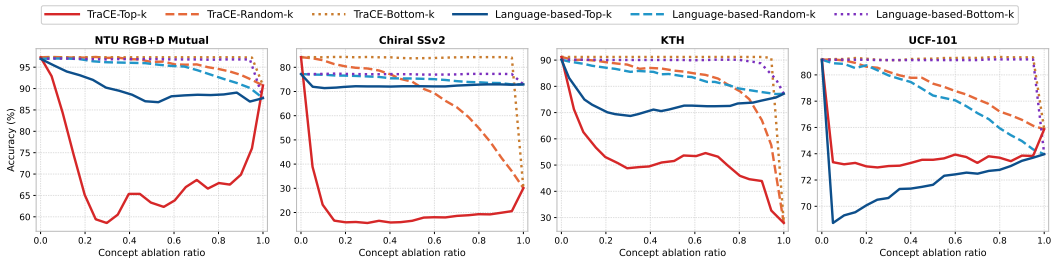


Figure 8: **Concept intervention curves show faithfulness of `TraCE`.** We deactivate concepts in descending (Top-k), random, and ascending (Bottom-k) order of contribution and report prediction accuracy as the removal ratio increases.

cepts [26], and SALF-CBM [7] on action recognition. Their original action-recognition formulations do not define question-conditioned answer scoring for VQA. Across benchmarks spanning single-person motion (KTH), multi-person interaction (NTU RGB+D Mutual), object-centric and direction-sensitive dynamics (Chiral SSv2), and general human actions (UCF-101), `TraCE` remains competitive among interpretable methods. It achieves the highest reported mean accuracy on KTH and Chiral SSv2, ties for the best on UCF-101, and remains competitive on the near-saturated NTU RGB+D Mutual split.

Trajectory concepts remain predictive under object-category and visual-domain shifts. We further evaluate whether the trajectory-only variant remains predictive under object-category and visual-domain shifts. Using Something-Else object annotations [38], we construct an object-disjoint Chiral SSv2 [5, 18] split where object categories do not overlap between training and test. As shown in Table 3, the trajectory-only variant of `TraCE` achieves 94.4%, outperforming the non-interpretable baseline [60] at 89.5%. We also train the concept pipeline on Chiral SSv2 and apply it to aligned TOMATO [53] examples without target-domain training. The trajectory-only variant achieves 43.1%, compared to 41.5% for the baseline. Together, these results suggest that trajectory concepts retain predictive motion cues under object category shift and show modest cross-domain transfer.

Table 3: **Generalization under object-category and visual-domain shifts.**

Method	Sth.-Else disjoint-obj.	SSv2→ TOMATO
Baseline w/o interpretability	89.5	41.5
`TraCE` w/o static concepts	94.4	43.1

4.3 Faithfulness of `TraCE`

We examine the directional semantics of activated trajectory concepts. Interpretability alone does not establish that concept explanations reflect model behavior. We therefore examine whether activated concepts retain their directional meaning and whether the model's predictions depend on them, before illustrating their use for debugging.

Trajectory concepts respond consistently to motion perturbations. We examine sensitivity to signed motion direction at both the prediction and concept levels on the directional-class subset of Chiral SSv2 [5, 18], whose labels specify motion directions. Temporal reversal swaps opposite directions, while a 90° rotation maps each direction to its rotated counterpart. As shown in Table 4, `TraCE` achieves 81.1% accuracy on original inputs, 79.2% consistency under temporal reversal, and 63.1% under rotation. To assess concept semantics directly, we assign each concept a fixed direction label from the net displacement of its training-cluster medoid. The highest-activated concept matches the ground-truth direction in 67.5% of held-out videos, compared with a 25% uniform four-way reference. Qualitative examples in Figure 7 show that activated trajectory concepts track directional changes under temporal reversal and rotation. Top-5 concept replacement and direction agreement are reported in Appendix G.2. These analyses support motion sensitivity and directional semantic alignment; decision dependence is tested below.

Table 4: **Prediction consistency under perturbations.**

Perturbation	Acc.
Original	81.1
Reverse	79.2
Rotation	63.1

Contribution rankings identify decision-relevant concepts. Beyond semantic alignment, we test whether contribution rankings identify concepts used for prediction. We progressively deactivate trajectory concepts in descending, random, or ascending contribution order. In Figure 8, top-ranked removal causes sharper accuracy drops than random or bottom-ranked removal on NTU RGB+D Mutual, Chiral SSv2, and KTH; the separation is weaker on UCF-101. The corresponding native-VQA analysis shows that removing the top 5% of trajectory concepts reduces accuracy by

Table 5: **Count-matched trajectory concept interventions.** We report accuracy, its change from the unablated model (ΔAcc.), and the correct-to-incorrect flip rate on direction-annotated subsets. Mean ± standard deviation is computed over 100 concept-subsampling draws. See Appendix G.3 for details.

	Chiral SSv2 (AR)			TOMATO (VQA)		
Removed concepts	Acc. (%)	ΔAcc.	Flip rate (%)	Acc. (%)	ΔAcc.	Flip rate (%)
None	81.1	–	–	46.4	–	–
GT direction	56.9±0.1	**−24.2**	**30.1±0.1**	23.9±0.4	**−22.5**	**70.5±0.5**
Opposite direction	82.2±0.1	+1.1	3.6±0.1	54.0±0.2	+7.6	30.8±0.0
Random non-GT	82.1±0.2	+1.0	3.7±0.2	45.4±1.3	−1.0	21.3±2.0

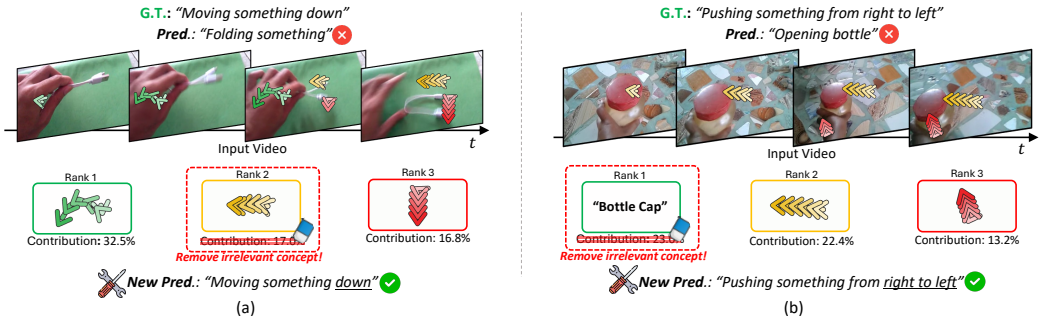


Figure 9: **Concept-level intervention.** (a) Deactivating a mismatched trajectory concept fixes a motion error. (b) Deactivating the spurious object concept "bottle cap" shifts the model from static context to motion evidence. These examples illustrate actionable concept-level debugging with TraCE.

11.9 points on TOMATO and 40.5 points on STAR, compared with 4.3 and 1.3 points under random removal (Table 24). These comparisons assess contribution-ranking faithfulness rather than semantic correctness by themselves.

Direction-targeted interventions link semantics to predictions. To connect directional semantics with decision dependence, we use the fixed direction labels above to remove equal numbers of ground-truth (GT)-direction, opposite-direction, or random non-GT concepts on the same directional class subset of Chiral SSv2 used in Table 4 and a direction-annotated subset of TOMATO. As shown in Table 5, GT-direction removal reduces AR accuracy by 24.2 points and flips 30.1% of initially correct predictions; neither control reduces accuracy, and both yield flip rates below 4%. For VQA, GT-direction removal produces the largest accuracy drop (22.5 points) and flip rate (70.5%). The VQA flip rate is descriptive because only 13 of the 28 samples are initially correct. These interventions provide model-internal evidence that direction-aligned trajectory concepts influence predictions in both tasks.

Concept intervention supports sample-level debugging. These concept-level interventions also enable targeted correction of individual errors. In Figure 9 (a), the model misclassifies *Moving [something] down* as *Folding [something]*, with a mismatched leftward-motion concept contributing to the prediction. Deactivating that concept corrects the prediction. In Figure 9 (b), the model predicts *Opening bottle* for a *Pushing [something] from right to left* video because the object concept "bottle cap" is highly influential. Removing this spurious object concept shifts the model from static context to motion evidence and corrects the prediction. TraCE exposes whether an error stems from misleading motion concepts or static context and enables targeted concept-level correction, illustrating how the concept interface can support model debugging.

5 Conclusions

In this work, we address *motion-source under-specification* in explainable video understanding. We propose TraCE, a concept-based framework that derives *source-flexible* motion concepts from optical-flow trajectories to capture motion from actors, objects, and interactions while separating dynamic evidence from static object and scene context. Across action recognition and video question answering, TraCE maintains competitive task performance while producing explanations that users find interpretable and that intervention analyses show to be faithful to model behavior. These results suggest that trajectory concepts offer a practical typed interface for explaining motion evidence in video understanding.

Acknowledgments and Disclosure of Funding

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (RS-2025-02216217, 50%) and (RS-2026-25616405, 25%). Additionally, it was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT) under grant RS-2024-00353131 (10%) and the Institute of Information & Communications Technology Planning & Evaluation (IITP)-ITRC (Information Technology Research Center) grant funded by the Korea government (MSIT) (IITP-2026-RS-2023-00259004, 10%). This research was also supported by the MSIT (Ministry of Science and ICT), Korea, under the National Program for Excellence in SW (2023-0-00042, 5%) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation) in 2026.

References

[1] Geo Ahn, Jiwook Han, Youngrae Kim, Joonseok Lee, and Jinwoo Choi. EVIDENT: Routing MLLM adaptation through entity-grounded visual evidence for cross-domain video temporal grounding. In *NeurIPS*, 2026. 3

[2] Geo Ahn, Inwoong Lee, Taeoh Kim, Minho Shim, Dongyoon Wee, and Jinwoo Choi. Why can't I open my drawer? Mitigating object-driven shortcuts in zero-shot compositional action recognition. In *ECCV*, 2026. 1, 3

[3] Kyungho Bae, Geo Ahn, Youngrae Kim, and Jinwoo Choi. DEVIAS: Learning disentangled video representations of action and scene for holistic video understanding. In *ECCV*, 2024. 3

[4] Kyungho Bae, Jinhyung Kim, Sihaeng Lee, Soonyoung Lee, Gunhee Lee, and Jinwoo Choi. MASH-VLM: Mitigating action-scene hallucination in Video-LLMs through disentangled spatial-temporal representations. In *CVPR*, 2025. 3

[5] Piyush Nitin Bagad and Andrew Zisserman. Chirality in action: Time-aware video representation learning by latent straightening. In *NeurIPS*, 2025. 3, 8, 9, 20, 21, 24, 25, 26, 30, 38, 48

[6] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. 1, 3, 5, 19, 31, 32, 49

[7] Itay Benou and Tammy Riklin Raviv. Show and tell: Visually explainable deep neural nets via spatially-aware concept bottleneck models. In *CVPR*, 2025. 2, 7, 8, 9, 25, 26, 28

[8] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 1, 3

[9] Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P. Calmon, and Himabindu Lakkaraju. Interpreting CLIP with sparse linear concept embeddings (SpLiCE). In *NeurIPS*, 2024. 8, 36

[10] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 3

[11] Junwen Chen and Yu Kong. Explainable video entailment with grounded visual evidence. In *ICCV*, 2021. 3

[12] Tieyuan Chen, Huabin Liu, Tianyao He, Yihang Chen, Chaofan Gan, Xiao Ma, Cheng Zhong, Yang Zhang, Yingxue Wang, Hui Lin, et al. MECD: Unlocking multi-event causal discovery in video reasoning. In *NeurIPS*, 2024. 3

[13] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. VideoLLaMA 2: Advancing spatial-temporal modeling and audio understanding in Video-LLMs. *arXiv preprint arXiv:2406.07476*, 2024. URL https://arxiv.org/abs/2406.07476. 31, 32

[14] Jinwoo Choi, Chen Gao, Joseph CE Messou, and Jia-Bin Huang. Why can't i dance in the mall? learning to mitigate scene bias in action recognition. In *NeurIPS*, 2019. 1

[15] Andong Deng, Tongjia Chen, Shoubin Yu, Taojiannan Yang, Lincoln Spencer, Yapeng Tian, Ajmal Saeed Mian, Mohit Bansal, and Chen Chen. Motion-grounded video reasoning: Understanding and perceiving motion at pixel level. In *CVPR*, 2025. 3

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 16, 21

[17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 3

[18] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 3, 8, 9, 20, 21, 24, 25, 26, 30, 38, 48

[19] Thomas Hartley, Kirill Sidorov, Christopher Willis, and David Marshall. SWAG-V: explanations for video using superpixels weighted by average gradients. In *WACV*, 2022. 2, 3

[20] De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *CVPR*, 2018. 1

[21] Ying Ji, Yu Wang, and Jien Kato. Spatial-temporal concept based explanation of 3D ConvNets. In *CVPR*, 2023. 3

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 22

[23] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *ICML*, 2020. 3

[24] Matthew Kowal, Achal Dave, Rares Ambrus, Adrien Gaidon, Konstantinos G Derpanis, and Pavel Tokmakov. Understanding video transformers via universal concept discovery. In *CVPR*, 2024. 3

[25] Dongho Lee, Jongseo Lee, and Jinwoo Choi. Cast: Cross-attention in space and time for video action recognition. In *NeurIPS*, 2023. 3

[26] Jongseo Lee, Wooil Lee, Gyeong-Moon Park, Seong Tae Kim, and Jinwoo Choi. Disentangled concepts speak louder than words: Explainable video action recognition. In *NeurIPS*, 2025. 2, 3, 5, 7, 8, 9, 17, 19, 22, 23, 25, 26, 28, 29, 36, 38

[27] Jongseo Lee, Wooil Lee, Gyeong-Moon Park, Seong Tae Kim, and Jinwoo Choi. PCBEAR: Pose concept bottleneck for explainable action recognition. In *CVPRW*, 2025. 2, 3

[28] Jongseo Lee, Joohyun Chang, Dongho Lee, and Jinwoo Choi. CA^2ST: Cross-attention in audio, space, and time for holistic video recognition. *TPAMI*, 2026. 3

[29] Jongseo Lee, Hyuntak Lee, Sunghun Kim, Sooa Kim, Jihoon Chung, and Jinwoo Choi. Which way did it move? Diagnosing and overcoming directional motion blindness in Video-LLMs. In *NeurIPS*, 2026. 3

[30] Yicong Li, Xiang Wang, Junbin Xiao, Wei Ji, and Tat-Seng Chua. Invariant grounding for video question answering. In *CVPR*, 2022. 3

[31] Zhenqiang Li, Weimin Wang, Zuoyue Li, Yifei Huang, and Yoichi Sato. Towards visually explaining video understanding networks with perturbation. In *WACV*, 2021. 2, 3

[32] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In *EMNLP*, 2024. 1, 3, 31, 32

[33] Huabin Liu, Xiao Ma, Cheng Zhong, Yang Zhang, and Weiyao Lin. TimeCraft: Navigate weakly-supervised temporal grounded video question answering via bi-directional reasoning. In *ECCV*, 2024. 3

[34] Huabin Liu, Filip Ilievski, and Cees GM Snoek. Commonsense video question answering through video-grounded entailment tree reasoning. In *CVPR*, 2025. 3

[35] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 20

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 20

[37] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–298. University of California press, 1967. 30, 31

[38] Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *CVPR*, 2020. 7, 9, 24, 38

[39] Etienne Meunier and Patrick Bouthemy. Segmenting the motion components of a video: A long-term unsupervised model. *TPAMI*, 2025. 1

[40] Tuomas Oikarinen, Subhro Das, Lam Nguyen, and Lily Weng. Label-free concept bottleneck models. In *ICLR*, 2023. 3, 5, 8, 16, 17, 19, 22, 25, 36

[41] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 6, 17

[42] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U^2-Net: Going deeper with nested U-structure for salient object detection. *Pattern recognition*, 106: 107404, 2020. 18, 49

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 19

[44] Sukrut Rao, Sweta Mahajan, Moritz Böhle, and Bernt Schiele. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In *ECCV*, 2024. 3

