2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Exploring Vision-Language Models for Open-Vocabulary Zero-Shot Action Segmentation

Asim Unmesh[1], Kaki Ramesh[2], Mayank Patel[1], Rahul Jain[1], and Karthik Ramani[1]

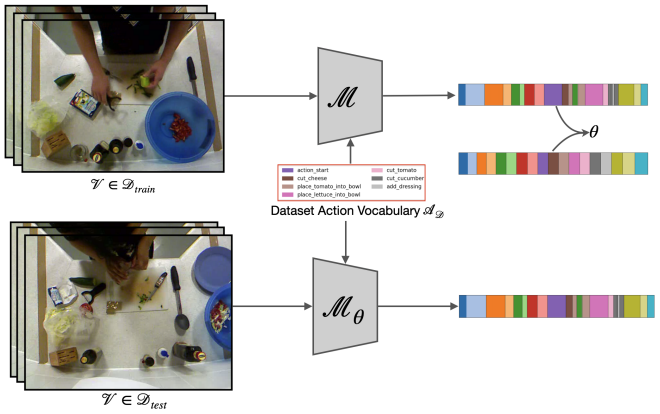
***Abstract*— Temporal Action Segmentation (TAS) requires dividing videos into action segments, yet the vast space of activities and alternative breakdowns makes collecting comprehensive datasets infeasible. Existing methods remain limited to closed vocabularies and fixed label sets. In this work, we explore the largely unexplored problem of Open-Vocabulary Zero-Shot Temporal Action Segmentation (OVTAS) by leveraging the strong zero-shot capabilities of Vision–Language Models (VLMs). We introduce a training-free pipeline that follows a segmentation-by-classification design: (i) *Frame–Action Embedding Similarity (FAES)* matches video frames to candidate action labels, and (ii) *Similarity-Matrix Temporal Segmentation (SMTS)* enforces temporal consistency. Beyond proposing OVTAS, we present a systematic study across 14 diverse VLMs, providing the first broad analysis of their suitability for open-vocabulary action segmentation. Experiments on standard benchmarks show that OVTAS achieves strong results without task-specific supervision, underscoring the potential of VLMs for structured temporal understanding. We release code and embeddings at our project page.**

I. Introduction

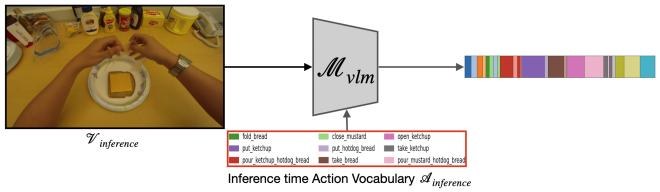
Temporal Action Segmentation (TAS) has been an active area of research [1] with applications in human activity understanding [2], surgical robotics [3], robot task learning [4], and action assessment [5]. The goal in action segmentation is to assign action labels to every frame of a video, segmenting it into meaningful units. Despite substantial progress, existing TAS methods remain constrained to a *closed vocabulary* of action labels. Models are typically trained and evaluated on the same fixed set of classes, limiting their ability to generalize to new actions or unseen domains.

This closed-vocabulary assumption is particularly restrictive because the space of possible action vocabularies is vast: a single activity may be decomposed into dozens of steps, and each domain—such as kitchen tasks, assembly, or surgery—contains hundreds of distinct activities. Even for the same task, alternative segmentations may exist, reflecting different emphases such as object-centered versus process-centered perspectives. Constructing annotated datasets that cover this variability is infeasible, and as a result, closed-vocabulary TAS methods struggle to scale beyond label spaces defined in the dataset.

Inspired by the application of Vision–Language Models (VLMs) [6], [7] for action understanding tasks, we propose the **Open-Vocabulary Temporal Action Segmentation (OVTAS) pipeline**, a training-free, zero-shot approach that leverages the open-vocabulary and zero-shot capabilities of VLMs. Contrastively trained VLMs such as CLIP and SigLIP demonstrate strong image-level zero-shot recognition of novel categories by aligning visual and textual embeddings. Our key idea is to use this capability for TAS in a "segmentation-by-classification" setting.



(a) Other approaches



(b) Proposed approach

Fig. 1: **Problem Setup:** Existing approaches in 1a are fixed vocabulary and do not generalize to unseen videos. 1b illustrates our proposed method for open-vocabulary and zero-shot action segmentation.

