ACTIONREFINER: REFINING PREDICTIONS WITH INVERSE VERBS FOR ACTION RECOGNITION IN EGOCENTRIC VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite advances in egocentric action recognition using vision-language models, distinguishing actions with identical nouns but inverse verbs (e.g., "open fridge" vs. "close fridge") remains challenging because these actions often only differ in their temporal order of execution. We present *ActionRefiner*, an approach to improve action recognition via a Causal Temporal Generation Process (CTGP) and a residual classification module. Given action intermediate spatio-temporal features, CTGP learns latent temporal representations for verbs and nouns conditioned on action context via VAE-based modelling and a domain estimator. A residual classifier then produces corrective logits that are fused with the baseline logits to predict actions. ActionRefiner consistently improves strong baselines and establishes new state-of-the-art performance on Epic-Kitchens-100 and EGTEA Gaze+. In particular, it achieves a new state-of-the-art 57.7 Top-1 accuracy on Epic-Kitchens-100 and 82.7 on EGTEA Gaze+. Notably, the majority of improvements over LaViLa on EGTEA Gaze+ is on inverse-verb actions, demonstrating the importance of learning action-aware temporal representations of verbs and nouns.

1 INTRODUCTION

Understanding human actions from videos is a core challenge in computer vision with a wide range of applications ranging from robotics and planning Jang et al. (2020); Li et al. (2021); Shi et al. (2025) to assistive technologies Guo et al. (2024). Action recognition in egocentric videos in particular has attracted increasing attention Huang et al. (2020); Kazakos et al. (2021); Min & Corso (2021); Yan et al. (2022) and recent improvements has mainly been driven by increasingly powerful spatio-temporal architectures and large-scale training paradigms Zhao et al. (2023); Zhao & Krähenbühl (2023); Chalk et al. (2024).

Despite these advances, distinguishing actions with same nouns but inverse verbs (e.g., "open fridge" vs. "close fridge") remains challenging. Such actions share the same visual context and objects but differ in subtle temporal cues, such as motion direction or ordering. As a result, even state-of-the-art transformer-based vision-language models, such as LaViLa Zhao et al. (2023) exhibit a significantly higher error rate on inverse-verb actions (20%) compared to other actions (14.7%) on EGTEA Gaze+ dataset Li et al. (2018). To address this challenge, previous work explored individual network branches to learn verb and noun representations separately Wang et al. (2020; 2021) and merge noun and verb probabilities to predict actions. However, these works learn verbs and nouns alone, it does not account for the impact of full action context which limits the ability to capture the underlying dynamics of the scene.

To address these limitations, we propose to incorporate verb and noun representations that are aware of action context, More specifically, we aim to model the causal effect of full actions on verb and noun dynamics and capture their dependencies in the temporal generation process. We present **ActionRefiner**, a method that leverages causal temporal generation processes of verb and noun to refine predictions in egocentric action recognition tasks. Our framework is designed as a plug-in module that can be integrated with any method producing intermediate spatio-temporal representations from videos. More specifically, we first finetune two visual encoders to predict verb and noun separately

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and extract temporal features for verb and noun. We also extract action features from an existing method trained to predict action. Next, the Causal Temporal Generation Processes (CTGP) learn latent temporal representations for verb and noun. Specifically, we deploy VAE-based networks to model the causal temporal generation process for verb and noun in the latent space. A domain estimator learns action context from the action features and uses it as domain information to guide the generation process for verb and noun. Finally, a residual classification module uses temporal latent representations of verb and noun to generate action logits. These action logits are then merged with the logits from the existing baseline model, to refine the final action prediction.

We evaluate our method on Epic-Kitchens-100 Damen et al. (2022) and EGTEA Gaze+ Li et al. (2018) datasets and show that it achieves new state-of-the-art performance on both datasets when plugged into a strong baseline. Specifically, using TIM Cai et al. (2025) as the action prediction baseline model, ActionRefiner achieves 57.7 top-1 accuracy on Epic-Kitchens-100, with an improvement of 1.3 Top-1 accuracy. Similarly, on EGTEA Gaze+ with LaViLa Zhao et al. (2023) as the baseline, our method achieves an improvement of 0.9 in Top-1 accuracy (82.7 Top-1 accuracy overall). These results show that ActionRefiner provides a generalizable correction mechanism that enhances action recognition performance.

Our contributions are threefold: First, we propose ActionRefiner, a plug-in method that leverages action-aware verb and noun representations to refine predictions and improves multiple baselines. ActionRefiner also achieves SOTA performances on Epic-Kitchens-100 and EGTEA Gaze+ in egocentric action recognition. Second, we show CTGP can learn better representations than generic temporal modeling methods by modeling verb and noun latent dynamics conditioned on action context. Third, our qualitative results show that our method improves the cases that have inverse verbs more than the other cases. This demonstrates that ActionRefiner can better capture subtle dynamics, leading to more semantically consistent predictions where prior methods often fail.

2 Related Work

2.1 Action Recognition

LaViLa Zhao et al. (2023) used a LLM-based narrator and a LLM-based rephraser to augment training data and trained the vision encoder before finetuning the downstream action recognition datasets. AVION Zhao & Krähenbühl (2023) adapted the same method as LaViLa but with additional training techniques to speed up the training speed. TIM Chalk et al. (2024) used video and audio information and combined the two modality in one transformer for action recognition. Unlike previous methods, it used longer video clips which contained part of action before and after current action. This makes TIM has the advantage of seeing more action context and video context. Llavaction Qi et al. (2025) proposed to use multi-modal large language models (MLLMs) for action recognition. Although it obtained higher accuracy than TIM on Epic-Kitchens-100, Llavaction reformulated the task in multiple choice questions. Specifically, the model is prompted to select the correct action among 20 choices. Epic-Kitchens-100, on the other hand, has 3806 action class, Llavaction significantly reduced the search space of action, hence making the recognition task much easier.

2.2 Causal Temporal Representation Learning

Causal representation learning aims to identify causal variables from observation data such as image pixels Schölkopf et al. (2021); Ahuja et al. (2023). Temporal causal representation learning Song et al. (2024); Lippe et al. (2022) extends it to identify the latent causal process from time series observations. A critical aspect in temporal causal representation learning is identifiability. Previous works have shown that identifiability can be established under mild assumptions Yao et al. (2022); Li et al. (2024b); Song et al. (2024) with variant task domains from simple Cartpole to human motion, instructional videos and so on. Causality-based methods have drawn attention in the tasks related to actions in videos such as domain adaptation for motion Rastegar et al. (2024); Cai et al. (2025), few-shot and zero-shot action recognition Li et al. (2024a); Wang et al. (2023), targeting action detection in long video Xu et al. (2025) and group activity recognition Zhang et al. (2024).