[45] Arthur S Reber. Implicit learning and tacit knowledge. *Journal of Experimental Psychology: General*, 118(3):219, 1989. 2

[46] Avinab Saha, Shashank Gupta, Sravan Kumar Ankireddy, Karl Chahine, and Joydeep Ghosh. Exploring explainability in video action recognition. In *CVPRW*, 2024. 3

[47] Kate Sanders, Nathaniel Weir, and Benjamin Van Durme. TV-TREES: Multimodal entailment trees for neuro-symbolic video reasoning. In *EMNLP*, 2024. 3

[48] Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *CVPR*, 2019. 5, 18, 23, 30, 31

[49] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *ICPR*, 2004. 3, 20, 24, 48

[50] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 7, 19, 25, 42

[51] Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani. Only time can tell: Discovering temporal data for temporal modeling. In *WACV*, 2021. 1

[52] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *CVPR*, 2016. 3, 8, 20, 21, 24, 26, 48

[53] Ziyao Shangguan, Chuhan Li, Yuxuan Ding, Yanan Zheng, Yilun Zhao, Tesca Fitzgerald, and Arman Cohan. TOMATO: Assessing visual temporal reasoning capabilities in multimodal foundation models. In *ICLR*, 2025. 3, 9, 21, 24, 25, 48

[54] Tongqing Shi, Ge Yan, Tuomas Oikarinen, and Tsui-Wei Weng. Multimodal concept bottleneck models. In *NeurIPSW*, 2025. 6, 29, 36

[55] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 1, 3

[56] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. OpenAI GPT-5 system card. *arXiv preprint arXiv:2601.03267*, 2026. 25, 26, 28, 29, 31, 32

[57] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. A dataset of 101 human action classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 3, 20, 21, 24, 48

[58] Divyansh Srivastava, Ge Yan, and Tsui-Wei Weng. VLG-CBM: Training concept bottleneck models with vision-language guidance. In *NeurIPS*, 2024. 3, 6, 25, 33, 35

[59] Alexandros Stergiou, Georgios Kapidis, Grigorios Kalliatakis, Christos Chrysoulas, Remco Veltkamp, and Ronald Poppe. Saliency tubes: Visual explanations for spatio-temporal convolutions. In *ICIP*, 2019. 2, 3

[60] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022. 1, 3, 8, 9, 15, 20, 21, 25, 30, 33, 49

[61] John YA Wang and Edward H Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, 1994. 1

[62] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. InternVid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*, 2024. 32, 33

[63] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. InternVideo2: Scaling foundation models for multimodal video understanding. In *ECCV*, 2024. 1, 3, 5, 8, 15, 19, 20, 22, 25, 30, 32, 33, 49

[64] Yihan Wang and Jia Deng. WAFT: Warping-alone field transforms for optical flow. In *ICLR*, 2026. 5, 17, 21, 49

[65] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *EMNLP*, 2020. 17, 19

[66] Eric Wong, Shibani Santurkar, and Aleksander Madry. Leveraging sparse linear layers for debuggable deep networks. In *ICML*, 2021. 6, 17, 23

[67] Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. STAR: A benchmark for situated reasoning in real-world videos. In *NeurIPS*, 2021. 3, 21, 23, 24, 36, 37, 48

[68] Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. Can I trust your answer? Visually grounded video question answering. In *CVPR*, 2024. 3

[69] Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. Energy-based concept bottleneck models: Unifying prediction, concept intervention, and probabilistic interpretations. In *ICLR*, 2024. 3

[70] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *CVPR*, 2023. 3

[71] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR*, 2023. 3

[72] Yuanhan Zhang, Bo Li, Haotian Liu, Yong Jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. LLaVA-NeXT: A strong zero-shot video understanding model, April 2024. URL https://llava-vl.github.io/blog/2024-04-30-llava-next-video/. 31, 32

Appendix

In this supplementary material, we provide additional details on the architecture, the optical flow / trajectory pipeline, datasets, baselines, and the user study, to complement the main paper. We organize the supplementary material as follows:

- Architecture details (Appendix A)
- Implementation details (Appendix B)
- Dataset details (Appendix C)
- Baseline details (Appendix D)
- Setup and protocol of the user study (Appendix E)
- Additional experimental results (Appendix F)
- Additional semantic and robustness analyses (Appendix G)
- Limitations (Appendix H)
- Broader impacts (Appendix I)

A Architecture Details

In this section, we describe the architectural components of `TraCE` in detail, complementing the overview given in the main paper.

A.1 Action Recognition Backbone

For action recognition, we use VideoMAE-B/16 [60] as the default video encoder f_V. Given an input video, f_V outputs a video-level feature $\mathbf{x}_i \in \mathbb{R}^d$, and the video-side concept layer g_V takes this feature as its input. VideoMAE-B/16 inherits its layout from ViT-Base. It stacks 12 transformer blocks with a hidden dimension of 768 and 12 attention heads. We thus set the feature dimensionality to $d = 768$. The encoder splits each input video into spatio-temporal tubelets of size $2 \times 16 \times 16$. It then applies full space-time self-attention over the tubelet tokens. Each token therefore attends to both spatial neighbors and temporally co-located patches. We feed an input video $\mathbf{V}_i \in \mathbb{R}^{T \times C \times H \times W}$ to f_V. Here, T denotes the number of sampled frames, C the number of color channels, and $H \times W$ the spatial resolution of each frame. f_V then outputs $\mathbf{x}_i = f_V(\mathbf{V}_i) \in \mathbb{R}^{768}$. We use $T = 16$ frames at $H \times W = 224 \times 224$ spatial resolution across all action recognition experiments. We initialize f_V with publicly available VideoMAE-B/16 weights and discard the original classification head. We fine-tune f_V on KTH and NTU RGB+D Mutual, and use the publicly released fine-tuned weights for Chiral SSv2 and UCF-101 (Appendix B.1). After fine-tuning, we freeze f_V and pre-extract $\mathbf{x}_i$ for every training and validation sample. This pre-extraction strategy decouples backbone training from concept-layer training. It also enables efficient and reproducible optimization of g_V and the sparse classifier $\mathbf{W}_{\text{AR}}$. Note that g_V depends on f_V only through the feature vector $\mathbf{x}_i$.

A.2 Video Question Answering Backbone

For video question answering, we use the Stage-2 1B variant of InternVideo2 [63]—denoted as InternVideo2$_{s2}$-1B—as the video encoder f_V, and we use its paired text tower as the text encoder f_T. We adopt InternVideo2$_{s2}$-1B because Stage-2 aligns the video and text encoders into a shared embedding space through video–audio–speech–text contrastive learning. This alignment matches the video-side concept layer g_V and the text-side concept projector g_T to a shared concept space, rather than to two independently learned spaces. The resulting concept relevance $\mathbf{r}_{ij}$ produced by g_T thus lives in the same space as the video concept activation $\mathbf{z}_i$ produced by g_V. The matching operator in Section 3.4.2 of the main paper then compares the two outputs dimension-wise.

Video tower. The video tower of InternVideo2$_{s2}$-1B inherits the same ViT backbone as the Stage-1 encoder, augmented with attention pooling. Stage-2 initializes the video tower from the Stage-1 checkpoint and continues training under cross-modal contrastive, matching, and masked language modeling losses. We feed an input video $\mathbf{V}_i \in \mathbb{R}^{T \times C \times H \times W}$ to f_V. f_V then outputs a video-level feature $\mathbf{x}_i = f_V(\mathbf{V}_i) \in \mathbb{R}^d$. We use $T = 4$ frames at $H \times W = 224 \times 224$ spatial resolution across all video question answering experiments.

Text tower. The text tower of InternVideo2$_{s2}$-1B is initialized from the first 19 layers of BERT-Large [16] and is jointly trained with the video tower during Stage 2. For each question–answer pair $[Q_i; a_{ij}]$, f_T outputs a text feature $\mathbf{u}_{ij} = f_T([Q_i; a_{ij}]) \in \mathbb{R}^{d_T}$.

Frozen encoders, trainable concept layers. We freeze both f_V and f_T throughout VQA training. We only update the video-side concept layer g_V, the text-side concept projector g_T, and the per-group weights γ_g used in the matching operator. This design isolates the explanatory components of `TraCE` from the large foundation backbones. It also keeps the shared concept space well-aligned with the pretrained video–text embedding space. We initialize f_V and f_T with the publicly released InternVideo2$_{s2}$-1B weights. Note that g_V and g_T depend on f_V and f_T only through the feature vectors $\mathbf{x}_i$ and $\mathbf{u}_{ij}$. The `TraCE` concept interface is therefore agnostic to the specific choice of the video–text foundation model.

A.3 Concept Layers

Video-side concept layer. The video-side concept layer $g_V : \mathbb{R}^d \to \mathbb{R}^M$ projects the video-level feature $\mathbf{x}_i$ onto the typed concept interface, where $M = M_t + M_o + M_s$. We implement g_V as a group-wise bias-free linear projection. Specifically, g_V consists of three disjoint weight matrices, $\mathbf{W}_V^t \in \mathbb{R}^{M_t \times d}$, $\mathbf{W}_V^o \in \mathbb{R}^{M_o \times d}$, and $\mathbf{W}_V^s \in \mathbb{R}^{M_s \times d}$, one per concept group. Given $\mathbf{x}_i$, the raw concept scores are

$$\mathbf{a}_i^t = \mathbf{W}_V^t \mathbf{x}_i, \quad \mathbf{a}_i^o = \mathbf{W}_V^o \mathbf{x}_i, \quad \mathbf{a}_i^s = \mathbf{W}_V^s \mathbf{x}_i. \tag{7}$$

We apply an element-wise sigmoid only to the trajectory branch: $\mathbf{z}_i^t = \sigma(\mathbf{a}_i^t)$, $\mathbf{z}_i^o = \mathbf{a}_i^o$, $\mathbf{z}_i^s = \mathbf{a}_i^s$. We omit the bias term in all three branches, following prior label-free concept bottleneck models [40].

We train the trajectory branch with soft-label binary cross-entropy against the soft trajectory presence label $\mathbf{c}_i^t \in [0,1]^{M_t}$ defined in Section 3.2.1:

$$L_t = -\frac{1}{M_t} \sum_{k=1}^{M_t} \left[c_{i,k}^t \log z_{i,k}^t + (1 - c_{i,k}^t) \log(1 - z_{i,k}^t) \right]. \tag{8}$$

The sigmoid in the trajectory branch matches the per-dimension probability interpretation required by this loss. We train the object and scene branches with the cosine-cubed loss [40] against the pseudo-labels $\mathbf{c}_i^g$ defined in Section 3.2.2:

$$L_g = -\frac{(\mathbf{z}_i^g)^3 \cdot (\mathbf{c}_i^g)^3}{\|(\mathbf{z}_i^g)^3\|_2 \, \|(\mathbf{c}_i^g)^3\|_2}, \quad g \in \{o, s\}. \tag{9}$$

The object and scene branches keep their raw scores because this loss operates on directional alignment rather than per-dimension probabilities.

The typed structure of g_V is shared across action recognition and video question answering, while we instantiate a separate g_V for each task. The two tasks differ only in how g_V is updated. For action recognition, we freeze g_V after concept bottleneck learning and train only the sparse classifier $\mathbf{W}_{\text{AR}}$. For video question answering, we initialize g_V from the concept bottleneck learning stage and continue updating it under concept supervision while keeping the video encoder frozen.

Text concept layer. We use the text-side concept projector g_T only in video question answering, where it maps the question–answer text feature into the same typed concept space as g_V. $g_T : \mathbb{R}^{d_T} \to \mathbb{R}^M$ is a group-wise bias-free linear projection, sharing the same structural design as g_V. g_T consists of three disjoint weight matrices, $\mathbf{W}_T^t \in \mathbb{R}^{M_t \times d_T}$, $\mathbf{W}_T^o \in \mathbb{R}^{M_o \times d_T}$, and $\mathbf{W}_T^s \in \mathbb{R}^{M_s \times d_T}$, one per concept group. For each question–answer pair $[Q_i; a_{ij}]$ with text feature $\mathbf{u}_{ij} = f_T([Q_i; a_{ij}])$, the question-conditioned concept relevance is

$$\mathbf{r}_{ij} = g_T(u_{ij}) = [\mathbf{r}_{ij}^t; \mathbf{r}_{ij}^o; \mathbf{r}_{ij}^s]. \tag{10}$$

Both g_V and g_T project into the same shared concept space, and the matching operator (Section 3.4.2) compares their outputs dimension-wise after a ReLU ℓ_2-normalization $\nu(\cdot)$.

We supervise object and scene relevance by text-concept similarity to natural-language concept names with a candidate-averaged cosine loss:

$$L^g_{\text{text}} = -\frac{1}{J}\sum_{j=1}^{J} \frac{\mathbf{r}^g_{ij}\cdot\mathbf{d}^g_{ij}}{\|\mathbf{r}^g_{ij}\|_2\,\|\mathbf{d}^g_{ij}\|_2}, \quad g\in\{o,s\}, \tag{11}$$

where $\mathbf{d}^g_{ij}$ is the text-side pseudo-label obtained by comparing the embedding of $[Q_i;a_{ij}]$ with the concept-name embeddings in $\mathbb{C}^g$, both encoded with the `all-mpnet-base-v2` model [65]. Trajectory relevance, in contrast, has no language names and cannot be supervised by text-concept similarity. We therefore align question-conditioned trajectory relevance with video-side trajectory evidence through an InfoNCE loss [41], where the relevance vector of the correct answer is the positive and other candidates are negatives:

$$L^t_{\text{align}} = -\log \frac{\exp(\text{sim}(\mathbf{z}^t_i,\mathbf{r}^t_{iy_i})/\tau)}{\sum_{j=1}^{J}\exp(\text{sim}(\mathbf{z}^t_i,\mathbf{r}^t_{ij})/\tau)}, \tag{12}$$

where $\text{sim}(\cdot,\cdot)$ is cosine similarity and τ is a temperature. This alignment lets the text branch learn which trajectory concepts are useful for each candidate answer without requiring language descriptions of motion.

A.4 Classifier Head

Sparse linear classifier for action recognition. For action recognition, we attach a sparse linear classifier $\mathbf{W}_{\text{AR}}\in\mathbb{R}^{K\times M}$ on top of the typed concept activation $\mathbf{z}_i$, where K is the number of action classes and $M = M_t + M_o + M_s$ is the total number of concepts. The classifier predicts $\hat{\mathbf{y}}_i = \text{softmax}(\mathbf{W}_{\text{AR}}\mathbf{z}_i)$. We optimize cross-entropy with elastic-net regularization on $\mathbf{W}_{\text{AR}}$ [66]:

$$L_{\text{AR}} = -\log\hat{\mathbf{y}}_i[y_i] \;+\; \lambda\left[(1-\alpha)\tfrac{1}{2}\|\mathbf{W}_{\text{AR}}\|_F^2 + \alpha\|\mathbf{W}_{\text{AR}}\|_{1,1}\right], \tag{13}$$

where $\|\cdot\|_F$ is the Frobenius norm and $\|\cdot\|_{1,1}$ is the element-wise ℓ_1 norm. The element-wise ℓ_1 penalty drives many entries of $\mathbf{W}_{\text{AR}}$ to zero. Each prediction therefore relies on a compact subset of trajectory, object, and scene concepts. We optimize $\mathbf{W}_{\text{AR}}$ with the GLM-SAGA solver [66], following prior label-free concept bottleneck models [26, 40].