However, VLM predictions at the frame level are temporally inconsistent, as they operate independently across frames. To overcome this, OVTAS follows a two-stage segmentation-by-classification design. Stage 1, *Frame–Action Embedding Similarity (FAES)*, computes similarities between frame embeddings and text embeddings of action labels. Stage 2, *Similarity-Matrix Temporal Segmentation (SMTS)* [8], decodes these similarities into temporally consistent label sequences using an optimal transport-based approach.

Understanding how different Vision–Language Models behave in the task of open-vocabulary temporal action segmentation, informs future research and application about the choices of VLM models and their sizes. Given the various VLM models, varying across families (such as CLIP, SigLIP) and model sizes, a systematic exploration is needed to

[1]Purdue University, USA
[2]Birla Institute of Technology and Science (BITS) Hyderabad, India

inform the research community of performance levels across different choices. To address this, we systematically explore 14 diverse VLMs across families and sizes, uncovering performance trends.

Our contributions are as follows:

- **Pipeline Design**: We introduce the **OVTAS pipeline**, a two-stage framework—FAES followed by SMTS—that produces temporally consistent per-frame labels without any task-specific training or fine-tuning.
- **Comprehensive VLM Study**: We extensively evaluate state-of-the-art VLM families across three TAS benchmarks, uncovering performance trends and key factors influencing success in open-vocabulary segmentation.

To enable further research, we release our codebase and the extracted vision–language embeddings of 14 VLMs for all three datasets. Since feature extraction from large VLMs demands substantial computational resources, providing ready-to-use embeddings eases this barrier, enabling broader research into VLMs for Action Segmentation and other action understanding tasks.

II. Related Works

Temporal Action Segmentation (TAS) has been studied under different supervision regimes. **Weakly Supervised Methods.** These approaches aim to reduce the need for dense frame-level annotation by exploiting weaker forms of supervision that are cheaper to obtain. Examples include (a) timestamp supervision [12], [20], where only a few time instances are annotated; (b) action set supervision [15], [21], where only the unordered list of actions is provided; and (c) transcript supervision [14], [17], where the sequence of actions is known but not aligned to frames. Among these, timestamp supervision typically yields the best performance, followed by action sets, and then transcripts. **Semi-Supervised Methods.** Semi-supervised methods [10], [10], [22] use dense labels for only a subset of videos, showing that even a small number of fully labeled videos can outperform weak supervision on larger datasets. **Unsupervised Methods.** Unsupervised TAS methods [8], [23]–[25] operate without any action-level labels, relying only on video-level activity labels. **Our Approach.** In contrast, our OVTAS pipeline explores a new regime: *training-free, zero-shot, open-vocabulary TAS*. By leveraging VLMs, OVTAS segments videos without any task-specific training and generalizes to unseen action labels, addressing the closed-vocabulary constraint in prior work. A comparison of existing temporal action segmentation approaches is given in Table I.

III. Method

A. Overview

We propose **Open-Vocabulary Temporal Action Segmentation (OVTAS)**, a *training-free, zero-shot* pipeline that requires only a set of candidate action labels (the *action set supervision*) and video frames as inputs. Action set supervision assumes that, given knowledge of the high-level task or activity (e.g., "making tea"), we have access to the set