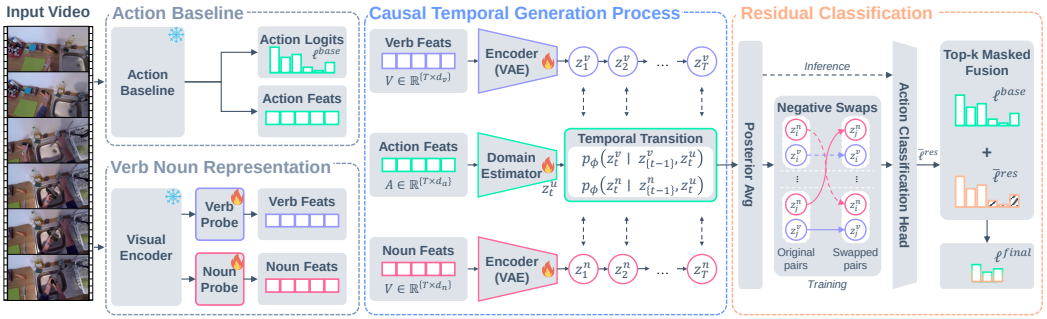


Figure 1: Overview of ActionRefiner. **Action Baseline:** Given an egocentric video clip, a frozen action baseline provides action logits $\boldsymbol{\ell}^{\text{base}}$ and temporal action features. **Verb Noun Representation:** visual encoders are finetuned to predict verb and noun and verb and noun temporal features are extracted. **Causal Temporal Generation Process:** the verb and noun features are encoded by separate VAEs into latent sequences. A domain estimator derives an action-context signal from action features and conditions the temporal transition priors of both latent streams. **Residual Classification:** The posterior averaging module samples multiple latent representations of verb and noun and averages them. Negative verb–noun pairs are only created within the batch during training to regularize the latent space; this step is skipped during inference. The action prediction head generates $\bar{\boldsymbol{\ell}}^{\text{res}}$, which is then fused with $\boldsymbol{\ell}^{\text{base}}$ through top-k masked fusion to obtain the final action prediction $\boldsymbol{\ell}^{final}$.

3 METHOD

Our method, ActionRefiner, refines action recognition by learning verb and noun representations through modelling their temporal dependencies under the action context. This design is motivated by two key observations. First, a model trained for predicting action labels has representations optimized for action recognition, which fails to capture representations of verb and noun individually, incorporating verb and noun information could help action recognition. Second, independently trained verb or noun models for verb or noun prediction lack awareness of the full action context. The temporal representation from a model trained for predicting action implicitly contains the information of effects between verb and noun and it should be accounted when learning the representations of verb and noun. We first train two separate visual encoders to classify verbs and nouns and extract verb-specific and noun-specific per-timestep features. Then we train ActionRefiner module to identify the causal temporal generation processes for verbs and nouns. Using the latent causal variable $\mathbf{z}_t^v$ for verb and $\mathbf{z}_t^n$ for noun, a classifier predicts the logits for predicting action, which serves as a residual correction and it is added to baseline logits. The baseline remains unchanged, and ActionRefiner operates purely as a plug-in module. Our method is agnostic to the choice of baseline as long as it has intermediate temporal features $\mathbf{A}$.

3.1 VISUAL ENCODERS AND ACTION BASELINE

We assume a baseline model for action recognition has modules or layers e.g. transformers where dense temporal features are accessible. We freeze the baseline model and extract intermediate feature $\mathbf{A} \in \mathbb{R}^{T' \times d_a}$ with T' the temporal dimension of action and d_a the feature dimension. We then train two identical visual encoders to classify verbs and nouns and extract $\mathbf{V} \in \mathbb{R}^{T \times d_v}$ and $\mathbf{N} \in \mathbb{R}^{T \times d_n}$ with T the temporal dimension for verb and noun feature, d_v the verb feature dimension and d_n the noun feature dimension.

3.2 CAUSAL TEMPORAL GENERATION PROCESS

Following Yao et al. (2022); Li et al. (2024a), we describe the general data generation process with non-stationary non-parametric causal relation and time lag equals to 1,

$$\mathbf{x}_t = \mathbf{g}(\mathbf{z}_t), z_{it} = f_i\left(\{z_{j,t-1} | z_{j,t-1} \in \mathbf{Pa}(z_{it})\}, z_{it}^u, \epsilon_{it}\right), \tag{1}$$

the observation $\mathbf{x}_t$ is generated from the latent variable $\mathbf{z}_t$ by the nonlinear mixing function $\mathbf{g}$. For each dimension i in the latent variable $\mathbf{z}_t$, z_{it} is generated from time-delayed parents of z_{it} ($\mathbf{Pa}(z_{it})$), domain information z^u_{it}, and noise ϵ_{it} by the transition function $f_i(\cdot)$. In our ActionRefiner, $\mathbf{x}_t$ for verb and noun is $\mathbf{v}_t$ and $\mathbf{n}_t$ respectively. The domain information $\mathbf{z}^u_t$ is learned from $\mathbf{a}_t$. We formulate the probabilistic presentations for verb and noun as follows,

$$p(\mathbf{v}_{1:T}, \mathbf{z}^v_{1:T}, \mathbf{z}^u_{1:T}) = p_\theta(\mathbf{v}_1|\mathbf{z}^v_1)p_\phi(\mathbf{z}^v_1|\mathbf{z}^u_1)p(\mathbf{z}^u_1)\prod_{t=2}^{T} p_\theta(\mathbf{v}_t|\mathbf{z}^v_t)p_\phi(\mathbf{z}^v_t|\mathbf{z}^v_{t-1}, \mathbf{z}^u_t)p(\mathbf{z}^u_t|\mathbf{z}^u_{t-1}), \quad (2)$$

$$p(\mathbf{n}_{1:T}, \mathbf{z}^n_{1:T}, \mathbf{z}^u_{1:T}) = p_\theta(\mathbf{n}_1|\mathbf{z}^n_1)p_\phi(\mathbf{z}^n_1|\mathbf{z}^u_1)p(\mathbf{z}^u_1)\prod_{t=2}^{T} p_\theta(\mathbf{n}_t|\mathbf{z}^n_t)p_\phi(\mathbf{z}^n_t|\mathbf{z}^n_{t-1}, \mathbf{z}^u_t)p(\mathbf{z}^u_t|\mathbf{z}^u_{t-1}), \quad (3)$$

each process is a non-stationary non-parametric transition Yao et al. (2022); Li et al. (2024a). $p_\theta(\mathbf{v}_1|\mathbf{z}^v_1)$ and $p_\theta(\mathbf{n}_1|\mathbf{z}^n_1)$ are the decoders for verb and noun. $p_\phi(\mathbf{z}^v_t|\mathbf{z}^v_{t-1}, \mathbf{z}^u_t)$ and $p_\phi(\mathbf{z}^n_t|\mathbf{z}^n_{t-1}, \mathbf{z}^u_t)$ are the temporal dynamics transitions for verb and noun.