Sparse matching operator for video question answering. For video question answering, we replace the linear classifier with a sparse matching operator that scores each candidate answer against the typed video-side concept activation. Given the video-side concept activation $\mathbf{z}_i$ and the question-conditioned concept relevance $\mathbf{r}_{ij}$ for candidate a_{ij}, we compute the compatibility score

$$s_{ij} = \sum_{g\in\{t,o,s\}} \exp(\gamma_g)\,\mu_{K_g}(\mathbf{z}^g_i,\mathbf{r}^g_{ij}), \tag{14}$$

where μ_{K_g} sums the top-K_g entries of $\nu(\mathbf{z}^g_i)\odot\nu(\mathbf{r}^g_{ij})$, $\nu(\mathbf{x}) = \text{ReLU}(\mathbf{x})/(\|\text{ReLU}(\mathbf{x})\|_2+\varepsilon)$ is a ReLU ℓ_2-normalization, and $\odot$ denotes element-wise multiplication. We use $K_t = K_o = K_s = 5$ across all video question answering experiments. We set the small constant to $\varepsilon = 10^{-8}$ to stabilize the normalization. The per-group scalar γ_g is a learnable parameter that controls the positive weight of concept group g. We initialize $\gamma_g = 0$ for all groups, which corresponds to $\exp(\gamma_g) = 1$ at the start of training, so that all three concept groups contribute equally. We then update γ_g jointly with g_V, g_T, and other VQA training parameters. The predicted answer is $\hat{a}_i = a_{ij_i^{\text{pred}}}$, where $j_i^{\text{pred}} = \arg\max_j s_{ij}$. At inference, we compute s_{ij} for every candidate $a_{ij}\in A_i$ and select the one with the largest compatibility score.

A.5 Concept Discovery

A.5.1 Trajectory Concept

Optical flow estimation. We use WAFT [64] as the optical flow estimator across all experiments. WAFT is a pretrained dense flow estimator that produces a per-pixel displacement field between two consecutive frames. We use the official released WAFT weights and do not fine-tune the estimator on any target dataset. Keeping the flow estimator frozen makes the trajectory pipeline backbone-

agnostic and reproducible across datasets. For each video, we sample frames at the original frame rate and compute forward flow between every pair of adjacent frames.

Trajectory generation. Given a video $\mathbf{V}_i$, we partition it into N_s temporal segments of equal duration and extract an L-frame clip centered at each segment center. Within a segment, let $\mathbf{F}_r : \Omega \to \mathbb{R}^2$ denote the forward flow field from frame r to frame $r+1$, where $\Omega = \{0,\dots,W-1\} \times \{0,\dots,H-1\}$ is the discrete pixel grid. For each seed pixel $\mathbf{p}_0 \in \Omega$, we trace a visual trajectory by recursively accumulating optical flow:

$$\mathbf{p}_{r+1} = \mathbf{p}_r + \mathbf{F}_r(\mathbf{p}_r), \quad r = 0,\dots,L-2. \tag{15}$$

Although the seed $\mathbf{p}_0$ lies on the discrete grid, subsequent positions $\mathbf{p}_r$ may take sub-pixel coordinates. We therefore evaluate $\mathbf{F}_r(\mathbf{p}_r)$ via bilinear interpolation. We represent each trajectory as the concatenated sequence of displacement vectors,

$$\tau(\mathbf{p}_0) = [\Delta\mathbf{p}_0,\dots,\Delta\mathbf{p}_{L-2}] \in \mathbb{R}^{2(L-1)}, \tag{16}$$

where $\Delta\mathbf{p}_r = \mathbf{p}_{r+1} - \mathbf{p}_r$. Applying this procedure to every segment yields the raw trajectory set $\widetilde{\mathbb{T}}_i$ for $\mathbf{V}_i$. The dataset-specific values of L and N_s are listed in Appendix B.3.

Trajectory filtering. Raw dense trajectories contain three failure modes: static background, weak or near-zero motion, and noisy flow predictions in low-texture regions. We address all three with a two-stage filter applied per segment. For notational simplicity, we describe the procedure for one segment and omit the segment index. Let $\widetilde{\mathscr{P}}$ denote the set of seed pixels in the segment, and let $M : \Omega \to \{0,1\}$ denote the foreground saliency mask obtained by an off-the-shelf foreground estimator [42] on the segment's center frame. **Stage 1 (foreground gating).** For each seed pixel $\mathbf{p}_0 \in \widetilde{\mathscr{P}}$, we retain its trajectory only if $\mathbf{M}(\mathbf{p}_0) = 1$. This focuses the trajectory pool on visible entity-level motion such as people, manipulated objects, and interacting agents. **Stage 2 (magnitude ranking).** Among the remaining foreground trajectories, we compute the motion magnitude

$$m(\tau) = \sum_{r=0}^{L-2} \|\Delta\mathbf{p}_r\|_2, \tag{17}$$

and retain the top-ρ% trajectories ranked by $m(\tau)$. We use $\rho = 5$ in all experiments. This stage suppresses low-amplitude flow noise and concentrates the trajectory vocabulary on salient entity-level motion. Applying both stages to every segment yields the filtered trajectory set $\mathbb{T}_i$ for $\mathbf{V}_i$.

Trajectory clustering and labeling. We aggregate the filtered trajectories over the training set, $\mathbb{T} = \bigcup_{i=1}^{N} \mathbb{T}_i$, and cluster them with FINCH [48]. FINCH produces a hierarchy of partitions at different levels of granularity without requiring a predefined number of clusters. We select a single partition level for each downstream dataset, and the resulting number of clusters M_t varies across datasets accordingly. We list the partition level and the resulting M_t per dataset in Appendix B.3. Let $\pi : \mathbb{T} \to \{1,\dots,M_t\}$ denote the cluster assignment of the selected partition. The k-th trajectory concept is the set of trajectories assigned to cluster k, $C_k^t = \{\tau \in \mathbb{T} \mid \pi(\tau) = k\}$. Each concept captures a recurring trajectory-level motion pattern shared across videos. For each video $\mathbf{V}_i$, we count how often its filtered trajectories are assigned to each concept,

$$h_{i,k} = \sum_{\tau \in \mathbb{T}_i} \mathbb{1}[\pi(\tau) = k], \tag{18}$$

and form a soft trajectory presence label

$$\mathbf{c}_i^t = \frac{\mathbf{h}_i}{\|\mathbf{h}_i\|_1} \in [0,1]^{M_t}. \tag{19}$$

The entry $c_{i,k}^t$ is the relative frequency of trajectory concept k in $\mathbf{V}_i$, and serves as the soft target for the trajectory branch in Equation (8).

Concept visualization. To visualize a trajectory concept, we select the medoid trajectory of its cluster as the canonical example. The medoid is the trajectory with the smallest average distance to all other members of the cluster, measured in the same flattened displacement space used for clustering. For each visualized concept, we identify the most strongly activating training video and

```
"List the main physical objects that are clearly visible in this video.\n"
"\n"
"Rules:\n"
"- Include ONLY objects you can clearly see and identify.\n"
"- If no clear object is identifiable, output exactly: none\n"
"- Do NOT guess or list objects that might be present but are not clearly visible.\n"
"- Include only tangible, movable objects.\n"
"- Do NOT include people or body parts.\n"
"- Do NOT include actions or activities.\n"
"- Do NOT include background locations (e.g., park, street, living room).\n"
"- Use short, singular, lowercase noun phrases (e.g., \"cup\", \"bicycle\", \"laptop\").\n"
"- One concept per line. No compound phrases joined by \"and\".\n"
"- Maximum 5 items. Fewer is fine.\n"
"- No full sentences. Bullet list format.\n"
"\n"
"Examples (text descriptions of hypothetical videos and the expected output):\n"
"\n"
"Video: Hands pick up a red cup from a wooden table next to a book and a pen.\n"
"Output:\n"
"- cup\n"
"- book\n"
"- pen\n"
"\n"
"Video: A blurry, dark scene where nothing is clearly identifiable.\n"
"Output:\n"
"- none\n"
"\n"
"Video: Someone places a smartphone, a pair of glasses, and a set of keys onto a desk.\n"
"Output:\n"
"- smartphone\n"
"- glasses\n"
"- key\n"
"\n"
"Video: A hand pours water from a bottle into a glass.\n"
"Output:\n"
"- bottle\n"
"- glass\n"
"\n"
"Now produce the output for the given video.\n"
```

(a) Object concept candidate prompt

```
"List the background scenes or environment types that are clearly visible in this video.\n"
"\n"
"Rules:\n"
"- Include ONLY scenes you can clearly identify from direct visual evidence.\n"
"- If no scene is clearly identifiable, output exactly: none\n"
"- Do NOT guess, infer, or imagine scenes that are not visually obvious.\n"
"- Do NOT include people, actions, objects, or equipment.\n"
"- Use short, singular, lowercase noun phrases (e.g., \"kitchen\", \"city street\", \"forest\").\n"
"- One concept per line. No compound phrases joined by \"and\".\n"
"- Maximum 3 items. Fewer is fine.\n"
"- No full sentences. Bullet list format.\n"
"\n"
"Examples (text descriptions of hypothetical videos and the expected output):\n"
"\n"
"Video: A person chops vegetables on a marble counter, with cabinets and a stove behind them.\n"
"Output:\n"
"- kitchen\n"
"- countertop\n"
"\n"
"Video: Two hands rotate a small object directly in front of a plain white background, no surroundings visible.\n"
"Output:\n"
"- none\n"
"\n"
"Video: A child runs across grass with trees in the distance and a slide nearby.\n"
"Output:\n"
"- park\n"
"- playground\n"
"\n"
"Video: Cars drive past tall glass buildings on a busy road with traffic lights.\n"
"Output:\n"
"- city street\n"
"\n"
"Now produce the output for the given video.\n"
```

(b) Scene concept candidate prompt

Figure 10: **Concept candidate generation prompts.** (a) Object and (b) scene prompts used to query Qwen3-VL [6].

locate the region in that video where the concept activation comes from using Grad-CAM [50]. We then overlay the medoid trajectory on the Grad-CAM-localized region, so that the visualization shows both the canonical motion pattern and the spatial region in the video that triggers the concept. We use Grad-CAM only to localize concept responses for visualization purposes, not to compute the explanations themselves. We follow this same protocol for all qualitative figures and supplementary visualizations involving trajectory concepts.

A.5.2 Object and Scene Concept

Object and scene concept candidate generation. We construct object and scene concept vocabularies by querying the training videos with the Video-LLM Qwen3-VL [6]. Unlike prior class-name prompting [26, 40], this video-grounded querying makes the candidate vocabulary reflect visible evidence rather than action-category priors. For each training video, we run two independent Video-LLM queries, one for object candidates and one for scene candidates, and we union the per-video outputs over the training set. The object prompt asks Qwen3-VL to list the main physical objects clearly visible in the video, with explicit instructions to exclude people, body parts, actions, and background locations, and to output a short bullet list of singular lowercase noun phrases. The scene prompt asks Qwen3-VL to list the background scenes or environment types clearly visible in the video, with explicit instructions to exclude people, actions, objects, and equipment. Both prompts cap the output length and provide in-context examples to enforce the bullet-list format. We provide the full prompt templates in Figure 10. Aggregating the per-video outputs yields the candidate sets $\widetilde{\mathbb{C}}^o$ and $\widetilde{\mathbb{C}}^s$.

Object and scene concept filtering. We refine $\widetilde{\mathbb{C}}^o$ and $\widetilde{\mathbb{C}}^s$ with four filters from LF-CBM [40]. **(i) Length filter.** We discard any candidate concept longer than 30 characters to keep concepts simple and to avoid unnecessarily long phrases. **(ii) Redundancy filter.** We remove duplicate and synonymous concepts using a weighted text-embedding similarity score that combines two encoders: MPNet [65] and CLIP ViT-B/16 [43]. **(iii) Activation filter.** For each remaining candidate, we compute its average video–text similarity score under the pretrained vision–language encoder [63] across the training set, and we discard concepts whose average score falls below an activation cutoff. This filter removes concepts that are not actually represented in the training data. **(iv) Projection filter.** After training the concept layer with the surviving candidates, we further remove concepts whose projected activations are consistently low across the training set, using a projection cutoff. This filter removes concepts that the model does not rely on after optimization. We do not apply the LF-CBM filter that removes concepts similar to target class names, because our object and scene candidates are generated from video evidence rather than action class names and rarely overlap with

Table 6: **Hyperparameters used for fine-tuning the action recognition backbone f_V on KTH and NTU RGB+D Mutual.** For Chiral SSv2 and UCF-101, we directly use the publicly released VideoMAE-B/16 fine-tuned weights from the official repository.

Config	KTH	NTU RGB+D Mutual
Model	ViT-Base/16	ViT-Base/16
Pretrained weights	VideoMAE-B/16 (SSv2)	VideoMAE-B/16 (SSv2)
Num. frames	16	16
Input size	224×224	224×224
Short-side size	224	224
Frame sampling	uniform	uniform
Optimizer	AdamW	AdamW
Optimizer momentum	β_1=0.9, β_2=0.999	β_1=0.9, β_2=0.999
Base learning rate	5e$-$4	1e$-$3
Weight decay	0.05	0.05
Learning rate schedule	cosine decay	cosine decay
Warmup epochs	5	5
Layer-wise lr decay	0.75	0.75
Training epochs	50	50
GPUs	4$\times$ RTX 3090	4$\times$ RTX 3090
Batch size per GPU	10	10
Augmentation	none	none
Test segments	2	2
Test crops	3	3

class labels. We tune the activation and projection cutoffs per dataset; the exact values and the resulting vocabulary sizes M_o and M_s are reported in Appendix B.3.

Concept labeling. For each training video $\mathbf{V}_i$ and each concept group $g \in \{o, s\}$, we generate a soft pseudo-label using a pretrained vision-language encoder $E_T(\cdot), E_V(\cdot)$. We use InternVideo2 [63] as the encoder to align the pseudo-labels with the VQA backbone described in Appendix B.2. Specifically, we encode every concept name in $\mathbb{C}^g$ with the text encoder, yielding the ℓ_2-normalized concept embedding matrix $E_T(\mathbb{C}^g) \in \mathbb{R}^{M_g \times d_E}$. We encode the video $\mathbf{V}_i$ with the video encoder, yielding $E_V(\mathbf{V}_i) \in \mathbb{R}^{d_E}$. The soft pseudo-label is

$$\mathbf{c}_i^g = E_T(\mathbb{C}^g) E_V(\mathbf{V}_i) \in \mathbb{R}^{M_g}, \quad g \in \{o, s\}. \tag{20}$$

We use the raw cosine similarity scores as soft pseudo-labels without further normalization. We use $\mathbf{c}_i^o$ and $\mathbf{c}_i^s$ as the soft targets for the object and scene branches in Equation (9).

B Implementation Details

B.1 Action Recognition Backbone

Data preprocessing. We sample $T = 16$ frames per video as input to f_V. We use dense sampling for UCF-101 [57] and uniform sampling for KTH [49], NTU RGB+D Mutual [52], and Chiral SSv2 [5, 18]. After sampling, we resize each frame so that the shorter side is 224 pixels and apply random cropping to 224×224. We do not apply any other data augmentation; in particular, we disable random erasing, mixup, and CutMix during fine-tuning.

Pretrained weights. For Chiral SSv2 and UCF-101, we directly use the publicly released VideoMAE-B/16 [60] fine-tuned weights from the official repository, without further fine-tuning. For KTH and NTU RGB+D Mutual, we initialize f_V from the VideoMAE-B/16 weights pretrained on Something-Something-V2 [18] and fine-tune on each target dataset. Something-Something-V2 is a large-scale, temporally fine-grained dataset, and the self-supervised VideoMAE weights pretrained on it are particularly effective for downstream datasets where temporal dynamics play a crucial role, such as KTH and NTU RGB+D Mutual.

Training. We fine-tune f_V on KTH and NTU RGB+D Mutual for 50 epochs with 5 warmup epochs and a cosine learning rate schedule [35]. We use the AdamW optimizer [36] ($\beta_1 = 0.9$, $\beta_2 =$

0.999, weight decay 0.05), apply layer-wise learning rate decay of 0.75, and use a base learning rate of 5e−4 for KTH and 1e−3 for NTU RGB+D Mutual. We conduct backbone training on 4 NVIDIA GeForce RTX 3090 GPUs with a batch size of 10 per GPU. We linearly scale the base learning rate with respect to the total batch size using the rule $lr = \text{base learning rate} \times \text{batch size}/256$, following VideoMAE [60]. We list the per-dataset hyperparameters in Table 6. After fine-tuning, we freeze f_V and pre-extract its video-level features $\mathbf{x}_i$ for every training and validation sample. This pre-extraction strategy decouples backbone training from concept-layer training and enables efficient and reproducible optimization of g_V and $\mathbf{W}_{\text{AR}}$.