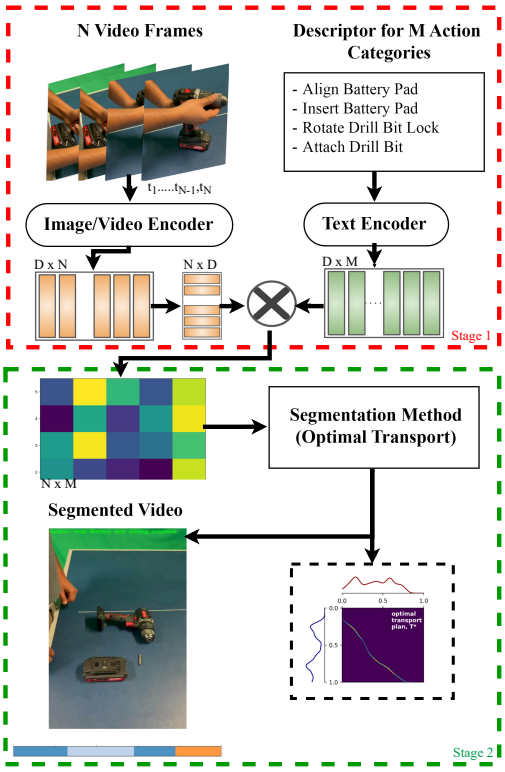


Fig. 2: Open-Vocabulary Temporal Action Segmentation (OVTAS) Pipeline. Our 2-stage pipeline adopts a "segmentation by classification" approach to tackle temporal action segmentation (TAS). Stage 1, Frame–Action Embedding Similarity (FAES), generates a similarity matrix by matching frames with action labels. Stage 2, Similarity-Matrix driven Temporal Segmentation (SMTS), uses optimal transport with a temporal prior to enforce temporal consistency, producing stable action segments.

of all possible fine-grained actions (e.g., "boil water," "pour tea," "add sugar") but not their order or boundaries. Our method proceeds in two stages: (i) *Frame–Action Embedding Similarity (FAES)* computes similarities between video frames and action label embeddings, and (ii) *Similarity-Matrix driven Temporal Segmentation (SMTS)* decodes these similarities into temporally consistent action segments using entropy-regularized optimal transport with a temporal prior.

B. Terminology

Notation. Bold uppercase = matrices, bold lowercase = vectors, italics = scalars. Let T be the number of frames, N the number of action labels, and C the embedding dimension.

A (Activity) A high-level category for an entire video (e.g., "making tea").

$\mathbf{X} \in \mathbb{R}^{T \times C}$ **(Frame Embeddings)** Row-wise ℓ_2-normalized frame embeddings from the VLM vision encoder; $\|\mathbf{x}_t\|_2 = 1$.

$\mathbf{A} \in \mathbb{R}^{N \times C}$ **(Action Embeddings)** Row-wise ℓ_2-normalized text embeddings for the N action labels from the VLM text encoder; $\|\mathbf{a}_n\|_2 = 1$.

TABLE I: Comparison of our method with other methods in TAS based on generalization ability and training data needs. ✓denotes applicability.

Method	Zero-Shot	Open-Vocab	Unsupervised	Weakly Sup.	Semi-Sup.	Training-Free
ASAL [9]	✗	✗	✓	✗	✗	✗
COIN-SSL [10]	✗	✗	✗	✗	✓	✗
NN-Viterbi [11]	✗	✗	✗	✓	✗	✗
ASAL-SSL [9]	✗	✗	✗	✗	✓	✗
TCCNet [12]	✗	✗	✗	✓	✗	✗
UDE [13]	✗	✗	✓	✗	✗	✗
CTC [14]	✗	✗	✗	✓	✗	✗
D3TW [15]	✗	✗	✗	✓	✗	✗
ASRF-unsup. [16]	✗	✗	✓	✗	✗	✗
C2F-TCN [8]	✗	✗	✓	✗	✗	✗
HTK [17]	✗	✗	✗	✓	✗	✗
BCN [18]	✗	✗	✗	✓	✗	✗
HVQ [19]	✗	✗	✓	✗	✗	✗
U-OT [8]	✗	✗	✓	✗	✗	✗
OVTAS (Ours)	✓	✓	✗	✓	✗	✓

$\mathbf{S} \in \mathbb{R}^{T\times N}$ **(Similarity Matrix)** Cosine similarity (dot product) between frame and action embeddings:

$$\mathbf{S} = \mathbf{X}\mathbf{A}^\top, \quad \ell_t = \mathbf{x}_t\,\mathbf{A}^\top \in \mathbb{R}^{1\times N}.$$

$\mathbf{P} \in \mathbb{R}^{T\times N}$ **(Frame Classification Probabilities)** Row-wise softmax over actions:

$$\mathbf{P} = \mathrm{softmax}_N(\mathbf{S}), \quad \mathbf{p}_t = \mathrm{softmax}(\ell_t).$$