The ActionRefiner consists of four components: per-timestep VAE encoders for verb and noun streams, a domain estimator that extracts a conditioning signal from the action features, the temporal transition governs temporal dynamics in latent space, and a residual classifier generates logits and merge with baseline logits.

VAE. We use $\mathbf{V}$ and $\mathbf{N}$ as the observation for verb and noun. The encoder for verb and noun at timestep t are $q^\omega_v(\hat{\mathbf{z}}^v_t|\mathbf{v}_t)$ and $q^\omega_n(\hat{\mathbf{z}}^n_t|\mathbf{n}_t)$ with $\hat{\mathbf{z}}^v_t \sim \mathcal{N}(\mu_v, \sigma_v), \hat{\mathbf{z}}^n_t \sim \mathcal{N}(\mu_n, \sigma_n)$. The decoders $p_\theta(\hat{\mathbf{v}}_t|\hat{\mathbf{z}}^v_t)$ and $p_\theta(\hat{\mathbf{n}}_t|\hat{\mathbf{z}}^n_t)$ generate the estimation of the verb observation $\hat{\mathbf{v}}_t$ and the noun observation $\hat{\mathbf{n}}_t$.

Domain Estimator. We make the temporal dimension of action features T' and verb and noun T the same by max pooling. Then we learn the domain information $\hat{\mathbf{z}}^u_t$ from action features $\mathbf{A}$ by a domain estimator,

$$\hat{\mathbf{z}}^u_t = \text{GRU}\big(\hat{\mathbf{z}}^u_{t-1}, f_s(\mathbf{a}_t)\big), \quad (4)$$

where $f_s(\cdot)$ is a sparse adapter Deecke et al. (2022) that applies a learnable gating mechanism (Martins & Astudillo (2016)) over a bank of parallel adapters, selecting a data-dependent sparse combination per sample. This domain information $\hat{\mathbf{z}}^u$ captures the holistic action context that informs how verb and noun latent states should evolve over time.

Temporal Transition. The temporal transitions for verb and noun are $p_\phi(\mathbf{z}^v_t|\mathbf{z}^v_{t-1}, \mathbf{z}^u_t)$ and $p_\phi(\mathbf{z}^n_t|\mathbf{z}^n_{t-1}, \mathbf{z}^u_t)$. Since they are intractable, the below estimations are used,

$$p_\phi(\mathbf{z}^v_t|\mathbf{z}^v_{t-1}, \mathbf{z}^u_t) = p_\phi\big(\hat{f}_i^{-1}(\hat{\mathbf{z}}^v_{it}|\hat{\mathbf{z}}^v_{t-1}, \hat{\mathbf{z}}^u_{it})\big)\left|\frac{\partial f_i^{-1}}{\partial z^v_{it}}\right|, \quad (5)$$

$$p_\phi(\mathbf{z}^n_t|\mathbf{z}^n_{t-1}, \mathbf{z}^u_t) = p_\phi\big(\hat{f}_i^{-1}(\hat{\mathbf{z}}^n_{it}|\hat{\mathbf{z}}^n_{t-1}, \hat{\mathbf{z}}^u_{it})\big)\left|\frac{\partial f_i^{-1}}{\partial z^n_{it}}\right|, \quad (6)$$

we model verb and noun dynamics with separate transition priors, each conditioned on a shared action context $\mathbf{z}^u_t$. This reflects the assumption that verb and noun latents each evolve from their own past while remaining coupled through a shared action-level context, rather than independently.

3.3 Residual Classification

Posterior Averaging. Since the verb and noun encoders are variational, each forward pass yields a sample $\mathbf{z}_t \sim q(\mathbf{z}_t \mid \mathbf{x})$ rather than a deterministic latent representation. A single sample introduces variance into the residual $\ell^{\text{res},(r)}$, which directly affects the fused prediction. We reduce this variance by Monte Carlo averaging over K posterior samples. For each posterior sample r, we draw $\mathbf{z}^{v,(r)}_{1:T}$ and $\mathbf{z}^{n,(r)}_{1:T}$ independently to form the concatenated latent $\mathbf{h}^{(r)} = [[\hat{\mathbf{z}}^{v,(r)}_1; \hat{\mathbf{z}}^{n,(r)}_1], \ldots, [\hat{\mathbf{z}}^{v,(r)}_T; \hat{\mathbf{z}}^{n,(r)}_T]] \in \mathbb{R}^{2Td_z}$, where $[\cdot;\cdot]$ denotes feature concatenation.

Action Classification Head. For the r-th posterior sample $\mathbf{h}^{(r)}$, a residual head $f_{\text{res}}(\cdot)$ maps this representation to sample-specific correction logits $\boldsymbol{\ell}^{\text{res},(r)} = f_{\text{res}}(\mathbf{h}^{(r)}) \in \mathbb{R}^{C_a}$. and we take the mean of K correction logits,

$$\bar{\boldsymbol{\ell}}^{\text{res}} = \frac{1}{K} \sum_{r=1}^{K} \boldsymbol{\ell}^{\text{res},(r)}, \tag{7}$$

setting $K = 1$ recovers single-sample inference; larger K yields a lower-variance posterior predictive estimate of the residual correction.

Top-k Masked Fusion. A naive fusion rule, $\boldsymbol{\ell}^{\text{base}} + \alpha\, \bar{\boldsymbol{\ell}}^{\text{res}}$, modifies logits across all C_a action classes. In practice, a strong backbone already places the correct class within its top candidates, making global correction both unnecessary and a source of noise on easy classes. We therefore restrict the correction to the backbone's shortlist, turning the ActionRefiner module into a reranker rather than a global corrector. Given backbone logits $\boldsymbol{\ell}^{\text{base}}$, we form a binary mask $\mathbf{m} \in \{0,1\}^{C_a}$ that selects the top-k candidates, $m_c = \left[c \in \text{TopK}(\boldsymbol{\ell}^{\text{base}}, k)\right]$, and mask the averaged residual elementwise before fusion,

$$\boldsymbol{\ell}^{\text{final}} = \boldsymbol{\ell}^{\text{base}} + \alpha\, (\mathbf{m} \odot \bar{\boldsymbol{\ell}}^{\text{res}}), \tag{8}$$

where α is a scalar fusion weight. Classes outside the shortlist retain their original backbone score. We treat the shortlist size k as a backbone-dependent hyperparameter, since the optimal candidate policy depends on the architecture and calibration of the action baseline. In particular, top-k masking is most effective when the baseline produces a reliable shortlist with high oracle recall. Otherwise, the shortlist may exclude useful candidate classes and limit the effect of the residual correction.