B.2 Video Question Answering Backbone

We provide the architectural details of the video and text towers in Appendix A.2; here we describe data preprocessing, pretrained weights, and training details for the non-interpretable VQA baseline used as the reference point for `TraCE` in Table 2 of the main paper.

Data preprocessing. We sample $T = 4$ frames per video using uniform stride sampling: we partition the video into T equal-length segments and select the first frame of each segment. We resize each frame so that the shorter side is 224 pixels and apply center cropping to 224×224. For text inputs, we tokenize the question–answer pair $[Q_i; a_{ij}]$ with the BERT-Large [16] tokenizer paired with the InternVideo2 text tower, and we feed the tokenized sequence into the text tower without additional preprocessing. We do not apply any data augmentation during VQA training.

Pretrained weights. We initialize both the video tower and the text tower with the publicly released InternVideo2$_{s2}$-1B weights from the official repository, and we keep both encoders frozen throughout VQA training. The Stage-2 weights provide aligned video and text representations through video–audio–speech–text contrastive learning, which lets the projection layer described below adapt the pretrained features to multiple-choice answering on top of an already aligned embedding space (Appendix A.2).

Training. The pretrained video–text alignment from Stage-2 is not directly tuned for multiple-choice video question answering, so we train a single linear projection layer on top of the frozen video and text features to adapt them to the multiple-choice format. Specifically, we apply two separate linear projections, $\phi_V : \mathbb{R}^d \to \mathbb{R}^{d_p}$ and $\phi_T : \mathbb{R}^{d_T} \to \mathbb{R}^{d_p}$, to the video feature $\mathbf{x}_i = f_V(\mathbf{V}_i)$ and the question–answer feature $u_{ij} = f_T([Q_i; a_{ij}])$, with a shared projection dimension $d_p = 512$. We ℓ_2-normalize the two projections and score each candidate answer with their dot product:

$$s_{ij}^{\text{base}} = \exp(\gamma) \cdot \frac{\phi_V(\mathbf{x}_i)}{\|\phi_V(\mathbf{x}_i)\|_2} \cdot \frac{\phi_T(u_{ij})}{\|\phi_T(u_{ij})\|_2}, \tag{21}$$

where γ is a learnable log-temperature that we initialize to 2.3, so that $\exp(\gamma) \approx 10$ at the start of training. We optimize the parameters of ϕ_V, ϕ_T, and γ with the cross-entropy loss over the candidate scores $\{s_{ij}^{\text{base}}\}_{j=1}^{J}$ against the ground-truth answer index y_i. We use the same training configuration across all video question answering datasets (TOMATO [53], STAR [67], and the multiple-choice reformulations of NTU RGB+D Mutual [52], Chiral SSv2 [5, 18], and UCF-101 [57]); we list the hyperparameters in Table 7.

B.3 `TraCE`

After training (or selecting) the backbone encoder, we freeze its parameters and pre-extract feature vectors for all training and validation samples. This strategy enables efficient and reproducible training of the concept layers and the task-specific heads. We conduct all concept-layer and classifier training on a single NVIDIA GeForce RTX 3090 GPU.

Optical flow estimation. We use WAFT [64] as the optical flow estimator across all datasets. We use the official released WAFT weights and do not fine-tune the estimator on any target dataset, keeping the trajectory pipeline backbone-agnostic and reproducible. For each video, we sample frames at the original frame rate and compute forward flow between every pair of adjacent frames in the L-frame clip extracted from each segment (Appendix A.5.1). We perform optical flow estimation once per dataset and cache the resulting flow fields and trajectories on disk, so that trajectory clustering and concept-layer training can both reuse the cached results without recomputation.

Table 7: **Hyperparameters for training the baseline projection layer.** The video and text encoders (f_V and f_T) are kept frozen; only ϕ_V, ϕ_T, and the learnable temperature γ are updated. We use the same configuration across all video question answering datasets.

Config	Value
Frozen backbone	InternVideo2$_{s2}$-1B
Num. frames	4
Input size	224×224
Frame sampling	uniform stride
Projection dimension d_p	512
Feature normalization	ℓ_2
Temperature init (γ)	2.3
Optimizer	Adam [22]
Base learning rate	1e$-$3
Weight decay	0
Gradient clipping	1.0
Batch size	256
Training epochs	100
Random seed	42

Video-side trajectory concept layer training. We train the video-side trajectory branch $\mathbf{W}_V^t$ separately from the object and scene branches, following DANCE [26]. We optimize the soft-label binary cross-entropy loss L_t in (8) against the soft trajectory presence label $\mathbf{c}_i^t$ obtained from FINCH cluster assignments (Appendix A.5.1). We use the Adam optimizer [22] with a learning rate of 1e$-$3 and a batch size of 256, and we train for 3,000 steps with early stopping based on validation loss. The trajectory branch outputs sigmoid activations $\mathbf{z}_i^t = \sigma(\mathbf{a}_i^t)$, as defined in Section 3.3. We use the same hyperparameters across all action recognition and video question answering datasets.

Video-side object and scene concept layer training. We train the video-side object branch $\mathbf{W}_V^o$ and scene branch $\mathbf{W}_V^s$ separately, also following DANCE [26]. We optimize the cosine-cubed loss L_g in (9) against the video-text pseudo-labels $\mathbf{c}_i^g$ obtained from the InternVideo2 vision-language encoder [63] (Appendix A.5.2). We use the same Adam optimizer (lr $=$ 1e$-$3), batch size (256), and step budget (3,000 steps with early stopping on validation loss) as for the trajectory branch. The main difference from the trajectory branch lies in the loss function: trajectory concepts are trained with soft-label binary cross-entropy because they have soft presence labels, while object and scene concepts are trained with the cosine-cubed loss following label-free concept bottleneck models [26, 40]. The object and scene branches retain their raw scores, $\mathbf{z}_i^o = \mathbf{a}_i^o$ and $\mathbf{z}_i^s = \mathbf{a}_i^s$, as defined in Section 3.3. We use the same hyperparameters across all datasets.

Text-side trajectory concept layer training. We train the text-side trajectory branch $\mathbf{W}_T^t$ jointly with the video-side trajectory branch $\mathbf{W}_V^t$ and the trajectory matching weight γ_t, since trajectory concepts have no natural-language names and the text branch must instead be supervised by the video-side trajectory evidence (Section 3.4.2). We retain the video-side trajectory concept loss L_t during VQA training and optimize it together with the trajectory alignment loss L_{align}^t in (12) and the answer cross-entropy L_{QA}, following (6). We use the Adam optimizer with a learning rate of 1e$-$3 and a batch size of 256, and we train for 100 epochs. We initialize $\mathbf{W}_V^t$ from concept bottleneck learning and continue updating it under concept and answer supervision. We initialize $\mathbf{W}_T^t$ from scratch and set $\gamma_t = 0$. We use the same hyperparameters across all video question answering datasets.

Text-side object and scene concept layer training. During VQA training, we jointly update the text-side object and scene branches $\mathbf{W}_T^o, \mathbf{W}_T^s$, the corresponding video-side branches, and their matching weights γ_o, γ_s. The objective combines the cosine-cubed loss on the video side ($L_o + L_s$ in (9)), the candidate-averaged cosine loss on the text side ($L_{\text{text}}^o + L_{\text{text}}^s$ in (11)), and the video question answering cross-entropy L_{QA}, with weights of 0.8 on the concept losses and 0.2 on L_{QA}. We use the Adam optimizer with a learning rate of 1e$-$3 and a batch size of 256, and we train for 5,000 steps. We initialize $\mathbf{W}_V^o$ and $\mathbf{W}_V^s$ from concept bottleneck learning and use Xavier uniform initialization for $\mathbf{W}_T^o$ and $\mathbf{W}_T^s$. We set $\gamma_o = \gamma_s = 0$. We use the same hyperparameters across all video question answering datasets.

Table 8: **Dataset-specific concept configurations used by `TraCE`.** We report the trajectory length L, the number of segments per video N_s, the number of trajectory concepts M_t after FINCH clustering, the activation and projection cutoffs used to filter object and scene candidates, and the number of object and scene concepts after / before filtering (M_o and M_s). Cutoffs marked "–" are not applied: TOMATO and STAR are video question answering datasets evaluated with the unfiltered object and scene vocabularies. Trajectory concepts have no candidate filtering stage.

Dataset	L	N_s	M_t	Activation cutoff	Projection cutoff	M_o (after / before)	M_s (after / before)
NTU RGB+D Mutual	16	5	121	0.3	0.2	55 / 80	28 / 28
Chiral SSv2	16	3	154	0.3	0.2	404 / 1093	286 / 519
UCF-101	16	3	134	0.3	0.3	1103 / 1181	578 / 615
KTH	16	6	111	0.1	0.1	26 / 27	33 / 33
TOMATO	16	23	70	–	–	182 / 182	94 / 94
STAR	50	16	274	–	–	513 / 513	142 / 142

Concept configuration. We describe the dataset-specific concept configurations used across all experiments in Table 8. For each dataset, we list the trajectory length L, the number of segments per video N_s, the resulting number of trajectory concepts M_t produced by FINCH [48], the activation and projection cutoffs used to filter object and scene candidates, and the number of object and scene concepts before and after filtering, M_o and M_s. The trajectory length L is set to 16 for all datasets except STAR [67], where we increase it to 50 to capture longer-horizon motion required for situated reasoning. We do not apply the activation or projection cutoffs to TOMATO and STAR, since these video question answering datasets are evaluated with a fixed object and scene vocabulary that we keep unfiltered to preserve coverage over diverse question contents. Trajectory concepts have no candidate filtering stage; their count M_t is determined entirely by FINCH's automatic partition selection on the filtered trajectory set $\mathbb{T}$ (Appendix A.5.1).

Interpretable classifier for action recognition training. We employ the GLM-SAGA solver [66] to optimize the sparse linear classifier $\mathbf{W}_{\mathrm{AR}}$. We use $\alpha = 0.99$, $\lambda = 0.05$, a step size of 5,000, and a batch size of 512, following DANCE [26]. Early stopping is applied based on validation loss to avoid overfitting. During this stage, the video encoder f_V and the video-side concept layer g_V are both frozen; only $\mathbf{W}_{\mathrm{AR}}$ is updated.

Concept matching for video question answering. For video question answering, we initialize the video-side concept layer from concept bottleneck learning and continue updating it under concept supervision. We learn the text-side concept projector and matching weights through the answer loss, text-side concept supervision, and trajectory alignment. The overall VQA objective follows (6):

$$L_{\mathrm{VQA}} = \lambda_{\mathrm{QA}} L_{\mathrm{QA}} + \lambda_{\mathrm{concept}} L_{\mathrm{concept}} + \lambda_{\mathrm{text}} (L^{o}_{\mathrm{text}} + L^{s}_{\mathrm{text}}) + \lambda_{\mathrm{align}} L^{I}_{\mathrm{align}}, \tag{22}$$

with $\lambda_{\mathrm{QA}} = 0.2$, $\lambda_{\mathrm{concept}} = 0.8$, $\lambda_{\mathrm{text}} = 0.8$, and $\lambda_{\mathrm{align}} = 1.0$. We use the same loss weights across all video question answering datasets. We keep both encoders f_V and f_T frozen throughout; only g_V, g_T, and γ_g are updated.

Inference. During inference, we resize each frame so that the shorter side is 224 pixels and apply a center crop of 224×224. We do not apply any test-time augmentation and report all results using a single-view setting (1 clip $\times$ 1 crop). For action recognition, the predicted class is $\hat{c}_i = \arg\max_c [\mathbf{W}_{\mathrm{AR}} \mathbf{z}_i]_c$. For video question answering, the predicted answer is $\hat{a}_i = a_{i j_i^{\mathrm{pred}}}$, where $j_i^{\mathrm{pred}} = \arg\max_j s_{ij}$.

C Dataset Details

We evaluate `TraCE` on a suite of action recognition and video question answering datasets. The action recognition datasets test whether trajectory concepts capture dynamic evidence across diverse motion sources, while the video question answering datasets test whether the same typed concept interface supports language-grounded reasoning.

C.1 Action Recognition Datasets

NTU RGB+D. NTU RGB+D [52] is a large-scale indoor action recognition dataset captured with multiple camera views and Kinect depth sensors. We use two subsets: NTU RGB+D Mutual and NTU RGB+D Single. NTU RGB+D Mutual consists of 5 mutual-interaction classes where two people interact with each other: *pushing other person*, *hugging other person*, *giving something to other person*, *walking towards each other*, and *walking apart from each other*. NTU RGB+D Single consists of 12 single-person classes organized into 6 opposing pairs (e.g., *drop* vs. *pickup*, *sitting down* vs. *standing up*, *wear jacket* vs. *take off jacket*). We use the standard cross-subject split for both subsets. We include NTU RGB+D Mutual as the primary benchmark in Table 2 of the main paper because multi-person interactions directly test whether trajectory concepts can explain motion from multiple interacting agents beyond single-person pose. We use NTU RGB+D Single only in the concept-quality user study (Task 2 in Appendix E).

Chiral SSv2. Chiral SSv2 [5, 18] is a subset of Something-Something V2 [18], consisting of 32 classes where paired classes differ by signed motion direction (e.g., *pulling [something] left to right* vs. *right to left*). This design makes the dataset a strong test of whether trajectory concepts track motion direction rather than static appearance. We use the official Chiral SSv2 split.

UCF-101. UCF-101 [57] consists of 101 human action classes collected from YouTube. Many classes can be recognized from static scene or object cues, making it a useful test of whether trajectory concepts provide complementary dynamic evidence alongside object and scene concepts. We use the official split 1 for all experiments.

KTH. The KTH Action dataset [49] consists of 6 action classes performed by 25 subjects in 4 different scenarios. All actions involve single-person body motion (walking, jogging, running, boxing, hand waving, and hand clapping), making KTH a controlled test of basic trajectory concept quality on well-separated temporal patterns. We use the standard split.

C.2 Video Question Answering Datasets

TOMATO. TOMATO [53] is a temporal reasoning benchmark for multimodal foundation models that requires understanding of temporal order, duration, and frequency. The dataset provides human-centric, real-world, and simulated video scenarios. Since no official training split is provided, we use the human-centric and simulated scenarios as the training set and the real-world scenarios as the test set. Each question is a multiple-choice item with four answer candidates.

STAR. STAR [67] is a situated reasoning benchmark built upon real-world videos of human actions and interactions. The dataset includes four question types: interaction, sequence, prediction, and feasibility, each requiring structured situation comprehension and logical reasoning. We use the official train/val/test split. Each question is a multiple-choice item with four answer candidates.

Action Recognition-to-QA Reformulation. To test whether the same typed concept interface supports both class prediction and language-grounded reasoning, we reformulate three action recognition datasets (NTU RGB+D Mutual, Chiral SSv2, and UCF-101) into a multiple-choice video question answering format. Given a video, we pose the question "What is the action shown in this video?" and use all action class names as answer candidates. The ground-truth answer is the original action label. This reformulation lets us evaluate `TraCE`'s VQA instantiation on the same videos used for action recognition, enabling a direct comparison of the two task heads under the same typed concept interface.

C.3 Generalization Settings

Object-disjoint Split of Chiral SSv2. To evaluate whether trajectory concepts capture motion patterns rather than object appearance shortcuts, we construct an object-disjoint split of Chiral SSv2 using the Something-Else object annotations [38]. We partition the training and test sets so that the object categories in the two sets do not overlap. This forces the model to rely on motion evidence from trajectory concepts rather than recognizing familiar objects. We report results in Table 3 of the main paper.

Cross-domain. We evaluate cross-domain transfer on 65 TOMATO [53] videos labeled with leftward, rightward, upward, or downward motion. We directly apply the trajectory-only `TraCE` pipeline trained on Chiral SSv2 [5, 18], keeping the video encoder, concept vocabulary, concept layer, and classification head fixed. For each video, we select the top-1 prediction from the original 32 Chiral SSv2 classes. A prediction is correct if it matches a source action class corresponding to the video's direction. For leftward motion, we accept *Pushing [something] from right to left* or *Pulling [something] from right to left*. For rightward motion, we accept *Pushing [something] from left to right* or *Pulling [something] from left to right*. For upward motion, we accept *Moving [something] up*. For downward motion, we accept *Moving [something] down*. All other predictions count as incorrect. We use the same evaluation criterion for the non-interpretable baseline.