C. Preliminaries: Optimal Transport Decoder

We adopt the Action Segmentation OT (ASOT) decoder of Xu and Gould [8]. Given frame–action similarities $\mathbf{S} \in \mathbb{R}^{T\times N}$, we define a visual cost $\mathbf{C} = \mathbf{1} - \mathbf{S}$ and a diagonal temporal prior R with

$$R_{ij} = \left|\tfrac{i}{T} - \tfrac{j}{N}\right|,$$

which encourages monotone alignment. We then solve for a coupling $\Pi \in \mathbb{R}^{T\times N}$:

$$\Pi^\star = \arg\min_{\Pi\in U(\mathbf{u},\mathbf{v})} \langle \Pi, \mathbf{C}+\rho R\rangle - \varepsilon H(\Pi), \quad \mathbf{u} = \tfrac{1}{T}\mathbf{1}_T,\ \mathbf{v} = \tfrac{1}{N}\mathbf{1}_N, \tag{1}$$

where $U(\mathbf{u},\mathbf{v})$ is the transport polytope and $H(\Pi)$ the entropy of Π. We use the same regularization and solver style as [8], and decode frame labels by

$$\hat{y}_t = \arg\max_j \Pi^\star_{t,j}.$$

Instantiations for OVTAS. (i) **Open-vocabulary costs: C** is derived from VLM similarities (FAES). (ii) **Action-set supervision:** only the set of actions is assumed known, and their order is randomized when constructing the temporal prior R. (iii) **Hyperparameters:** $(\varepsilon, \rho, \ldots)$ are chosen via grid search on a small set of selected videos per dataset; the selected values are fixed for all experiments. We note that training-free refers strictly to the absence of backpropagation based weight updates. Hyperparameter selection via grid search on a held-out set is doesn't require backpropagation based training, and hence is training free.

D. Stage 1: Frame–Action Embedding Similarity (FAES)

FAES computes raw similarity scores between frame and action embeddings:

1) **Prompt Construction.** Each action label is normalized into a natural-language phrase (e.g., "pour_coffee" → "pour coffee"). These normalized labels are tokenized and encoded by the VLM text encoder.
2) **Embed frames and actions.** Obtain $\mathbf{X} \in \mathbb{R}^{T\times C}$ and $\mathbf{A} \in \mathbb{R}^{N\times C}$ from the VLM encoders, applying row-wise ℓ_2-normalization.
3) **Compute similarities.**

$$\mathbf{S} = \mathbf{X}\mathbf{A}^\top \in \mathbb{R}^{T\times N}, \quad \ell_t = \mathbf{x}_t\mathbf{A}^\top.$$

E. Stage 2: Similarity Matrix–driven Temporal Segmentation (SMTS)

SMTS consumes the similarity matrix $\mathbf{S}$ and produces temporally consistent segments:

1) **Action Order (Action Set Supervision).** We assume action set supervision, we know only the set of actions present in the video, not their order. While constructing the similarity matrix, the ordering of the action embeddings is random. Action-set supervision is a reasonable as knowing activity typically implies knowing the set of possible actions.
2) **OT Alignment.** We apply the ASOT decoder (Eq. 1) with visual cost $\mathbf{C}$ and temporal prior R, yielding a coupling $\Pi^\star$. The entropy term ensures the problem is convex and yields a unique solution. In practice, $\Pi^\star$ is computed via Sinkhorn [26] iterations with log-stabilization, which scales linearly in $T\times N$ per iteration.
3) **Action Segmentation.** Assign each frame t the action with maximum OT mass:

$$\hat{y}_t = \arg\max_j \Pi^*_{t,j}.$$

TABLE II: Video duration statistics in seconds.

Dataset	Min (s)	Max (s)	Mean (s)
GTEA	42.27	133.93	74.35
Breakfast	12.27	649.53	137.40
50 Salads	251.60	605.00	385.25

TABLE III: Number of ground-truth segments per video.

Dataset	Min	Max	Mean
GTEA	24	49	36.3
Breakfast	1	18	5.4
50 Salads	15	27	20.7

IV. Experiments

A. Setup

1) Datasets: Experiments are performed on three standard Action Segmentation Benchmarks: Breakfast, 50 Salads, and the Georgia Tech Egocentric Activities (GTEA) dataset [27]–[29]. **Breakfast** [29] contains 1,712 videos (77 hours total) of 52 participants performing ten breakfast-related activities (e.g., making coffee, frying eggs). The recordings are captured from 3 to 5 third-person cameras, with annotations for 48 fine-grained actions. **50 Salads** [28] includes 50 top-view videos (5.5 hours total) of subjects preparing mixed salads, annotated with 17 action classes such as cutting, mixing, and adding ingredients. **GTEA** [27] consists of 28 egocentric videos (0.4 hours total) of four participants preparing seven different meals, annotated with 71 action classes. The statistics for video lengths of each dataset is mentioned in Table II, and number of action segments in the annotated ground truth is mentioned in Table III. We evaluate each dataset using its official evaluation splits and report the final results as the average across splits.