3.4 Training Objective

We train the correction module with a joint objective that combines action classification, feature reconstruction, and two KL regularizers,

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{cls}} + \lambda_{\text{swap}}\, \mathcal{L}_{\text{swap}}}_{\text{Residual Classification}} + \underbrace{\lambda_{\text{rec}}\, \mathcal{L}_{\text{rec}} + \lambda_{\text{prior}}\, \mathcal{L}_{\text{KL}}^{\text{prior}} + \lambda_{\text{trans}}\, \mathcal{L}_{\text{KL}}^{\text{trans}} + \lambda_{\text{vn}}\, \mathcal{L}_{\text{vn}}}_{\text{CTGP}}, \tag{9}$$

where, $\mathcal{L}_{\text{cls}}$ is the label-smoothed cross-entropy loss Szegedy et al. (2016) on the fused action logits, $\mathcal{L}_{\text{swap}}$ penalizes mismatched verb-noun combinations. Within each batch, we construct negative latent pairs by replacing one verb or noun while keeping the other fixed: noun-swapped examples $(\hat{\mathbf{z}}_i^v, \hat{\mathbf{z}}_j^n)$ with $y_j^n \neq y_i^n$, and verb-swapped examples $(\hat{\mathbf{z}}_j^v, \hat{\mathbf{z}}_i^n)$ with $y_j^v \neq y_i^v$. Each negative latent pair is passed through the action classification head. Since the swapped pair should no longer support the original action label y_i^a, we penalize the logits where the original action remains higher than the best alternative by a margin,

$$\mathcal{L}_{\text{swap}} = \mathbb{E}_{(i,j)\in\mathcal{S}} \left[w_i \left[\boldsymbol{\ell}_{\text{swap}}^{\text{res}}[y_i^a] - \max_{c \neq y_i^a} \boldsymbol{\ell}_{\text{swap}}^{\text{res}}[c] + \rho \right]_+ \right], \tag{10}$$

where $\mathcal{S}$ is the set of valid swapped pairs in the batch, $\mathbb{E}_{(i,j)\in\mathcal{S}}[\cdot]$ denotes averaging over these pairs, and ρ is the swap margin. The scalar w_i is a difficulty weight defined as $w_i = (1 - p_{i,y_i^a}^{\text{base}})^\eta$, where $p_{i,y_i^a}^{\text{base}} = \text{softmax}(\boldsymbol{\ell}_i^{\text{base}})_{y_i^a}$ is the frozen action backbone's probability for the ground-truth action class.

$\mathcal{L}_{\text{rec}}$ is an MSE reconstruction loss on the verb and noun features, $\mathcal{L}_{\text{KL}}^{\text{prior}}$ regularizes the first L latent steps toward a standard Gaussian prior, and $\mathcal{L}_{\text{KL}}^{\text{trans}}$ regularizes later latent steps against the action-conditioned transition prior described above. Both KL terms are summed over the verb and noun streams. We additionally apply verb and noun supervision on temporally averaged latent means,

$$\mathcal{L}_{\text{vn}} = \text{CE}(W_v \bar{\boldsymbol{\mu}}^v, y^v) + \text{CE}(W_n \bar{\boldsymbol{\mu}}^n, y^n), \tag{11}$$

where $\bar{\boldsymbol{\mu}}^v = \frac{1}{T}\sum_t \boldsymbol{\mu}_t^v$ and $\bar{\boldsymbol{\mu}}^n = \frac{1}{T}\sum_t \boldsymbol{\mu}_t^n$. This prevents the latent space from collapsing under reconstruction and KL pressure and keeps the verb/noun latents discriminative.

Table 1: Action recognition on EPIC-Kitchens-100 in Top-1 accuracy. Our ActionRefiner is applied as a plug-in correction module on top of existing baselines. Δ Top-1 reports gain over the corresponding baseline.

Method	Top-1	Δ Top-1
MoViNet Kondratyuk et al. (2021)	47.7	-
MeMViT Wu et al. (2022)	48.4	-
MTCN Kazakos et al. (2021)	49.6	-
Omnivore Girdhar et al. (2022)	49.9	-
MTV Yan et al. (2022)	50.5	-
LaViLa Zhao et al. (2023)	51.0	–
LaViLa + ActionRefiner (Ours)	**53.5**	**+2.5**
AVION Zhao & Krähenbühl (2023)	54.4	–
AVION + ActionRefiner (Ours)	**56.6**	**+2.2**
TIM Chalk et al. (2024)	56.4	–
TIM + ActionRefiner (Ours)	**57.7**	**+1.3**

4 Experiment

4.1 Datasets

We evaluate our ActionRefiner on two popular egocentric action recognition datasets: Epic-Kitchens-100 Damen et al. (2022) and EGTEA Gaze+ Li et al. (2018). Epic-Kitchens-100 is a large and challenging dataset that contains 700 untrimmed egocentric videos recorded in kitchens. It consists of 89,977 clips of actions in total. Following previous works Girdhar et al. (2022); Chalk et al. (2024), we predict the 3806 classes present in the train and validation set to avoid predicting invalid actions, corresponding to 97 verb classes and 300 noun classes. EGTEA Gaze+ has 28 hours of recordings of cooking activities. It includes 10325 trimmed video clips of fine-grained actions across 106 action classes. We report experimental results on train/test split 1.

4.2 Implementation Details

We finetune three different types of visual encoders, i.e. V-JEPA 2.1 Gigantic (2B) Mur-Labadia & et al. (2026), V-JEPA 2.1 Large (300M) Mur-Labadia & et al. (2026) and MViT-S (34.5M) Li et al. (2022b), to predict noun and verb on both Epic-Kitchens-100 and EGTEA Gaze+ datasets. We select V-JEPA 2.1 Gigantic Mur-Labadia & et al. (2026) as the visual encoder for Epic-Kitchens-100 and MViT-S Li et al. (2022b) for EGTEA Gaze+ as they achieve the best accuracies on verb and noun prediction on the respective datasets. This yields better representations for verb and noun. The results are in Sec C.1 in Appendix.

We train ActionRefiner on Epic-Kitchens-100 for 400 epochs with AdamW, learning rate $3\text{e}{-4}$, batch size 32, and fusion weight $\alpha = 1$. We use posterior averaging with $K = 5$, top-k masked fusion uses $k = 25$ for TIM and $k = C_a$ for AVION and LaViLa. We set $\lambda_{\text{vn}} = 0.5$, $\lambda_{\text{swap}} = 0.2$, margin $\rho = 0.2$, and difficulty weighting exponent $\eta = 1.5$. The regularization weights are initialized as $\lambda_{\text{rec}} = 5\text{e}{-4}$, $\lambda_{\text{prior}} = 2.5\text{e}{-3}$, and $\lambda_{\text{trans}} = 0.25$, and are linearly decayed from epoch 10 to epoch 200 to final factors of 0.05, 0.1, and 0.1, respectively. For EGTEA Gaze+, we train with 150 epochs with Adam, learning rate $1\text{e}{-4}$, batch size 16, and fusion weight $\alpha = 1.5$. The rest is the same as Epic-Kitchens-100. More details are in Sec B in Appendix.