D Baseline Details

We provide implementation details for every baseline compared against `TraCE` in the main paper. For fair comparison, all baselines share the same backbone encoder as `TraCE` unless explicitly noted.

D.1 Saliency-based Explanations

GradCAM. Grad-CAM [50] generates class-discriminative saliency maps by computing the gradient of the predicted class score with respect to intermediate layer activations. We use Grad-CAM solely for visualization purposes: to localize which spatial region in a video activates a given trajectory concept (Appendix A.5.1). For the VideoMAE-B/16 backbone [60], we extract Grad-CAM maps from `blocks[9].norm`; for the InternVideo2 video tower [63], we use `blocks[38].norm`. Grad-CAM is not used as a baseline explanation method in any quantitative comparison.

SALF-CBM. SALF-CBM [7] is a spatially-aware concept bottleneck model originally designed for image classification, generating visual explanations via spatially localized concept activations. We extend it to video by adding a temporal axis to the spatial concept maps while keeping the rest of the architecture unchanged. We use the same frozen VideoMAE-B/16 backbone and follow the official implementation for all other components. We compare against SALF-CBM in the user study (Figure 5 of the main paper) and in Table 2 for action recognition.

D.2 Concept-based Explanations

LF-CBM. Label-Free CBM [40] discovers concept vocabularies from language descriptions and trains a concept bottleneck model without manual concept annotation. We use the official LF-CBM codebase without modification, replacing only the backbone with the same frozen VideoMAE-B/16 features used by `TraCE`. LF-CBM uses language-derived concepts for all concept types, including motion, which limits its ability to represent fine-grained motion patterns that are difficult to express in words. We compare against LF-CBM in Table 2 for action recognition.

VLG-CBM. VLG-CBM [58] trains concept bottleneck models with vision-language guidance, using CLIP-based concept scoring to improve concept quality. We do not compare against VLG-CBM as a baseline in Table 2. Instead, we adopt the Number of Effective Concepts (NEC) metric proposed by VLG-CBM to evaluate concept quality and report results in the additional experimental results (Appendix F.2).

DANCE. DANCE [26] is a concept-based video explanation method that represents motion using human pose sequences extracted by a pretrained pose estimator. We use the same frozen VideoMAE-B/16 backbone and follow the official implementation. DANCE defines motion concepts over single-person pose sequences, which makes its motion vocabulary source-incomplete: it cannot directly represent object trajectories and multi-agent interactions. We compare against DANCE in Table 2 for action recognition and in the user study (Figure 5 of the main paper).

D.3 Language-based Explanations

GPT-5.4-derived concepts. We construct a language-based motion concept baseline by prompting GPT-5.4 [56] to generate motion-describing concepts for each action class. We then use these

language-derived motion concepts in the same concept bottleneck structure as TraCE, replacing trajectory concepts with the generated language motion concepts while keeping the same object and scene concepts, concept layer architecture, and sparse classifier. This baseline tests whether language descriptions can substitute for visual trajectory concepts in capturing fine-grained motion evidence. We compare against this baseline in the user study (Figure 5 of the main paper) and in the concept intervention curves (Figure 8 of the main paper).

E User Study Setup and Protocol

To assess the interpretability of TraCE, we conducted a user study to evaluate the quality of trajectory concepts and the effectiveness of trajectory-based explanations. The results of this study are summarized in Figure 5 of the main paper, and we provide further details in this section. Specifically, our study aimed to answer two key questions:

- Does TraCE's explanation better help users understand the model's prediction compared to other explanation methods?
- Do trajectory concepts effectively convey motion patterns in the video?

E.1 Recruitment

We collected responses from 50 adult participants recruited through the Prolific platform from English-speaking regions, regardless of gender. Each participant answered 57 questions, which included 2 sanity check questions to ensure engagement and attention. Participants were compensated £9 for the entire study. On average, participants completed the study in about 27 minutes. This study was reviewed and approved as exempt by the Institutional Review Board (IRB) at our institution. We did not collect any personal information from the participants, nor did we disclose any details about the model or research objectives during the study.

E.2 Study Design

E.2.1 Task 1: Quality of TraCE's Explanation

Task 1 aims to address the first key question: *Does TraCE's explanation better help users understand the model's prediction compared to other explanation methods?* To evaluate the interpretability of TraCE, we compare its explanations with those of three baseline methods: DANCE [26], which represents motion using human pose sequences, a language-based CBM using GPT-5.4-derived motion concepts [56], and SALF-CBM [7], a saliency-based concept bottleneck model.

Experimental setting. We conduct pairwise comparisons on motion-focused subsets of NTU RGB+D Mutual [52] and Chiral SSv2 [5, 18]. For each comparison, participants are shown a video and two explanations (Explanation A and Explanation B), where one is from TraCE and the other is from a baseline. The assignment of TraCE to A or B is randomized. Each explanation shows the top-3 contributing concepts overlaid on the input video.

Evaluation protocol. Participants are asked: "Which explanation better helps you understand why the model predicted the action?" Responses are collected on a five-point scale: *A is much better*, *A is slightly better*, *About the same*, *B is slightly better*, and *B is much better*. Each participant answers 18 pairwise comparison questions across the three baselines. We report the distribution of responses in Figure 5 (a) of the main paper.

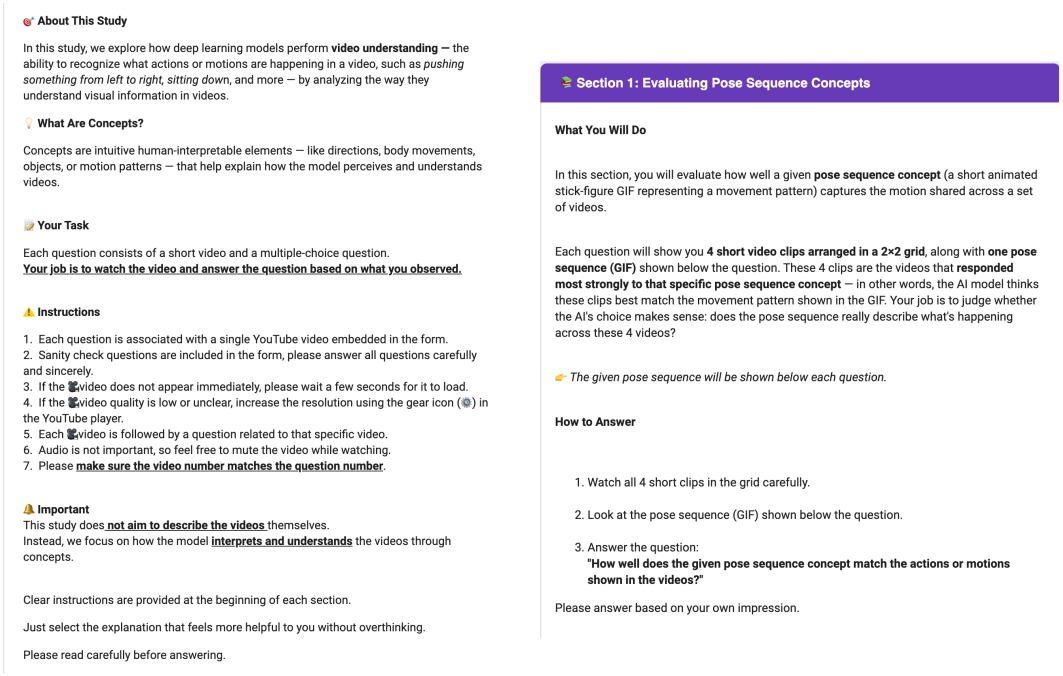
E.2.2 Task 2: Quality of TraCE's Trajectory Concept

Task 2 aims to address the second key question: *Do trajectory concepts effectively convey motion patterns in the video?* This task evaluates the interpretability of individual concepts by asking whether the concept matches the motion observed across its most strongly activating video clips.

Experimental setting. We compare three types of concepts: (i) *Trajectory concepts* from TraCE, (ii) *Pose sequence concepts* from DANCE [26], and (iii) *Language-derived motion concepts* generated by GPT-5.4 [56]. For each concept type, we randomly select one concept and visualize it with its top-4 most strongly activating video clips. This task is conducted on motion-focused subsets of NTU RGB+D Mutual, NTU RGB+D Single, and Chiral SSv2.

Evaluation protocol. Participants are asked: "How well does the given concept match the actions or motions shown in the video above?" Responses are collected on a five-point Likert scale (1: Strongly Disagree to 5: Strongly Agree). We independently evaluate each concept type without directly comparing them to each other. Each participant answers 37 concept evaluation questions across the three concept types. We report the average Likert scores in Figure 5 (b) of the main paper.

E.3 Interface and Instructions



(a) Task overview instruction interface

(b) Section-specific instruction interface

Figure 11: **Instruction interfaces shown to participants.** (a) We provide overview instruction at the beginning of the study. (b) We provide section-specific instructions before each task to guide participants throughout the study.

We provide clear instructions to participants prior to each task, as illustrated in Figure 11. At the beginning of the study, participants receive an overview instruction page explaining that concepts are intuitive human-interpretable elements—such as body movements, objects, or motions—that help explain how the model perceives and understands the videos. We emphasize that the study does not aim to describe the videos themselves, but to evaluate how well the model interprets and understands the videos through concepts. Before each task, section-specific instructions describe the evaluation criteria and how to answer.

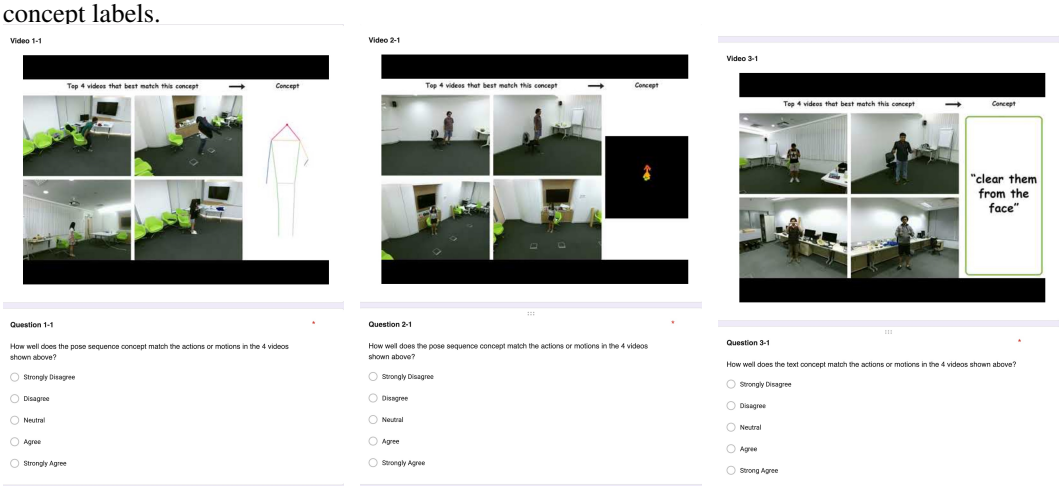
Task 1 interface. For each pairwise comparison question, we present a video with two explanations (Explanation A and Explanation B) side by side. Each explanation shows the top-3 contributing concepts overlaid on the input video. Participants are asked: "*Which explanation better helps you understand why the model predicted the action?*" Responses are collected on a five-point scale: *A is much better*, *A is slightly better*, *About the same*, *B is slightly better*, and *B is much better*. We show an example of the Task 1 evaluation interface in Figure 12.

Task 2 interface. For each concept evaluation question, we present a single concept alongside its top-4 most strongly activating video clips. Participants are asked: "*How well does the given concept match the actions or motions shown in the video above?*" Responses are collected on a five-point Likert scale (1: Strongly Disagree to 5: Strongly Agree). Each interface corresponds to one of the three concept types: trajectory concepts (`TraCE`), pose sequence concepts (DANCE), or language-



(a) TraCE vs DANCE (b) TraCE vs Language-based (c) TraCE vs SALF-CBM

Figure 12: **Example question and user interface used in Task 1.** Participants are asked to compare two explanations (A and B) for each sample or class. (a) DANCE [26], (b) Language-based [56], and (c) SALF-CBM [7]. In (a) and (b), we show a single video sample with top-3 sample-level concept contributions. In (c), SALF-CBM provides a heatmap-based spatial explanation alongside concept labels.



(a) Pose-based (b) Trajectory (c) Language-based

Figure 13: **Example question and user interface used in Task 2.** We show how participants evaluate the interpretability of a single concept using several highly activated video clips. Each interface corresponds to one of the three concept types.

derived motion concepts (GPT-5.4). We show an example of the Task 2 evaluation interface in Figure 13.

E.4 Additional Results

To further validate the user study results in Figure 5 (b) of the main paper, we report mean Likert scores under two conditions in Table 9. In the *original* condition, participants evaluate a concept alongside its top-4 most strongly activating video clips. In the *random* condition, we keep the same highly activated video clips but replace the displayed concept with a randomly selected concept from the same method. If a concept type is genuinely interpretable, the original condition should receive substantially higher ratings than the random condition, because the displayed concept should match the motion observed in the clips. `TraCE` achieves the highest mean score of 4.22 under the

Table 9: **Mean Likert scores for Task 2 (concept quality).** We report the average score (1: Strongly Disagree to 5: Strongly Agree) over 300 responses per condition. The *original* condition shows each concept with its top-4 most strongly activating video clips; the *random* condition shows a randomly selected concept with the same clips. Δ denotes the change from original to random (random − original).

Method	Original	Random	Δ
DANCE [26]	2.52	2.55	0.03
`TraCE`	**4.22**	2.67	−1.55
Language-based [56]	2.27	1.86	−0.41

original condition, indicating that participants strongly agree that trajectory concepts match the observed motion. When the concept is replaced with a random one, the mean drops sharply to 2.67 ($\Delta = -1.55$), confirming that the high ratings reflect genuine concept–motion alignment rather than a response bias. In contrast, DANCE shows almost no difference between the original (2.52) and random (2.55) conditions ($\Delta = +0.03$). We attribute this to the nature of the evaluation datasets: NTU RGB+D Single contains many simple, repetitive single-person motions (e.g., sitting down, standing up), so different pose sequence concepts tend to look similar to participants, making it difficult to distinguish the correct concept from a random one. The language-based baseline receives the lowest original score (2.27), and its random condition score drops modestly ($\Delta = -0.41$), indicating that participants can sometimes distinguish correct language concepts from random ones, but the overall interpretability remains low. Together, these results support the conclusion in the main paper that trajectory concepts are more interpretable and better aligned with underlying motion patterns than both pose-based and language-based alternatives.

E.5 Pairwise User Study in the VQA Setting

The user study in Figure 5 evaluates `TraCE` explanations for action recognition. We extend the evaluation to question-conditioned predictions through a blinded pairwise study with 50 adult participants recruited through Prolific. To isolate explanation quality from answer correctness, we include only examples for which both compared models produce the same correct answer. Each item presents an identical video, question, answer candidates, and model answer together with two concept-based explanations. Both explanations contain the same number of high-contribution concepts and use the same presentation format; the method identities are hidden. Participants answer, "Which explanation better helps you understand why the model chose its answer?"

Each participant evaluates five comparisons against the MM-CBM [54] variant with language-derived motion concepts and five against the MM-CBM variant with pose-sequence concepts, yielding 250 participant–item ratings per comparison. The examples span the native TOMATO and STAR benchmarks and multiple-choice VQA reformulations of NTU RGB+D Mutual and UCF-101.

Table 10: **Pairwise explanation preference in the VQA setting.** Each comparison contains 250 participant–item ratings from examples for which both models produce the same correct answer. The five response columns are ordered from `TraCE` being much better to the baseline being much better; the final column combines the first two responses.