2) Metrics: We use 5 key metrics to report the result: F1@10, F1@25, F1@50, Accuracy, and Edit Scores. While Accuracy measures the overall proportion of correctly labeled frames, the F1 scores at different overlap thresholds capture the quality of segment-wise alignment with the ground truth, balancing precision and recall. Edit Score evaluates the temporal ordering and continuity of predicted segments, penalizing over-segmentation and fragmentation. Together, these metrics provide a comprehensive evaluation of action segmentation performance.

3) Baselines: We construct four training–free, zero–shot, open–vocabulary baselines. Let a video of T frames have frame embeddings $F = \{f_t\}_{t=1}^{T} \in \mathbb{R}^{T\times D}$ and a label set $\mathscr{C}$ of size C with text (or label) embeddings $A = \{a_c\}_{c\in\mathscr{C}} \in \mathbb{R}^{C\times D}$. We use cosine similarity

$$S_{t,c} = \left\langle \frac{f_t}{\|f_t\|_2}, \frac{a_c}{\|a_c\|_2} \right\rangle \in [-1,1],$$

and report per–frame predictions $y_{1:T}$.

a) (1) Random–Uniform (RU).: Each frame is labeled by a uniform categorical draw over classes:

$$y_t \sim \mathrm{Cat}(\pi), \qquad \pi_c = \tfrac{1}{C} \quad \forall c \in \mathscr{C}, \quad t = 1,\dots,T.$$

(If a background class exists, it is included in $\mathscr{C}$ as usual.)

b) Equal–Splits family.: We take inspiration from [30], and construct 3 equal splits baselines. For constructing an equal splits baseline, we partition the timeline into K contiguous bins with edges

$$e_k = \left\lfloor \tfrac{kT}{K} \right\rfloor, \quad k = 0,\dots,K, \qquad B_k = \{e_k+1,\dots,e_{k+1}\}.$$

A single label $\hat{y}_k$ is chosen per bin and then expanded to frames: $y_t = \hat{y}_k$ for all $t \in B_k$. We use three training–free labeling rules:

c) (2) Equal–Splits Mean (ES–Mean).: Score each class by its mean similarity in the bin and take the argmax:

$$s_{k,c}^{\mathrm{mean}} = \frac{1}{|B_k|} \sum_{t\in B_k} S_{t,c}, \qquad \hat{y}_k = \arg\max_{c\in\mathscr{C}} s_{k,c}^{\mathrm{mean}}.$$

d) (3) Equal–Splits Vote (ES–Vote).: First take per–frame winners $\tilde{y}_t = \arg\max_c S_{t,c}$, then choose the modal class in the bin:

$$\hat{y}_k = \mathrm{mode}\left\{\tilde{y}_t : \ t \in B_k\right\}.$$

(Ties, if any, are broken by larger $s_{k,c}^{\mathrm{mean}}$.)

e) (4) Equal–Splits Non–Repetition Penalty (ES–NRP).: Equal–splits can collapse to a trivial solution where every bin receives the same label. The non–repetition penalty adds a small cost for assigning the same label to adjacent bins, discouraging this degenerate constant prediction.

Let $s_{k,c}$ be any fixed per–bin class score (we use $s_{k,c}^{\mathrm{mean}}$ above). We decode bin labels by a simple dynamic program that discourages repeating the same class in adjacent bins:

$$\hat{y}_{1:K} = \arg\max_{y_1,\dots,y_K\in\mathscr{C}} \sum_{k=1}^{K} s_{k,y_k} - \lambda \sum_{k=2}^{K} \mathbb{1}[y_k = y_{k-1}],$$

with $\lambda \geq 0$ fixed globally. Finally, $y_t = \hat{y}_k$ for all $t \in B_k$.
All baselines share the same frozen encoder features and prompts, require no dataset–specific tuning, and are evaluated with the same metrics as our method. We find that λ is best for all the datasets.