5 Results

5.1 Comparison with SOTA

We compare ActionRefiner with previous SOTA on Epic-Kitchens-100 and EGTEA Gaze+ datasets. Table 1 shows Top-1 accuracy on EPIC-Kitchens-100. We evaluate ActionRefiner as a plug-in

Table 2: Action recognition on EGTEA Gaze+ in Top-1 accuracy. Our ActionRefiner is applied as a plug-in correction module on top of existing baseline. † indicates the baseline results reproduced by ourself. Δ Top-1 reports gain over the baseline.

Method	Top-1	Δ Top-1
LSTA Sudhakaran et al. (2019)	61.9	-
MCN Huang et al. (2020)	62.6	-
SAP Wang et al. (2020)	64.1	-
Li *et al.* Li et al. (2022a)	65.7	-
Min *et al.* Min & Corso (2021)	69.6	-
SlowFast Feichtenhofer et al. (2019)	70.4	-
SDP Dai et al. (2023)	82.1	-
† MTCN Kazakos et al. (2021)	71.9	–
MTCN + ActionRefiner (Ours)	**73.9**	**+2.0**
LaViLa Zhao et al. (2023)	81.8	–
LaViLa + ActionRefiner (Ours)	**82.7**	**+0.9**

Table 3: Ablation of CTGP with and without $\mathcal{L}_{\text{vn}}$ on Epic-Kitchens-100 with TIM as the base model. † denotes our reproduction of baseline methods.

Method	Top-1
TIM Baseline	56.4
† TIM Baseline	56.6
TIM + CTGP w/o $\mathcal{L}_{\text{vn}}$	56.5
TIM + CTGP	57.4

method applied to existing baselines i.e. TIM, AVION and LaViLa, we report both absolute performance and the improvement (Δ Top-1) over each corresponding baseline method. We observe consistent gains when adding ActionRefiner to recent strong baselines. ActionRefiner improves Top-1 accuracy from 51 to 53.5 on LaViLa and from 54.4 to 56.6 on AVION. Applying ActionRefiner to TIM further boosts performance from 56.4 to 57.7, achieving the new SOTA. These results demonstrate the effectiveness of the plug-in property of ActionRefiner. The consistent improvements across diverse architectures shows that ActionRefiner provides a generalizable correction mechanism for action recognition performance. Table 2 presents Top-1 accuracy on EGTEA Gaze+. We evaluate ActionRefiner with MTCN and LaViLa. On MTCN, ActionRefiner improves MTCN by 2.0. When applying ActionRefiner to LaViLa, performance improves from 81.8 to 82.7, yielding a gain of +0.9. Despite the strong performance of LaViLa, ActionRefiner can still further improve the performance and establish new SOTA on EGTEA Gaze+.

5.2 Ablation Studies

Effect of CTGP We first ablate the ActionRefiner to see the effect of CTGP. We remove the components in residual classification and keep CTGP. We further compare CTGP with and without $\mathcal{L}_{\text{vn}}$ to show the importance of $\mathcal{L}_{\text{vn}}$. Table 3 shows the comparison on Epic-Kitchens-100 with TIM baseline. Adding $\mathcal{L}_{\text{vn}}$ improves Top-1 accuracy over the baseline, this shows that the VN loss $\mathcal{L}_{\text{vn}}$ is essential for CTGP.

Residual Classification Components. We ablate the components in residual classification in ActionRefiner on Epic-Kitchens-100 using TIM as baseline and keep CTGP. Table 4 starts from the reproduced baselines and progressively adds posterior averaging, top-k masked fusion and negative verb–noun swaps. This ablation isolates how each component contributes to the final correction, while keeping the action baseline and the V-JEPA verb/noun features fixed. Posterior averaging has Top-1 accuracy 57.4. Its effect is not yet visible at the action level. Top-k masked fusion reaches 57.6, restricting the correction to the top-k candidates confines it to instances on which the base

Table 4: Ablation of ActionRefiner components on Epic-Kitchens-100 with TIM as the baseline action recognition model. All ActionRefiner ablations follow the setting in Sec 4.2. Negative swaps include difficulty weighting. † denotes our reproduction of baseline methods.

Method	Posterior avg.	Top-k fusion	Swaps	Top-1
TIM Baseline	–	–	–	56.4
† TIM Baseline	–	–	–	56.6
TIM + ActionRefiner	✓	–	–	57.4
TIM + ActionRefiner	✓	✓	–	57.6
TIM + ActionRefiner	✓	✓	✓	**57.7**

Table 5: Top-1 accuracy (%) on the inverse-verb subset and the rest subset of EGTEA Gaze+. Gains over the corresponding base model are shown in parentheses. ActionRefiner improves more on inverse verb cases than on the rest for both base models.

Method	Inverse Verb	Rest	Full Test Set
LaViLa	80	85.3	81.8
LaViLa + ActionRefiner	**81.1 (+1.1)**	**86.1 (+0.8)**	**82.7 (+0.9)**
MTCN	68.5	79	72
MTCN + ActionRefiner	**70.8 (+2.3)**	**80.5 (+1.5)**	**73.9 (+1.9)**

classifier is uncertain and leaves confident predictions not affected. Adding negative swaps gives 57.7 Top-1 accuracy, which is 1.1 higher than the reproduced TIM baseline.

5.3 Analyses On Inverse Verb

Improvements on inverse verb. To study how ActionRefiner improves base models, we perform analyses on EGTEA Gaze+ dataset. Concretely, we construct two subsets, i.e. inverse verb and rest subsets. The inverse verb subset only contains the actions that have the same noun but have inverse verbs, e.g. "take tomato" and "put tomato". The rest subset contains the remaining actions. Top-1 accuracies of ActionRefiner on LaViLa and MTCN are shown in Table 5. On the inverse verb subset, ActionRefiner improves LaViLa and MTCN by 1.1 and 2.3 while 0.8 and 1.5 on the rest subset, this indicates that the performance gain of ActionRefiner is more attributed to inverse verb cases.

Action Context. To evaluate the effect of the action context in CTGP, we remove the domain estimator in CTGP to verify if action context helps with inverse verb. By removing the domain estimator, the generation process of verb and noun are not conditioned on action context anymore. Table 6 shows the comparison of ActionRefiner with and without action context i.e. the domain estimator. Without domain estimator, ActionRefiner has 80.8 top-1 accuracy on the inverse verb subset, 0.3 lower than ActionRefiner with domain estimator. On the rest subset, ActionRefiner without domain estimator is on par with ActionRefiner, but is lower by 0.2 on the full test set. Compared to baseline LaViLa (Table 5), ActionRefiner without domain estimator improves both inverse verb subset and rest subset by 0.8. This shows a more general improvement over the baseline. With domain estimator, however, CTGP can further improve recognizing action with inverse verb. Furthermore, the domain estimator shows a favorable accuracy/cost ratio with only contains 2.36M parameters.