Comparison	Much better	Slightly better	About the same	Slightly worse	Much worse	Prefer `TraCE`
Language-derived motion concepts	123 (49.2%)	53 (21.2%)	25 (10.0%)	34 (13.6%)	15 (6.0%)	**70.4%**
Pose-sequence concepts	167 (66.8%)	47 (18.8%)	23 (9.2%)	10 (4.0%)	3 (1.2%)	**85.6%**

As shown in Table 10, participants prefer `TraCE` in 70.4% of comparisons against explanations based on language-derived motion concepts and 85.6% against explanations based on pose-sequence concepts. Both models are correct on every evaluated item and display the same answer, so the preferences measure the perceived helpfulness of the explanations rather than differences in prediction correctness. Because the study contains both native and multiple-choice reformulated examples, we describe it as a VQA-setting study; native VQA faithfulness is evaluated separately in Appendix F.4.

F Additional Experimental Results

For fair comparison, all experiments in this section use the same backbone, VideoMAE-B/16 [60] for action recognition and InternVideo2$_{s2}$-1B [63] for video question answering unless otherwise noted. Unless specified otherwise, all ablation studies are conducted on Chiral SSv2 [5, 18] dataset.

F.1 Trajectory Concept Generation Ablation

Table 11: **Effect of the number of trajectory concepts.**

Chiral SSv2		NTU RGB+D Mutual		UCF-101		KTH	
# Concepts	Acc.	# Concepts	Acc.	# Concepts	Acc.	# Concepts	Acc.
870	86.8	681	97.5	4124	89.5	589	91.0
154	84.2	121	97.5	759	88.6	111	92.0
29	80.3	25	96.9	134	88.4	21	90.4

Effect of number of concepts. We ablate the effect of the number of trajectory concepts M_t by varying the FINCH partition level. A higher partition level yields fewer, coarser clusters, while a lower level produces more, finer-grained clusters. As shown in Table 11, the default configurations listed in Table 8 achieve accuracy within 3 percentage points of the finest partition across all datasets, while using substantially fewer concepts. On NTU RGB+D Mutual, accuracy is nearly invariant to the number of concepts (97.5% at both 681 and 121 concepts), reflecting the near-saturated nature of this benchmark. On KTH, the default partition ($M_t = 111$) slightly outperforms the finest partition ($M_t = 589$, 92.0% vs. 91.0%), suggesting that an excessive number of fine-grained concepts can introduce noise for datasets with simple, well-separated motion patterns. In contrast, Chiral SSv2 shows the largest sensitivity to concept count (86.8% $\rightarrow$ 80.3% from 870 to 29 concepts), which is expected because its paired directional classes require fine-grained trajectory discrimination. These results suggest that the default FINCH [48] partition provides a good trade-off between concept granularity and predictive performance across diverse motion types.

Effect of the number of segments. As shown in Table 12, accuracy is robust to the number of temporal segments N_s, with $N_s = 3$ achieving the best result (84.2%). Using a single segment ($N_s = 1$) slightly reduces accuracy (83.2%) because the entire video is summarized by one clip, losing temporal coverage. Increasing to $N_s = 5$ also decreases accuracy to 82.6%. We set N_s per dataset by dividing the average number of frames in the training set by the trajectory length L, which provides a natural balance between temporal coverage and clip quality.

Effect of trajectory length L. As shown in Table 13, longer trajectories ($L = 32$) achieve the highest accuracy (87.3%), while the default length ($L = 16$) achieves 84.2% and shorter trajectories ($L = 8$) achieve 86.9%. However, accuracy alone does not reflect concept interpretability. We visualize trajectory concepts discovered at each length in Figure 14. Short trajectories ($L = 8$) capture only brief, fragmented motion that lacks sufficient temporal context to convey a meaningful motion pattern. Long trajectories ($L = 32$) tend to encode the entire action class rather than reusable motion primitives shared across classes—analogous to classifying an image by the object itself rather than by its distinguishing attributes. We therefore adopt $L = 16$ as the default, which captures representative motion fragments that are both discriminative and generalizable as an interpretable trajectory vocabulary.

Effect of clustering method. We compare FINCH [48] with K-Means [37] as the trajectory clustering algorithm. As shown in Table 14, the two methods achieve comparable accuracy (84.2% vs. 84.6%), confirming that trajectory concept quality is largely agnostic to the choice of clustering algorithm. We adopt FINCH as the default because it automatically determines the number of clusters without requiring a predefined k, making the pipeline easier to apply across datasets without per-dataset hyperparameter tuning.

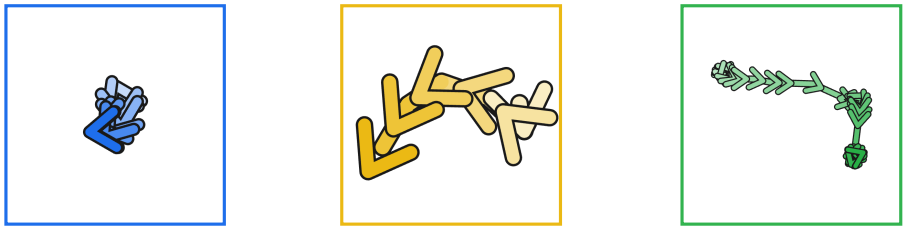
Soft vs. hard concept labels. We compare three strategies for converting trajectory cluster assignments into video-level concept labels: (i) *Binary*, which assigns 1 if any trajectory in the video belongs to the cluster and 0 otherwise; (ii) *Soft* (default), which uses the normalized frequency

Table 12: **Effect of the number of segments.**

N_s	Acc.
1	83.2
3	84.2
5	82.6

Table 13: **Effect of the trajectory length.**

L	Acc.
8	86.9
16	84.2
32	87.3



(a) L=8 (b) L=16 (c) L=32

Figure 14: **Trajectory concepts discovered with different trajectory lengths.** Short trajectories ($L = 8$) capture only local and fragmented motion patterns with limited temporal structure. Long trajectories ($L = 32$) often encode near-complete action dynamics, reducing concept reusability across classes. In contrast, the default trajectory length ($L = 16$) captures representative motion fragments that remain both interpretable and reusable across different videos. These visualizations illustrate the trade-off between local motion specificity and action-level over-specialization when choosing the trajectory length.

$\mathbf{c}_i^t = \mathbf{h}_i / \|\mathbf{h}_i\|_1$; and (iii) *Max normalized*, which divides each count by the maximum count across all concepts for the video. As shown in Table 15, the soft labeling strategy achieves the best accuracy (84.2%), outperforming both binary (82.4%) and max normalized (83.3%) alternatives. Binary labels discard frequency information and treat rare and dominant motion patterns equally, which reduces the model's ability to distinguish fine-grained directional cues. Soft labels preserve relative frequency, providing richer supervision that better captures the distribution of trajectory concepts within each video.

F.2 Analysis

Effect of Video-LLM. We compare five language models for generating object and scene concept candidates: GPT-5.4 [56], LLaVA-NeXT-Video [72], Video-LLaVA [32], VideoLLaMA2 [13], and Qwen3-VL [6]. As shown in Table 16, Video-LLaVA and VideoLLaMA2 achieve the highest accuracy (77.7% and 77.3%), while GPT-5.4 produces the fewest concepts (152) and the lowest accuracy (61.5%). Qwen3-VL achieves competitive accuracy (74.6%) despite generating the most candidates (1,683). However, accuracy alone does not capture concept quality. We qualitatively observe that many models produce concepts that violate the typed separation between object and scene groups: for example, scene-like terms appear in the object list, or action verbs are included among object candidates. We show representative examples of such violations in Figure 15. We adopt Qwen3-VL as the default because it produces the cleanest separation between object and scene concepts when

Table 14: **Effect of clustering method on trajectory concept discovery.**

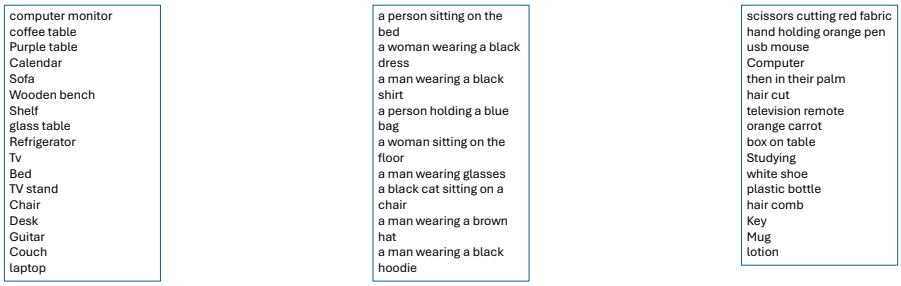
Method	Acc.
FINCH [48]	84.2
K-Means [37]	84.6

Table 15: **Effect of the trajectory labeling method.**

Method	Acc.
Binary	82.4
Soft	84.2
Max normalized	83.3

Table 16: **Effect of different object and scene concept discovery method.**

Method	# Concepts	Acc.
GPT-5.4 [56]	152	61.5
LLaVA-NeXT-Video [72]	393	71.8
Video-LLaVA [32]	1,026	77.7
VideoLLaMA2 [13]	1,478	77.3
Qwen3-VL [6]	1,683	74.6



(a) LLaVA-NeXT-Video[72] scene concept candidates (b) Video-LLaVA [32] object concept candidates (c) VideoLLaMA 2 [13] scene concept candidates

Figure 15: **Qualitative comparison of video-language models for text concept generation.** We visualize representative object and scene concept candidates generated by different video-language models. (a) LLaVA-NeXT-Video [72] produces relatively clean scene-centric concepts. (b) Video-LLaVA [32] frequently mixes appearance descriptions and human-centric phrases into object concepts. (c) VideoLLaMA2 [13] sometimes generates action phrases or interaction descriptions instead of static scene concepts. These examples highlight that higher downstream accuracy does not necessarily imply cleaner object–scene concept separation.

guided by our structured prompts (Figure 10), even though other models may achieve slightly higher accuracy with noisier concept vocabularies.

Vision-language model ablation. Object and scene concept layers are supervised by soft pseudo-labels computed from video–text similarity using a pretrained vision–language encoder (Appendix A.5.2). We ablate the choice of this encoder while keeping the same concept candidates and backbone. As shown in Table 17, replacing InternVid [62] with InternVideo2 [63] as the labeling encoder improves accuracy from 83.4% to 84.2%. We attribute this improvement to the stronger video–text alignment provided by InternVideo2's Stage-2 contrastive pretraining, which produces more accurate soft pseudo-labels for each concept. We therefore use InternVideo2 as the default vision–language encoder for concept labeling in all experiments.

Disentanglement analysis. We analyze whether the three concept groups (trajectory, object, scene) use disentangled subspaces of the backbone feature $\mathbf{x}_i$. For this additional analysis, we ablate two orthogonality mechanisms: (1) *SVD null-space projection*, which removes previously used subspace dimensions from the backbone feature before training the next concept group, and (2) $\mathbf{W}_c$ *row-orthogonality loss* with weight λ, which penalizes cosine similarity between weight rows of different groups. We measure disentanglement via top-K backbone-dimension overlap: for each group g, we compute a per-dimension importance score $\text{imp}_g[j] = \text{mean}_n|\mathbf{x}_n[j]| \cdot \text{mean}_c|\mathbf{W}_c^g[c,j]|$ over validation samples, select the top-K dimensions, and count shared indices across group pairs. The random-chance baseline for $K = 50$ is $K^2/d = 3.26$ shared dimensions.

Group ordering sensitivity. Since concept groups are trained sequentially, the training order may affect both accuracy and disentanglement. We fix SVD + Ortho ($\lambda = 1.0$) and train all six permutations of {trajectory (T), object (O), scene (S)}. As shown in Table 19, the ordering induces a 1.2%p accuracy spread and a 1.7× variation in maximum pairwise top-50 overlap. *Trajectory-first* orders (TSO, TOS) achieve the highest average accuracy (86.85%), because the trajectory branch uses a structurally distinct BCE head and naturally occupies a separate subspace, imposing no representational cost on subsequent groups. *Object-first* orders (OTS, OST) yield the lowest accuracy

Table 17: **Evaluation of VLM encoder for object and scene concept labeling**

Video encoder	VLM encoder	Acc.
VideoMAE [60]	+ InternVid [62]	83.4
	+ InternVideo2 [63]	84.2

Table 18: **Component ablation for concept group disentanglement.** We report validation accuracy (%) and top-50 backbone-dimension overlap between group pairs. Random-chance overlap is 3.26. The object–scene pair (object ∩ scene) is the structurally hardest target because both groups are text-based visual concepts.

Variant	Acc. (%)	traj ∩ obj	traj ∩ scene	obj ∩ scene
Baseline	84.20	12	10	34
SVD only (α=1.0)	84.55	10	19	19
Ortho only (λ=1.0)	86.92	8	9	21
SVD + Ortho (λ=1.0)	86.78	8	9	22
SVD + Ortho (λ=2.0)	**87.83**	8	8	26

(85.94% average), as the object group has the largest concept set (404 concepts) and consumes a large subspace first, leaving insufficient residual representation for the remaining groups. For disentanglement, placing the semantically similar object–scene pair adjacent at the start of the sequence (SOT) achieves the lowest maximum overlap (13) and lowest sum overlap (29), because both groups can freely claim distinct subspaces before the residual capacity is reduced. TSO achieves the best accuracy (86.92%) with a moderate maximum overlap (20), making it a practical accuracy-priority choice. We use TOS for the component ablation in Table 18.

Component ablation. We compare four variants against the baseline (no orthogonality) in Table 18. The $\mathbf{W}_c$ orthogonality loss is the dominant driver of both accuracy and disentanglement. SVD alone contributes only +0.35%p in accuracy and becomes redundant at inference when the orthogonality loss is active: for all ortho-containing variants, removing the SVD projection at inference causes no accuracy change ($\Delta = 0$), indicating that the learned head does not rely on the projected subspace. Combining SVD with orthogonality at $\lambda = 2.0$ achieves the best accuracy (87.83%). The object–scene pair (object ∩ scene) has the highest overlap across all variants, reflecting the structural challenge that both groups are text-based visual concepts. All variants reduce object–scene overlap relative to the baseline (34 → 19–26), confirming that orthogonality mechanisms successfully encourage the two groups to use more distinct backbone dimensions. Notably, increasing λ from 1.0 to 2.0 improves accuracy (+1.0%p) but slightly increases top-50 overlap (22 → 26), because row-orthogonality can be satisfied by using the same dimensions with opposite sign or scale. This shows that $\mathbf{W}_c$ row-orthogonality and top-K dimension disjointness are complementary but distinct properties.

Concept combination. We evaluate the contribution of each concept type by training `TraCE` with different subsets of the three concept groups. As shown in Table 20, using all three groups (trajectory + object + scene) achieves the best accuracy (84.2%). Trajectory concepts alone already capture the majority of the performance (80.7%), confirming that motion evidence is the primary driver on this benchmark. Adding object or scene concepts on top of trajectory yields consistent gains (+1.7%p and +0.9%p, respectively), showing that static context provides complementary evidence that trajectory concepts alone do not capture. In contrast, removing trajectory concepts entirely (object + scene only) causes a substantial drop to 74.6%, a −9.6%p gap relative to the full model. This confirms that trajectory concepts are the most informative group, while object and scene concepts serve as useful but secondary complements.

Concept sparsity. We evaluate whether `TraCE` maintains predictive performance with a compact set of concepts using the Number of Effective Concepts (NEC) metric proposed by VLG-CBM [58]. We vary the regularization strength λ of the GLM-SAGA solver to control the average number of non-zero weights per class in $\mathbf{W}_{\text{AR}}$. A smaller NEC restricts the number of concepts used by each class. As shown in Table 21, `TraCE` maintains strong accuracy even under tight NEC budgets. At NEC = 25, accuracy reaches 83.8%, within 0.4 percentage points of the default model (84.2%).

Table 19: **Effect of concept group training order on accuracy and disentanglement.** We report validation accuracy (%) and top-50 backbone-dimension overlap between group pairs for all six permutations of {trajectory (T), object (O), scene (S)}. SVD + Ortho ($\lambda = 1.0$) is fixed.