TABLE IV: **Ablation on Temporal Prior and L2 Norm:** Parentheses show drops (↓)) from Best. Results on GTEA Dataset.

Metric	Best	Temporal Prior (abl.)	L2 Norm (abl.)
F1@10	21.33	1.12 (↓20.21)	2.36 (↓18.97)
F1@25	13.76	0.56 (↓13.20)	1.12 (↓12.64)
F1@50	5.44	0.19 (↓5.25)	0.62 (↓4.82)
Edit	51.07	5.04 (↓46.03)	7.23 (↓43.84)
Acc	28.08	6.54 (↓21.54)	8.43 (↓19.65)
Avg	23.14	2.69 (↓20.45)	3.95 (↓19.19)

4) Implementation Details: We implement our method and baselines using PyTorch. For the canonical action ordering which is required by the optimal transport algorithm, we do a random action ordering - thus only needing action set information. For extracting the action label embeddings, we perform simplistic transformations of the labels such

as "pour_coffee" in breakfast dataset to "pour coffee". For GTEA dataset we attach verbs to construct short phrases using the annotation. We select the hyperparameter values of optimal transport stage using the values given in [8]. Specifically, we set $\varepsilon = 0.07$, $\alpha = 0.5$, $r = 0.04$, $\lambda_{\text{frames}} = 0.11$, and $\lambda_{\text{actions}} = 0.01$. Different from [8] which uses optimal transport in an unbalanced setting, we find in our grid-search that balanced formulation gives the strongest results. We extract the per-frame Vision-Language Models (VLMs) features using an NVIDIA A6000 machine. We extract per-frame VLM features using off-the-shelf checkpoints without fine-tuning. Action embeddings are cached for efficiency. The chosen values are fixed for all experiments. We use 4 families of VLMs - Perception Encoders (PECore) [31] SigLIP [32], CLIP [33] and OpenCLIP [34]. The details of the VLMs are mentioned in Table V.

TABLE V: **VLM Details:** Parameter sizes (in millions) of the 14 VLM variants used in our experiments.

Abbrev	Checkpoint	#Params (M)
SigLIP-M1	so400m-p16-256-i18n	877.96
SigLIP-M2	large-p16-384	652.48
SigLIP-M3	so400m-p14-384	1128.76
OpenCLIP-M1	ViT-B/32 (laion2B-s34B-b79K)	151.28
OpenCLIP-M2	ViT-B/16 (laion2B-s34B-b88K)	149.62
OpenCLIP-M3	ViT-L/14 (laion2B-s32B-b82K)	427.62
OpenCLIP-M4	ViT-H/14 (laion2B-s32B-b79K)	986.11
OpenCLIP-M5	ViT-g/14 (laion2B-s34B-b88K)	1366.68
CLIP-M1	ViT-B/32	151.28
CLIP-M2	ViT-B/16	149.62
CLIP-M3	ViT-L/14	427.62
PECore-M1	B/16-224	447.66
PECore-M2	L/14-336	671.14
PECore-M3	G/14-448	2419.27

B. Performance Comparison

We report the baseline results and performance of our model in Table VI. Among the training-free baselines, ES_NRP consistently achieves the strongest results across datasets, outperforming ES_mean and ES_vote. This confirms that equal-splits baselines benefit from a non-repetition constraint, validating the use of NRP as the most representative baseline. GTEA consists of egocentric videos annotated with fine-grained action classes. The egocentric viewpoint introduces camera motion, making it the most challenging benchmark, which is reflected in the lowest performance. Across the five VLMs tested—SigLIP-M1, SigLIP-M2, OpenCLIP-M1, CLIP-M1, and PECore-M1—performance differences are relatively small on Breakfast and 50 Salads. In contrast, all models show larger drops on GTEA, reflecting the challenges posed by its egocentric viewpoint, rapid transitions, and large number of fine-grained classes. Our qualitative results are shown in Fig. 3. Our results show that, our OVTAS pipeline significantly outperforms baselines, and establishes encouraging results for the novel task of Open-Vocabulary Zero-Shot Action Segmentation.

C. Ablation Studies

To validate the components of our OVTAS pipeline, we perform ablation studies.

1) Stage Ablation: We ablate the Stage 1, by randomly permuting the frame features and action embedding features, and run the pipeline and measure the drop in performance for both the stages. For ablating stage 2, we perform frame level predictions using maximum probability label for for each frame's action classification probabilities. Our results show significant drops for all the metrics and for all the datasets - indicating the criticality of both the stages towards final performance.