Alternative Verb and Noun Modelling. We compare CTGP with Mamba transformer to verify the necessity of modelling verb and noun under causal framework. Specifically, we use two Mamba transformers for verb and noun, completely replacing CTGP but keep other components in ActionRefiner. Table 6 shows the comparison. Compared to CTGP, Mamba is 0.8, 2.5 and 2 lower on the inverse verb subset, rest subset and the full test set. This indicates CTGP better models verb and noun. Moreover, Mamba is also lower than LaViLa on full test set, indicating a generic temporal model over verb and noun features does not refine the base model's predictions.

Table 6: Comparisons of ActionRefiner, ActionRefiner without domain estimator and Mamba on EGTEA Gaze+ with LaViLa as the base model. We report top-1 accuracy, the number in parentheses indicates the difference.

	Inverse Verb	Rest	Full Test Set
ActionRefiner	**81.1**	**86.1**	**82.7**
ActionRefiner (w/o DS)	80.8 (-0.3)	86.1 (+0.0)	82.5 (-0.2)
LaViLa + Mamba	80.3 (-0.8)	84.6 (-2.5)	81.7 (-2)

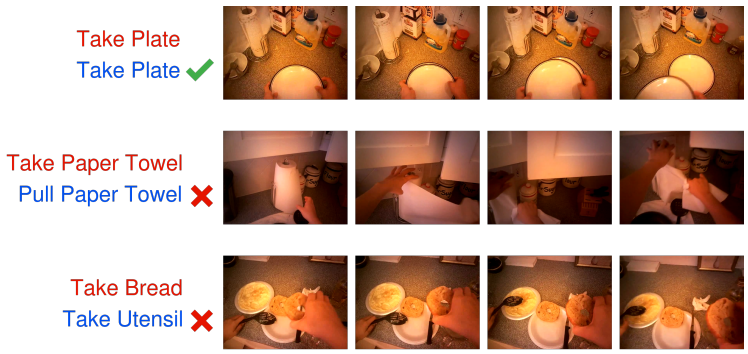


Figure 2: Qualitative Analysis on EGTEA Gaze+ dataset. Text in red is the ground truth action, Blue text is prediction from our method.

5.4 Qualitative Analysis

Figure 2 shows the qualitative results on EGTEA Gaze+ dataset. The ground truth is in red, and ActionRefiner's prediction is in blue. In the first row, our method correctly predicts the action. The second row shows a case when ActionRefiner predicts the wrong action. However, the annotation is noisy, i.e. the action in the video is "Pull Paper Towel" instead of "Take Paper Towel". Although our prediction is not the same as the label, it still predicts the actual action in the video. In the third row, ActionRefiner predicts "Take Utensil" instead of "Take Bread", but the person has the utensil in the left hand. This is a case where multiple actions exist in a video but only one is annotated. ActionRefiner can still predict part of the actions in a video even though some label is missing.

6 Conclusion

We presented ActionRefiner, a plug-in framework that refines prediction in egocentric action recognition by modelling verb and noun temporal dynamics through a causal temporal generation process. By leveraging temporal representations of verb and noun conditioned on action context, our method captures dependencies between verbs and nouns and improves recognition of challenging inverse-verb actions. Experiments on Epic-Kitchens-100 and EGTEA Gaze+ show consistent improvements over strong baselines, achieving state-of-the-art performance while remaining model-agnostic. Qualitative results further demonstrate that our method effectively improves predictions on inverse-verb actions. These results show the effectiveness of our approach in improving action recognition accuracy.

7 Disclose AI Use

In this work, we used generative AI tools for edit a research paper to improve readability. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, support qualitative and thematic data analysis, interpret results, and the rest of the required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Kartik Ahuja, Divyat Mahajan, Yixin Wang, and Yoshua Bengio. Interventional causal representation learning. In *International conference on machine learning*, pp. 372–407. PMLR, 2023.

Ruichu Cai, Junxian Huang, Zhenhui Yang, Zijian Li, Emadeldeen Eldele, Min Wu, and Fuchun Sun. Time series domain adaptation via latent invariant causal mechanism. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Jacob Chalk, Jaesung Huh, Evangelos Kazakos, Andrew Zisserman, and Dima Damen. Tim: A time interval machine for audio-visual action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18153–18163, 2024.

Guangzhao Dai, Xiangbo Shu, Rui Yan, Peng Huang, and Jinhui Tang. Slowfast diversity-aware prototype learning for egocentric action recognition. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 7549–7558, 2023.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 130:33–55, 2022.

Lucas Deecke, Timothy Hospedales, and Hakan Bilen. Visual representation learning over latent domains. In *International Conference on Learning Representations*, 2022.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6202–6211, 2019.

Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens Van Der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16102–16112, 2022.

Dan Guo, Kun Li, Bin Hu, Yan Zhang, and Meng Wang. Benchmarking micro-action recognition: Dataset, methods, and applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(7):6238–6252, 2024. doi: 10.1109/TCSVT.2024.3358415.

Yifei Huang, Minjie Cai, Zhenqiang Li, Feng Lu, and Yoichi Sato. Mutual context network for jointly estimating egocentric gaze and action. *IEEE Transactions on Image Processing*, 29:7795–7806, 2020.

Jinhyeok Jang, Dohyung Kim, Cheonshu Park, Minsu Jang, Jaeyeon Lee, and Jaehong Kim. Etri-activity3d: A large-scale rgb-d dataset for robots to recognize daily activities of the elderly. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10990–10997. IEEE, 2020.

Evangelos Kazakos, Jaesung Huh, Arsha Nagrani, Andrew Zisserman, and Dima Damen. With a little help from my temporal context: Multimodal egocentric action recognition. *arXiv preprint arXiv:2111.01024*, 2021.

Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16020–16030, 2021.

Haoxin Li, Wei-Shi Zheng, Jianguo Zhang, Haifeng Hu, Jiwen Lu, and Jian-Huang Lai. Egocentric action recognition by automatic relation modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):489–507, 2022a.

Shufei Li, Pai Zheng, Junming Fan, and Lihui Wang. Toward proactive human–robot collaborative assembly: A multimodal transfer-learning-enabled action prediction approach. *IEEE Transactions on Industrial Electronics*, 69(8):8579–8588, 2021.

Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4804–4814, 2022b.

Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 619–635, 2018.

Yuke Li, Guangyi Chen, Ben Abramowitz, Stefano Anzellott, and Donglai Wei. Learning causal domain-invariant temporal dynamics for few-shot action recognition. *arXiv preprint arXiv:2402.12706*, 2024a.

Zijian Li, Yifan Shen, Kaitao Zheng, Ruichu Cai, Xiangchen Song, Mingming Gong, Guangyi Chen, and Kun Zhang. On the identification of temporally causal representation with instantaneous dependence. *arXiv preprint arXiv:2405.15325*, 2024b.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017.

Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M Asano, Taco Cohen, and Efstratios Gavves. Causal representation learning for instantaneous and temporal effects in interactive systems. *arXiv preprint arXiv:2206.06169*, 2022.

André F. T. Martins and Ramón F. Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1614–1623, 2016.