Order	Acc. (%)	traj ∩ obj	traj ∩ scene	obj ∩ scene	sum
TSO	**86.92**	10	9	20	39
TOS	86.78	8	9	22	39
STO	86.64	12	16	12	40
SOT	86.50	13	7	**9**	**29**
OTS	86.15	14	8	12	34
OST	85.73	12	13	11	36

Table 20: **Effects of individual concept types on `TraCE` performance.**

Concept			Acc.
Trajectory	Object	Scene	
✓	✗	✗	80.7
✓	✓	✗	82.4
✓	✗	✓	81.6
✗	✓	✓	74.6
✓	✓	✓	84.2

Even at NEC = 5, where each class uses only five concepts on average, accuracy remains at 67.7%. These results show that a compact set of concepts per class retains much of the model's predictive performance.

F.3 Explanation Analysis

Motion perturbation analysis. We complement the perturbation analysis in Section 4.3 with direction-specific results on the same directional-class subset of Chiral SSv2. We evaluate classes specifying left, right, up, or down motion under temporal reversal and 90° counter-clockwise rotation. As shown in Figure 7, `TraCE` adapts its top trajectory concepts to the induced motion transformation. For the original input in Figure 7 (a), the top concepts align with the left-to-right motion and support the prediction. When the video is temporally reversed in Figure 7 (b), the top concepts switch to the opposite direction and the prediction changes accordingly. Under a 90° rotation in Figure 7 (c), the activated concepts rotate consistently with the transformed motion direction. For each test sample, we apply the perturbation and check whether the predicted motion group follows the expected directional transformation: temporal reversal should swap LEFT ↔ RIGHT and UP ↔ DOWN, while 90° rotation should map LEFT → DOWN, RIGHT → UP, UP → LEFT, and DOWN → RIGHT. As shown in Table 22, `TraCE` achieves an overall reverse consistency of 79.2% and a rotation consistency of 63.1%. Reverse consistency is higher for LEFT (90.4%), RIGHT (84.0%), and DOWN (94.0%) than for UP (55.8%). Rotation consistency is also higher for LEFT (83.3%) and RIGHT (90.8%) than for UP (44.2%) and DOWN (45.2%). The prediction-level results support motion sensitivity and complement the concept-level directional analysis in Appendix G.2.

Faithfulness analysis. We extend the faithfulness evaluation in Section 4.3 of the main paper by reporting per-dataset concept removal results in Table 23. For each sample, we deactivate either the top-K trajectory concepts ranked by contribution (Top-K) or K randomly selected trajectory concepts (Random-K), and measure Flip%, the percentage of samples whose predicted class changes. A faithful model should show high Flip% under Top-K removal and low Flip% under Random-K removal, indicating that the contribution ranking aligns with the model's actual decision process. Across all datasets, Top-K removal causes substantially more prediction flips than Random-K removal at both $K = 5$ and $K = 10$. Chiral SSv2 shows the largest Top-K effect (48.7% at K=5, 70.1% at K=10), consistent with the strong dependence of directional motion classes on a small number of discriminative trajectory concepts. NTU RGB+D Mutual shows the smallest effect (7.0% at K=5), reflecting the near-saturated nature of this benchmark where multiple trajectory concepts carry redundant evidence. UCF-101 shows a moderate Top-K effect (15.9% at K=5), consistent with the observation that many UCF-101 classes can be supported by static scene or object cues alongside

Table 21: **Accuracy under controlled Number of Effective Concepts (NEC) [58].** NEC limits the average number of non-zero classifier weights per class. Lower NEC yields sparser, more interpretable predictions at the cost of accuracy. Sparsity denotes the fraction of non-zero entries in $\mathbf{W}_{\text{AR}}$.

NEC	Sparsity (%)	Acc. (%)
5	0.63	67.7
10	1.21	77.2
15	1.79	81.3
20	2.41	82.9
25	3.03	83.8
30	3.57	84.0

Table 22: **Prediction consistency under perturbation on Chiral SSv2.** We report the percentage of samples whose predicted motion group follows the expected transformation after perturbation. Expected transformations are: LEFT $\leftrightarrow$ RIGHT and UP $\leftrightarrow$ DOWN under temporal reverse, and LEFT $\rightarrow$ DOWN, RIGHT $\rightarrow$ UP, UP $\rightarrow$ LEFT, DOWN $\rightarrow$ RIGHT under 90° counter-clockwise rotation.

Original group	Reverse consistency (%)	Rotation consistency (%)
Left	90.4	83.3
Right	84.0	90.8
Up	55.8	44.2
Down	94.0	45.2
Overall	79.2	63.1

trajectory evidence. In all cases, Random-K removal causes minimal prediction changes ($\leq 5.4\%$), confirming that random trajectory concepts do not carry decision-critical information.

F.4 Video Question Answering Analysis

The VQA results in Table 2 show that `TraCE` preserves competitive performance when predictions are routed through the typed trajectory/object/scene interface. We further evaluate whether predictions on the native TOMATO and STAR benchmarks depend on trajectory concepts that contribute to the predicted answer. For each sample, we rank trajectory concepts by their contribution to the concept-compatibility score for the predicted answer and set the selected trajectory concept activations to zero. We remove concepts in descending (Top), random, or ascending (Bottom) order of contribution.

Table 24: **Trajectory concept removal on native VQA benchmarks.** For each sample, we remove the indicated fraction of trajectory concepts in descending (Top), random, or ascending (Bottom) order of contribution and report answer accuracy (%).

	TOMATO			STAR		
Removal ratio	Top	Random	Bottom	Top	Random	Bottom
0%	38.0	38.0	38.0	53.4	53.4	53.4
5%	26.1	33.7	38.0	12.9	52.1	53.4
10%	18.5	33.7	38.0	14.2	50.9	53.4
20%	16.3	33.7	38.0	11.7	48.8	53.4
40%	16.3	27.2	38.0	10.6	44.9	53.4
60%	19.6	30.4	38.0	11.5	40.3	53.4
80%	17.4	26.1	38.0	14.8	33.2	53.4
100%	23.9	23.9	23.9	24.7	24.7	24.7

As shown in Table 24, removing only the top-5% trajectory concepts reduces accuracy by 11.9 points on TOMATO and 40.5 points on STAR. The corresponding random removals reduce accuracy by 4.3 and 1.3 points, while Bottom-ranked removal has no effect through an 80% removal ratio.

Table 23: **Faithfulness of trajectory concepts via concept removal.** We report Flip%, the percentage of samples whose prediction changes after deactivating trajectory concepts. We compare removing the top-K contributing concepts with removing K randomly selected concepts. Higher Top-K removal and lower Random-K removal indicate that the estimated concept importance aligns with the model's actual decision process.

	K=5		K=10	
Dataset	Top-K ↑	Random-K ↓	Top-K ↑	Random-K ↓
Chiral SSv2	48.7	3.7	70.1	5.4
KTH	35.1	2.1	46.0	2.8
UCF-101	15.9	0.2	18.4	0.3
NTU RGB+D Mutual	7.0	0.4	13.8	0.3
Average	26.7	1.6	37.1	2.2

These results show that native VQA predictions depend on a sparse set of trajectory concepts with high contribution to the predicted answer rather than on arbitrary bottleneck dimensions. At 100% trajectory concept removal, object and scene concepts remain active, accounting for the nonzero residual accuracy.

The interpretable baselines evaluated in the main paper are designed for action recognition and do not directly define question-conditioned answer scoring. We therefore construct two controlled variants that combine the MM-CBM [54] concept-space matching design with language-derived motion concepts from LF-CBM [40] or pose-sequence concepts from DANCE [26]. Because neither pose-sequence nor trajectory concepts have language names, the pose-sequence variant learns text-side relevance using the same answer-supervised alignment as `TraCE`. We additionally adapt SpLiCE [9] to question-conditioned answer scoring. All methods use the same frozen video and text backbones, data splits, answer candidates, and training budget.

Table 25: **Comparison with VQA-capable concept methods.** We report answer accuracy (%) as mean ± sample standard deviation over five stochastic training seeds, except for the single-run SpLiCE adaptation. The controlled variants combine the MM-CBM [54] concept-space matching design with language-derived motion concepts from LF-CBM [40] or pose-sequence concepts from DANCE [26]. The first three datasets are multiple-choice VQA reformulations of action-recognition datasets; TOMATO and STAR are native VQA benchmarks.

Method	NTU Mutual	Chiral SSv2	UCF-101	TOMATO	STAR	Avg.
Non-interpretable baseline	97.4±0.1	82.6±0.2	99.8±0.0	35.9±2.2	54.5±0.4	74.0
Language-derived motion concepts	97.6±0.1	82.0±0.4	99.7±0.1	31.7±3.6	56.5±0.5	73.5
Pose-sequence concepts	97.6±0.1	82.3±0.6	99.7±0.1	28.7±7.0	55.9±0.7	72.8
SpLiCE [9] adaptation	53.0	57.3	86.7	22.4	49.2	53.7
`TraCE`	97.1±0.3	82.6±0.6	99.7±0.1	37.0±1.7	55.4±1.2	**74.4**

As shown in Table 25, `TraCE` achieves the highest mean accuracy across the five benchmarks among the concept-based methods (74.4%), compared with 73.5% for language-derived motion concepts and 72.8% for pose-sequence concepts. On the native VQA benchmarks, `TraCE` obtains the highest TOMATO accuracy (37.0%) and remains competitive on STAR (55.4%). We do not interpret the small numerical differences as significant improvements. Together with the deletion results, these comparisons show that trajectory concepts support native VQA explanations while maintaining competitive predictive performance under the same concept-space matching design.

Concept combinations on STAR. We evaluate combinations of trajectory, object, and scene concepts on STAR [67]. We report single-run results for both STAR ablations in Table 26 and Table 27. As shown in Table 26, trajectory concepts alone achieve 54.8% accuracy, compared with 54.4% for object concepts and 52.9% for scene concepts. Combining all three concept types yields 53.4%. Adding static context does not improve the reported accuracy over trajectory concepts alone in this evaluation.

Table 26: **Effects of individual concept types on `TraCE` performance.**

Concept			Acc.
Trajectory	Object	Scene	
✓	✗	✗	54.8
✗	✓	✗	54.4
✗	✗	✓	52.9
✓	✓	✗	54.5
✓	✗	✓	52.9
✗	✓	✓	53.4
✓	✓	✓	53.4

Table 27: **Effect of text-side trajectory relevance learning objectives.**

Method	Acc.
InfoNCE	53.4
Cosine similarity	52.0
KL divergence	52.6
ℓ_1 regression	52.1
ℓ_2 regression	52.2

Text-side trajectory relevance learning on STAR. We compare loss functions for learning question-conditioned trajectory relevance on STAR [67]. As shown in Table 27, InfoNCE achieves the highest accuracy (53.4%), while the alternative objectives yield 52.0–52.6%. The comparison supports our choice of InfoNCE for answer-supervised trajectory alignment.

G Additional Semantic and Robustness Analyses

G.1 Sample-Level Motion-Source Grounding

The qualitative examples in Figure 4 show that trajectory concepts explain interactions and object dynamics, while Table 3 shows that the trajectory-only variant retains predictive motion cues under object category shift. We quantify whether high-contribution trajectory concepts are spatially grounded in the hands and manipulated objects in each video.

We use the same object-disjoint Chiral SSv2 split constructed from Something-Else annotations [38]. As described in Appendix A.5, we discover trajectory concepts exclusively from filtered training trajectories and retain the medoid trajectory of each training cluster. For each validation video, we assign every cached trajectory to its nearest fixed medoid in the same flattened displacement space used for clustering. The cached validation assignments are used to compute the validation loss and select the checkpoint, but validation trajectories do not participate in vocabulary construction or gradient updates. We rank trajectory concepts by their contribution to the predicted class, select the trajectories assigned to the Top-10, Random-10, or Bottom-10 concepts, and recover their seed pixels in the input video. For Random-10, we sample one set of ten concepts uniformly without replacement for each validation video and aggregate coverage over the validation set. Because every assigned trajectory retains its seed pixel, this analysis directly recovers the spatial support of each selected concept without Grad-CAM.

Something-Else provides frame-level bounding boxes for hands and manipulated objects. A selected trajectory covers a box only when its seed pixel lies inside the box in the first frame of the corresponding L-frame clip. *Hand* and *Object* measure the percentages of annotated hand and object boxes covered by at least one selected trajectory. *Both* measures the percentage of videos in which the selected trajectories cover at least one annotated hand and one manipulated object. We evaluate DANCE [26] analogously using the spatial support of its selected pose-sequence concepts.

Table 28: **Sample-level motion-source grounding on object-disjoint Chiral SSv2.** Top and Bottom select $K = 10$ concepts according to their contribution to the predicted class; Random samples one set of $K = 10$ concepts uniformly without replacement for each video. *Hand* and *Object* report the percentages of annotated boxes covered by the seed pixel of a selected trajectory in the first frame of its L-frame clip; *Both* reports the percentage of videos covering at least one hand and one manipulated object.

	DANCE [26]			`TraCE`		
Selection	Hand	Object	Both	Hand	Object	Both
Bottom-10	0.0	0.0	0.0	22.4	21.6	19.0
Random-10	4.1	0.7	0.5	29.1	28.5	25.1
Top-10	42.3	7.6	4.7	**64.5**	**63.5**	**58.1**

As shown in Table 28, source coverage increases with concept contribution for `TraCE`. The Top-10 trajectory concepts cover 64.5% of annotated hand boxes and 63.5% of manipulated-object boxes, with both sources covered in 58.1% of videos. The joint coverage decreases to 25.1% for Random-10 concepts and 19.0% for Bottom-10 concepts. In contrast, DANCE's Top-10 pose-sequence concepts cover 42.3% of annotated hand boxes but only 7.6% of manipulated-object boxes and both sources in 4.7% of videos. These results do not imply that each trajectory concept has a fixed entity label. Rather, they show that high-contribution trajectory concepts can be grounded at the sample level to motion from both people and manipulated objects, supporting their role as a source-flexible motion vocabulary.

G.2 Directional Semantic Validation

As shown in Table 4, `TraCE`'s predictions follow the expected directional transformations under temporal reversal and rotation. Prediction consistency alone, however, does not show whether the selected trajectory concepts carry the corresponding motion semantics. Because trajectory concepts are supervised by a soft distribution over multiple training-derived clusters, single-label precision or recall would discard the target structure. We instead evaluate signed motion direction on the Chiral SSv2 direction subset ($n = 572$) [5, 18], whose actions specify *left*, *right*, *up*, or *down* motion.

We assign each trajectory concept a fixed direction label from the net displacement of its medoid trajectory before evaluating the predicted concept activations. Across held-out videos, the highest-activated trajectory concept matches the ground-truth direction in 67.5% of cases, compared with the 25% four-way random reference.

For each original, temporally reversed, and 90°-rotated video, we measure whether the top-5 activated trajectory concepts follow the expected direction. Let C^5_{orig} and C^5_{pert} denote the sets of top-5 concept indices for the original and perturbed videos. We measure concept replacement as $1 - |C^5_{\text{orig}} \cap C^5_{\text{pert}}|/5$. For each transformed video, prediction correctness is determined using the expected transformed class.

Table 29: **Directional semantics under controlled perturbations.** Direction agreement is the fraction of top-5 activated trajectory concepts whose fixed direction labels, derived from their medoid trajectories, match the expected direction. For the original and perturbed top-5 concept-index sets C^5_{orig} and C^5_{pert}, replacement is $1 - |C^5_{\text{orig}} \cap C^5_{\text{pert}}|/5$. Prediction correctness after perturbation is evaluated against the expected transformed class; the final row reports random-reference values.

Prediction	Input	Direction agreement (%)	Replacement (%)
Correct	Original	76.3	0.0
Correct	Temporal reversal	**80.1**	**96.3**
Correct	90° rotation	**78.4**	**96.1**
Incorrect	Original	45.4	0.0
Incorrect	Temporal reversal	32.8	81.0
Incorrect	90° rotation	35.1	84.6
Random reference	–	25.0	96.8

As shown in Table 29, correctly predicted transformed videos replace 96.3% of the original top-5 concept identities under temporal reversal and 96.1% under rotation. The selected concepts achieve 80.1% direction agreement after reversal and 78.4% after rotation. Incorrect transformed predictions provide a control: they also exhibit substantial concept replacement, but their direction agreement decreases to 32.8% under reversal and 35.1% under rotation. Concept replacement alone therefore does not establish semantic consistency. Correct transformed predictions instead combine near-complete concept replacement with concepts whose fixed training-derived directions match the transformed motion at more than three times the 25% random reference. This analysis validates signed motion direction, where objective physical labels are available, rather than providing exhaustive semantic annotation or calibration for every fine-grained trajectory cluster.