Kyle Min and Jason J Corso. Integrating human gaze into attention for egocentric activity recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1069–1078, 2021.

Lorenzo Mur-Labadia and et al. V-jepa 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

Haozhe Qi, Shaokai Ye, Alexander Mathis, and Mackenzie W Mathis. Llavaction: evaluating and training multi-modal large language models for action understanding. *arXiv preprint arXiv:2503.18712*, 2025.

Sarah Rastegar, Hazel Doughty, and Cees GM Snoek. Background no more: Action recognition across domains by causal interventions. *Computer Vision and Image Understanding*, 242:103975, 2024.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Lei Shi, Paul-Christian Bürkner, and Andreas Bulling. ActionDiffusion: An Action-aware Diffusion Model for Procedure Planning in Instructional Videos. In *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

Xiangchen Song, Zijian Li, Guangyi Chen, Yujia Zheng, Yewen Fan, Xinshuai Dong, and Kun Zhang. Causal temporal representation learning with nonstationary sparse transition. *Advances in Neural Information Processing Systems*, 37:77098–77131, 2024.

Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9954–9963, 2019.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Junyan Wang, Yiqi Jiang, Yang Long, Xiuyu Sun, Maurice Pagnucco, and Yang Song. Deconfounding causal inference for zero-shot action recognition. *IEEE Transactions on Multimedia*, 26:3976–3986, 2023.

Xiaohan Wang, Yu Wu, Linchao Zhu, and Yi Yang. Symbiotic attention with privileged information for egocentric action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 12249–12256, 2020.

Xiaohan Wang, Linchao Zhu, Heng Wang, and Yi Yang. Interactive prototype learning for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8168–8177, 2021.

Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pp. 13587–13597, 2022.

Shaowu Xu, Xibin Jia, Junyu Gao, Qianmei Sun, Jing Chang, and Chao Fan. Cross-modal dual-causal learning for long-term action recognition. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 2919–2928, 2025.

Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3333–3343, 2022.

Weiran Yao, Guangyi Chen, and Kun Zhang. Temporally disentangled representation learning. *Advances in Neural Information Processing Systems*, 35:26492–26503, 2022.

Youliang Zhang, Wenxuan Liu, Danni Xu, Zhuo Zhou, and Zheng Wang. Bi-causal: Group activity recognition via bidirectional causality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1450–1459, 2024.

Yue Zhao and Philipp Krähenbühl. Training a large video model on a single machine in a day. *arXiv preprint arXiv:2309.16669*, 2023.

Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6586–6597, 2023.

APPENDIX

A COMPUTE RESOURCES

All experiments were run on internal GPU clusters. For Epic-Kitchens-100, each experimental run used one Slurm node equipped with an AMD EPYC 9555 CPU (64 cores, 128 threads), one NVIDIA H200 NVL GPU, and 1 TB ECC RAM. All EGTEA Gaze+ experiments were run on one A100 GPU with 40GB VRAM, 32 core Intel(R) Xeon(R) Gold 6338 CPU and 64GB RAM.

The Epic-Kitchens-100 reported experiments include V-JEPA probe finetuning for verb and noun features, feature and logit pre-computation for frozen action backbones, ActionRefiner training and evaluation on top of TIM, AVION, and LaViLa, and the ablation studies reported in section 5. Each individual run was executed as a single Slurm job on one H200 NVL node. On EGTEA Gaze+, we reported experiments include finetuning MViTv2 for verb and noun prediction, feature extraction from MViTv2 and LaViLa, ActionRefiner training and evaluation with LaViLa. All experiments were performed on one A100 GPU. Since ActionRefiner is trained on pre-extracted features with frozen action backbones, the reported experiments do not require training the action-recognition backbones from scratch.

For Epic-Kitchens-100, feature extraction for each verb or noun stream took approximately 27 hours. V-JEPA probe finetuning took approximately 40 hours for a 16-epoch run. Training and evaluating one ActionRefiner configuration on pre-extracted features took approximately 13 hours. These wall-clock times exclude Slurm queueing time. For EGTEA Gaze+, finetuning MViTv2 took around 5 hours. Feature extraction for MViTv2 and LaViLa took around 18 hours. Training and evaluating ActionRefiner with one configuration took around 2 hours.

Across the reported Epic-Kitchens-100 runs, the total compute was approximately 500 H200 GPU-hours and 400 hours on EGTEA Gaze+. The full research project required additional compute beyond the reported runs for preliminary experiments, failed runs, and hyperparameter tuning. These runs used the same resources, but we did not keep a complete accounting of their total GPU-hours.

B MORE IMPLEMENTATION DETAILS

Action feature extraction. For each backbone, we pre-compute the baseline action logits $\boldsymbol{\ell}^{\text{base}}$ and temporal action features $\mathbf{A}$. For TIM, which operates on long context windows, we construct action-query-aligned features: rather than assigning the full window representation to every action in the window, we select the temporal slice corresponding to the queried action interval and resample it to $T = 8$ steps. For LaViLa and AVION, we use the corresponding frozen action-stream features and logits directly.

Visual Encoders We use V-JEPA 2.1 Gigantic Mur-Labadia & et al. (2026) as the frozen visual encoders and finetune probes to predict verb and noun on Epic-Kitchens-100. To finetune V-JEPA 2.1 Gigantic, we train 50 epochs with batch size 8 and the warm up with 5 epochs. We use AdamW as the optimizer and the learning rate (lr) is $3e^{-4}$. A cosine scheduler is used to schedule the lr with the start lr $1e^{-6}$ for warm up and final lr $1e^{-5}$. The loss function is focal loss Lin et al. (2017). We use MViTv2-S as the visual encoder for EGTEA Gaze+. We use SDG optimizer with lr $3e^{-4}$, batch size is 8 and epoch is 50.

Method training. The action backbone is frozen throughout all experiments, and we train only the ActionRefiner module on pre-extracted features. The V-JEPA 2.1 Gigantic verb and noun probes produce $d_v = d_n = 1664$ dimensional features, while all action streams provide $d_a = 1024$ dimensional temporal features and baseline logits. We use $T = 8$ temporal steps for the ActionRefiner module, aligning the action stream to this length and using the matching temporal probe features for the verb and noun streams. We use separate verb and noun VAEs with latent dimension $d_z = 16$ and a residual MLP classifier over the flattened concatenated verb–noun latent sequence. We train for 400 epochs with AdamW, learning rate 3×10^{-4}, batch size 32, and fusion weight $\alpha = 1$. We use posterior averaging with $K = 5$ samples at evaluation. The verb/noun loss weight is $\lambda_{\text{vn}} = 0.5$. For negative swaps, we use $\lambda_{\text{swap}} = 0.2$, margin $\rho = 0.2$, and difficulty weighting exponent $\eta = 1.5$.

Reranking and regularization. In our experiments, TIM Chalk et al. (2024) benefits from reranking with a compact shortlist and we use $k = 25$. For LaViLa Zhao et al. (2023) and AVION Zhao & Krähenbühl (2023), validation performance is better with unrestricted residual fusion, which corresponds to setting $k = C_a$. When the ground-truth class falls outside the backbone top-k *during training*, we inject it into the mask so that the reranker receives a learning signal on such samples. At inference, the mask is built purely from the backbone logits.

C More Results

C.1 Visual Encoders

Table 7: Results of verb and noun prediction of different visual encoders on Epic-Kitchens-100 and EGTEA Gaze+ datasets. We report Top-1 accuracy on both datasets.

Dataset	Visual Encoder	Verb	Noun
Epic-Kitchen-100	V-JEPA 2.1 Large	71.6	60.2
	V-JEPA 2.1 Gigantic	**72.9**	**62.3**
	MViTv2	56.0	46.1
EGTEA Gaze+	V-JEPA 2.1 Large	72.3	73.1
	V-JEPA 2.1 Gigantic	72.3	74.3
	MViTv2	**83.5**	**82.7**

We finetune different models as visual encoders to predict verb and noun from videos. We show the Top-1 accuracy on both Epic-Kitchens-100 and EGTEA Gaze+ datasets in Table 7. On Epic-Kitchens-100, V-JEPA 2.1 Gigantic achieves the highest accuracy on both verb and noun prediction and V-JEPA 2.1 Large has slightly lower performance. MViTv2, on the other hand, is not comparable with V-JEPA. On the EGTEA Gaze+ dataset, MViTv2 has the best accuracy on verb and noun. The accuracies of V-JEPA models are around 10 lower.

Table 8: Ablation study on different visual encoder backbone in ActionRefiner on Epic-Kitchens-100 dataset. Δ Top-1 shows the improvements to baseline methods.

Visual Encoder (No. Para)	Method	Top-1	Δ Top-1
V-JEPA 2.1 Gigantic (2B)	TIM + ActionRefiner	57.7	1.3
	Avion + ActionRefiner	56.6	2.2
	LaViLa + ActionRefiner	53.5	2.5
V-JEPA 2.1 Large (300M)	TIM + ActionRefiner	57.4	1.0
	Avion + ActionRefiner	56.2	1.8
	LaViLa + ActionRefiner	53.5	2.5

To study the effect of the visual encoders for noun and verb, we perform ablation studies on two different visual encoder backbones described in Sec 3.1. The original ActionRefiner uses V-JEPA 2.1 Gigantic as visual encoders to predict verb and noun, the parameters of visual encoder is 2B. We additionally use a smaller visual encoder, V-JEPA 2.1 Large, with 300M parameters for comparison. Table 8 shows the results on Epic-Kitchens-100 dataset. ActionRefiner consistently improves all baseline methods across both backbone with different scales. V-JEPA Large (300M) already outperforms the respective baselines. This shows the effectiveness of ActionRefiner, the performance increase is not due to a strong visual encoder backbone. On the other hand, using the larger V-JEPA Gigantic (2B) backbone has better performance on predicting verb and noun (see Table 7 in Sec C.1) and it achieves higher performance on action recognition. This suggests our ActionRefiner can further benefit from a visual encoder that can learn better verb and noun representations.

Table 9: Ablation of ActionRefiner CTGP on Epic-Kitchens-100 with LaViLa action baseline. † denotes our reproduction of baseline methods.

Method	Top-1
LaViLa Baseline	51.0
† LaViLa Baseline	50.7
LaViLa + CTGP w/o $\mathcal{L}_{\text{vn}}$	51.6
LaViLa + CTGP	**53.2**

Table 10: Ablation of ActionRefiner components on Epic-Kitchens-100 with LaViLa action baseline. All ActionRefiner ablations follow the setting in Sec 4.2. Negative swaps include difficulty weighting. † denotes our reproduction of baseline methods.

Method	Posterior avg.	Top-k fusion	Swaps	Top-1
LaViLa Baseline	–	–	–	51.0
† LaViLa Baseline	–	–	–	50.7
LaViLa + ActionRefiner	✓	–	–	53.2
LaViLa + ActionRefiner	✓	✓	–	**53.6**
LaViLa + ActionRefiner	–	–	✓	52.1
LaViLa + ActionRefiner	✓	–	✓	52.9
LaViLa + ActionRefiner	✓	✓	✓	53.5

C.2 Additional Ablation on Residual Classification Components

Table 9 shows the ablation of CTGP on Epic-Kitchens-100 with LaViLa baseline. CTGP without $\mathcal{L}_{\text{vn}}$ improves reproduced LaViLa baseline by 0.9 whereas CTGP improves by 2.5 on Top-1 accuracy. This is consistent with the results on TIM baseline, showing that CTGP effectively improves accuracy across different baseline models.

Table 10 shows the component analysis with LaViLa as the action baseline. The progression reproduces the behavior observed on TIM. Posterior averaging has Top-1 at 53.2, matching CTGP accuracy. Top-k masked fusion reaches 53.6, which is 2.9 higher than the reproduced baseline. Negative verb–noun swaps behave differently on LaViLa. Further adding swaps results in 53.5 Top-1 accuracy, which is slightly lower than without swaps.

C.3 Ablation on VAE

Table 11: Mean and standard deviation of Top-1 accuracy (%) of vanilla VAE and CTGP on Epic-Kitchens-100 dataset.

	Vanilla VAE	CTGP
Mean	56.56	57.66
Std	0.03	0.11

With vanilla VAE setting, we use two vanilla VAEs to replace entire CTGP and keep the rest of ActionRefiner unchanged. Table 11 shows the comparison of CTGP and vanilla VAE on Epic-Kitchens-100 dataset. CTGP consistently outperfoms vanilla VAE.

C.4 Ablation on Verb and Noun Branch

To further study the contribution of verb and noun modelling, we ablate verb only and noun only variants of CTGP by removing noun branch and verb branch respectively. The results are shown in Table 12. By removing either noun or verb branch, the performances on both inverse verb subset and rest subset. Verb only and noun only are 0.9 and 0.8 lower on the inverse verb subset and both -0.5 on rest subset. Intuitively one would expect verb only performs better than noun only on inverse

verb set. The reason it is not probably is that the downstream task is action recognition where the model predicts a single action label rather than separate verb and noun labels. Consequently, both the verb and noun transition branches are optimized to improve the representation in the action label space, whose categorical distribution and decision boundaries differ from those of the individual verb and noun label spaces. Therefore, removing one branch is expected to affect the overall action recognition.

Table 12: Comparisons of verb only and noun only CTGP on EGTEA Gaze+ with LaViLa as the base model. We report top-1 accuracy, the number in parentheses indicates the difference.

	Inverse Verb	Rest	Full Test Set
CTGP	**81.1**	**86.1**	**82.7**
verb only	80.2 (−0.9)	85.6 (−0.5)	81.9 (−0.8)
noun only	80.3 (−0.8)	85.6 (−0.5)	82 (−0.7)