G.3 Ground-Truth-Targeted Concept Interventions

The contribution-ranked removal results in Figure 8 and Table 23 show that `TraCE` relies on high-contribution trajectory concepts. However, contribution-ranked removal does not determine whether the influential concepts align with the ground-truth motion. We therefore perform direction-targeted interventions on the Chiral SSv2 direction subset ($n = 572$) and the TOMATO direction subset ($n = 28$), using the fixed directional concept labels defined in Appendix G.2. For each sample, we remove concepts from the ground-truth-direction, opposite-direction, or random non-ground-truth pool. The random pool excludes concepts assigned to the ground-truth direction.

We consider two complementary interventions. In count-matched removal, for sample i, let G_i denote the set of ground-truth-direction concepts and O_i the set of opposite-direction concepts. We set $k_i = \min(|G_i|, |O_i|)$ and remove exactly k_i concepts from the ground-truth-direction, opposite-direction, or non-ground-truth pool. The non-ground-truth pool contains only opposite or orthogonal directions. When a pool contains more than k_i concepts, we randomly subsample k_i concepts and average the result over 100 draws. Thus, all three conditions remove the same number of concepts for each sample, and the random control never includes ground-truth-direction concepts. When $|G_i| > k_i$, some ground-truth-direction concepts remain active, making this a conservative test that controls intervention size without eliminating all same-direction redundancy.

Full-group removal instead deactivates every concept assigned to the ground-truth or opposite direction, testing the effect of removing potentially redundant concepts with the same directional semantics. Its random non-ground-truth control removes $|G_i|$ concepts from the non-ground-truth

pool and is averaged over 100 draws. The reported standard deviations therefore measure variability across concept-subsampling draws on the fixed evaluation set, not variability across training runs. We report post-intervention accuracy and the correct-to-incorrect flip rate, the percentage of initially correct predictions that become incorrect after intervention.

Table 30: **Ground-truth-targeted trajectory concept interventions.** We report post-intervention accuracy, its change from the unablated model, and the flip rate, i.e., the percentage of initially correct predictions that become incorrect. Count-matched results and the random non-ground-truth full-group results report mean ± standard deviation over 100 concept-subsampling draws on the fixed evaluation set.

Protocol	Task	Removed concepts	Acc. (%)	ΔAcc.	Flip rate (%)
Count-matched	AR	None (TraCE)	81.1	–	–
		GT direction	56.9±0.1	−24.2	30.1±0.1
		Opposite direction	82.2±0.1	+1.1	3.6±0.1
		Random non-GT	82.1±0.2	+1.0	3.7±0.2
	VQA	None (TraCE)	46.4	–	–
		GT direction	23.9±0.4	−22.5	70.5±0.5
		Opposite direction	54.0±0.2	+7.6	30.8±0.0
		Random non-GT	45.4±1.3	−1.0	21.3±2.0
Full-group	AR	None (TraCE)	81.1	–	–
		GT direction	53.7	−27.4	34.1
		Opposite direction	81.3	+0.2	4.5
		Random non-GT	82.8±0.2	+1.6	3.8±0.1
	VQA	None (TraCE)	46.4	–	–
		GT direction	25.0	−21.4	69.2
		Opposite direction	53.6	+7.2	30.8
		Random non-GT	42.2±1.4	−4.2	25.4±2.1

As shown in Table 30, ground-truth-direction removal produces the largest degradation under both protocols. On Chiral SSv2, count-matched removal reduces accuracy by 24.2 points and changes 30.1% of originally correct predictions to errors, whereas opposite-direction and random non-ground-truth removal do not reduce accuracy and yield flip rates below 4%. Full-group removal produces the same ordering, reducing accuracy by 27.4 points and yielding a 34.1% correct-to-incorrect flip rate. On TOMATO, ground-truth-direction removal reduces accuracy by 22.5 points under count matching and 21.4 points under full-group removal, with corresponding flip rates of 70.5% and 69.2%. Only 13 TOMATO samples are predicted correctly before intervention, so these VQA flip rates should be interpreted descriptively. A positive net accuracy change under a control can coexist with a nonzero correct-to-incorrect flip rate because overall accuracy also includes initially incorrect predictions that become correct. Together with the directional analysis in Appendix G.2, these results show that trajectory concepts aligned with the ground-truth motion contribute disproportionately to both action recognition and VQA predictions.

G.4 Foreground Masking and Camera Motion Diagnostics

The trajectory filter in Appendix A.5.1 retains a trajectory only when its seed pixel lies inside the estimated foreground saliency mask. On Chiral SSv2, foreground masking removes approximately 81% of the raw trajectories. We ablate this component over all Chiral SSv2 classes while keeping the remaining pipeline unchanged.

Table 31: **Effect of foreground masking on Chiral SSv2.** We keep the remaining trajectory discovery and learning pipeline unchanged and report validation accuracy (%).

Foreground masking	Accuracy (%)
On (TraCE)	**84.2**
Off	80.3

As shown in Table 31, foreground masking improves validation accuracy from 80.3% to 84.2%. The 3.9-point gain supports the use of foreground masking to suppress background-seeded trajectories. Foreground masking is not equivalent to camera ego-motion compensation: the foreground mask can be imperfect, and global camera motion can still affect trajectories seeded on foreground entities.

We separately test whether trajectory concepts can represent global camera motion when camera motion itself defines the target action. The diagnostic subset contains 65 Chiral SSv2 videos from six camera motion classes. We report classification accuracy and the entropy of the activated trajectory concepts' four-direction distribution, $H = -\sum_{d\in\{\mathrm{L,R,U,D}\}} p_d \log p_d$, where p_d is the normalized mass of this distribution assigned to direction d. We use the natural logarithm, so $H = 0$ indicates concentration in one direction and the maximum is $\log 4 \approx 1.386$.

Table 32: **Camera motion diagnostic on Chiral SSv2.** We report the number of videos, classification accuracy (%), and entropy of the activated concepts' four-direction distribution; the maximum entropy is $\log 4$.

Camera motion class	n	Accuracy (%)	Four-direction entropy
Turning the camera left while filming	16	93.8	0.00
Turning the camera right while filming	18	100.0	0.00
Turning the camera downward while filming	5	100.0	0.00
Turning the camera upward while filming	10	100.0	0.00
Approaching something with the camera	8	100.0	0.72
Moving away from something with the camera	8	100.0	1.37
All camera motion classes	65	**98.5**	–

As shown in Table 32, `TraCE` achieves 98.5% accuracy across the six camera motion classes. The left/right and upward/downward turning classes concentrate their activations in one directional group, whereas approaching and moving-away classes produce higher entropy consistent with radial expansion and contraction. This diagnostic shows that trajectory concepts can represent task-relevant camera motion; it does not establish invariance when camera motion is a nuisance. When entity motion and camera motion coexist, their optical-flow components may mix, motivating future work on video stabilization, global-motion subtraction, or joint motion decomposition.

G.5 Static-Context Stability and Baseline Clarification

The language-based baseline in Figure 8 is a learned CBM using language-derived motion concepts, not GPT-5.4 as a video classifier. GPT-5.4 is used only offline to generate a vocabulary of language-derived motion concepts. This baseline replaces `TraCE`'s trajectory branch with these concepts while holding the frozen backbone, object and scene concepts, sparse classifier, data splits, and evaluation protocol fixed. Each model ranks and removes concepts using its own contribution scores. We report the unablated accuracies corresponding to the zero-removal point of each curve in Table 33.

Table 33: **Unablated accuracies for the deletion curves in Figure 8.** We report the single-run accuracy (%) at the zero-removal point of each curve.

Dataset	Language-based baseline	`TraCE`
NTU RGB+D Mutual	97.0	**97.5**
Chiral SSv2	77.2	**84.2**
UCF-101	88.2	**88.6**
KTH	90.2	**92.0**

On UCF-101, the language-based baseline starts below `TraCE` (88.2% vs. 88.6%), so its stronger deletion effect in Figure 8 does not result from a higher initial accuracy. We interpret this result conservatively: the top-ranked language-derived motion concepts carry substantial predictive information on UCF-101, potentially including static contextual cues from objects or scenes. The deletion curves evaluate whether contribution rankings identify concepts used for prediction; the direction-targeted interventions in Appendix G.3 separately evaluate alignment with ground-truth motion.

The typed interface in Figure 2 separates dynamic trajectory concepts from static object and scene concepts. Using the same trained model and Chiral SSv2 direction subset as Appendix G.2, we test whether temporal reversal and rotation alter the semantics of the selected object and scene concepts. For each concept group, we select the top-k activated concepts before and after perturbation, average their embedding vectors, and compute the cosine similarity between the two aggregate embeddings. This metric evaluates aggregate semantic stability rather than exact concept-identity invariance.

Table 34: **Semantic stability of object and scene concepts under motion perturbations.** On the Chiral SSv2 direction subset, we average the concept embeddings of the top-k activated concepts in each object or scene group and report cosine similarity between the original and perturbed aggregate embeddings.

Perturbation	Concept group	$k=1$	$k=3$	$k=5$	$k=10$
Temporal reversal	Object	0.84	0.94	0.96	0.98
	Scene	0.85	0.94	0.96	0.98
90° rotation	Object	0.81	0.92	0.95	0.97
	Scene	0.83	0.93	0.95	0.97

As shown in Table 34, the selected object and scene concepts remain largely semantically stable under both perturbations. Cosine similarity ranges from 0.84 to 0.98 under temporal reversal and from 0.81 to 0.97 under rotation. These results complement the directional shifts in selected trajectory concepts in Appendix G.2: motion perturbations alter the selected directional content while largely preserving the aggregate semantics of static context.

H Limitations

While `TraCE` provides effective and interpretable explanations by disentangling trajectory, object, and scene concepts, it has several limitations.

First, trajectory concepts depend on the quality of the underlying optical flow estimator. Inaccurate flow predictions in challenging conditions—such as occlusion, fast motion, or low-texture regions—can produce noisy or misleading trajectories, which in turn degrade the quality of the discovered trajectory concepts. Moreover, optical-flow trajectories can mix foreground entity motion with global camera motion. Although the diagnostic in Appendix G.4 shows that trajectory concepts can represent task-relevant camera motion, it does not establish invariance to camera motion when it acts as a nuisance.

Second, due to the absence of explicit object and scene annotations in most video datasets, `TraCE` relies on a Video-LLM to generate concept candidates and on a pretrained vision–language encoder to produce soft pseudo-labels. Consequently, the resulting object and scene representations may inherit limitations of the underlying models, such as language priors and visual biases. Third, our concept visualizations rely on Grad-CAM [50] to localize where a trajectory concept is expressed, so the highlighted regions are approximate. Fourth, trajectory concepts are extracted from short clips (e.g., $L = 16$ frames), which limits their ability to represent long-range temporal structure.

To mitigate sensitivity to optical flow quality and nuisance camera motion, future work could explore more robust flow estimators, video stabilization, global-motion subtraction, joint motion decomposition, or alternative trajectory representations that do not depend on dense flow fields. To improve visualization precision, learned spatial grounding or attention-based localization could replace Grad-CAM [50] for concept overlay. Extending `TraCE` to long-range temporal reasoning may involve generalizing trajectory concepts to broader visual dynamics, such as modeling temporal interactions among scene elements beyond pixel-level flow.

I Broader Impacts

This work aims to advance explainability in video understanding by representing dynamic evidence as trajectory concepts and separating it from static object and scene context. By deriving motion concepts from optical-flow trajectories, `TraCE` enables source-flexible explanations that cover people, manipulated objects, and multi-agent interactions within a single framework. This broader coverage makes trajectory-based explanations applicable to a wider range of real-world scenarios, such

as object-centric manipulation and multi-agent environments, and may extend to egocentric video. Furthermore, the same typed concept interface supports both action recognition and video question answering, demonstrating that concept-level explanations need not be redesigned for each task. By exposing whether a model relies on meaningful motion cues, static contextual shortcuts, or spurious correlations, `TraCE` can help practitioners diagnose and correct model errors through concept-level intervention, promoting trustworthy deployment in high-stakes domains such as healthcare, education, and human–AI interaction. We hope this line of research fosters safer, more interpretable, and human-aligned video AI systems. At the same time, explanations can be over-trusted: a plausible concept explanation does not by itself guarantee correct model behavior, so explanations should complement rather than replace rigorous evaluation. In addition, improved action recognition and interpretability tools could be misused for surveillance; we encourage their use in accordance with applicable privacy regulations and ethical guidelines.

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 accurately summarize the contributions and scope of the paper. The claims are supported by experimental results, as detailed in Section 4.

   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 limitations of our work are thoroughly discussed in the dedicated "Limitations" section in the appendix (Appendix H).

   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: The paper does not include any 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 paper provides all necessary details to reproduce the main experimental results, including detailed descriptions of the dataset, model architecture, and evaluation metrics. These are thoroughly documented in the appendix.

   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: [Yes]

Justification: The implementation and instructions for reproducing the main experiments are publicly available at https://github.com/KHU-VLL/TraCE. Additional qualitative results are available on the project page at https://leewooil.github.io/TraCE_project_page/. All datasets used in this paper are publicly available from their original sources, and the appendix documents the experimental settings and evaluation protocols.

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 paper specifies all the necessary training and test details, including data splits, hyperparameters, and the type of optimizer used. These details are presented in the appendix.

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: [Yes]

Justification: The main performance table reports the mean and sample standard deviation over five random seeds for each dataset. The table caption defines the reported variability, and the text avoids interpreting the nominal VQA difference as a significant improvement.

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: We report compute resource details—including GPU type and per-experiment training configuration—in the appendix. This includes hardware specifications for each stage of training, ensuring reproducibility.

   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: The research conducted in this paper fully conforms with 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 discussion of broader impacts in the appendix. Specifically, we highlight the positive societal impact of improving the interpretability of video understanding models through source-flexible trajectory-based explanations, which can enhance transparency and trust in applications such as healthcare, education, and safety-critical systems. We emphasize that the primary goal of this work is to support more interpretable and trustworthy video understanding systems.

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: [N/A]

Justification: The paper does not involve the release of data or models with high risk for misuse, hence this question is not applicable.

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 use publicly available datasets (NTU RGB+D [52] Mutual under the ROSE Lab academic license, UCF-101 [57] and KTH [49] under their academic licenses, STAR [67] under Apache 2.0, and Chiral SSv2 [5, 18] and TOMATO [53] available

on Hugging Face). We also use the pretrained VideoMAE [60] encoder (CC-BY-NC 4.0), InternVideo2 [63](Apache 2.0), WAFT optical flow estimator [64] (BSD-3-Clause), U2Net [42] (Apache 2.0), and Qwen3-VL [6] (Apache 2.0). All sources are cited, and licenses are respected.

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: We curate a subset of the NTU RGB+D dataset (NTU RGB+D Mutual, NTU RGB+D Single) to facilitate interpretable evaluation. The curation process and selection criteria are clearly described in the appendix.

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: [Yes]

Justification: The paper includes a user study with human subjects to evaluate the interpretability of concept-based explanations. Key methodological details are provided in the main paper. The supplementary material includes full task instructions, interface screenshots, and compensation information.

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: [Yes]

    Justification: The study involving human subjects was reviewed by an IRB and received an exemption determination based on minimal risk and anonymized data collection. No personally identifiable information was collected, and all participants provided informed consent prior to participation.

    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: We employed a Video-LLM, specifically Qwen3-VL, to generate candidate object and scene concepts that form part of our concept vocabulary. These LLM-generated concepts serve as human-interpretable descriptors used in our explanation framework. The LLM was not involved in model training or prediction but contributed to the construction of interpretable object and scene concepts, which are integral to our method.

    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.