2) L2-Norm and Temporal Prior Ablation: We also independently ablate L2-norm in stage1, and use of temporal priors in stage 2. We present our results on GTEA dataset. Table IV. Our results demonstrate the criticality of both those design choices for strong performance.

V. Discussions

1) VLM analysis: In this section we analyze the impact of VLM family choice and model size on OVTAS performance. We first compare different families of VLMs to understand which architectures and pre-training strategies are most effective for temporal action segmentation. We then study the effect of scaling model size within each family to examine whether larger checkpoints bring consistent improvements.

a) VLM Family Analysis: We compare the four VLM families (SigLIP, CLIP, OpenCLIP, and PECore) by averaging their performance across the three benchmark datasets. As shown in Fig. 5, consistent trends are observed across all datasets with the same ranking: the SigLIP family outperforms all others, followed by CLIP, while OpenCLIP and PECore trail behind. This consistency indicates that the relative strength of each family is stable across domains, with SigLIP providing the most reliable performance for OVTAS.

b) VLM Size Analysis: The plot in Fig. 4 shows that simply scaling VLMs up does not yield better action-segmentation performance using OVTAS—in all VLM families. Larger checkpoints underperform their smaller counterparts. Thus, as future research direction, we can take up (i) stronger text prompting and (ii) video frame pre-processing such as cropping. Together these may extract frame and action label embeddings possibly giving performance increase with increasing VLM size.

2) Action Confusion Analysis: We examine per action confusion patterns on GTEA to understand where OVTAS fails semantically. Table X reports the recognition accuracy and dominant confusion for each action type. `take` acts as a dominant attractor for most manipulation actions, as its reaching motion overlaps with the onset of nearly every other action. `shake` is almost entirely misclassified, reflecting the limitation of image-trained VLMs on dynamic motions.

3) Effect of Video Length and Action Segment Counts:

a) : We investigate how video duration influences performance by grouping videos into predefined length intervals. The results in Table VIII reveal a consistent pattern across all datasets: performance decreases as video length increases.

TABLE VI: **Comparison of baselines and all VLM variants across three datasets** (50 Salads, GTEA, and Breakfast). Avg is the mean of F1 scores, Edit, and Accuracy. Maximum average value for each dataset and each family is boldfaced.

Method	50 Salads				GTEA				Breakfast			
	F1@10,25,50	Edit	Acc	Avg	F1@10,25,50	Edit	Acc	Avg	F1@10,25,50	Edit	Acc	Avg
Random	0.00 / 0.00 / 0.00	0.20	5.46	1.13	0.12 / 0.01 / 0.01	2.06	5.61	1.56	0.00 / 0.00 / 0.00	0.72	17.45	3.63
ES_mean	1.49 / 0.13 / 0.04	5.40	5.63	2.54	7.08 / 3.99 / 1.67	13.00	5.99	6.35	20.56 / 11.97 / 3.63	22.39	17.14	15.14
ES_vote	2.77 / 0.84 / 0.31	6.72	5.68	3.26	7.32 / 4.63 / 1.87	12.68	5.96	6.49	22.15 / 13.69 / 4.52	24.66	17.15	16.43
ES_nrp	6.81 / 5.90 / 2.73	8.00	7.24	**6.14**	7.75 / 5.35 / 2.01	8.29	12.96	**7.27**	24.38 / 19.46 / 8.73	36.35	11.83	**20.15**
Ours (SigLIP-M1)	42.6 / 31.4 / 14.1	88.7	31.5	**41.7**	21.1 / 13.0 / 4.9	51.2	28.3	23.7	54.0 / 39.5 / 15.0	92.7	30.9	46.4
Ours (SigLIP-M2)	42.4 / 31.3 / 14.2	87.9	31.4	41.4	21.3 / 13.8 / 5.4	51.1	28.1	**23.9**	54.0 / 39.5 / 15.2	92.7	30.9	**46.5**
Ours (SigLIP-M3)	42.7 / 30.5 / 14.3	87.3	31.3	41.2	20.8 / 13.0 / 4.9	50.7	27.9	23.5	53.8 / 39.2 / 15.1	92.5	30.8	46.3
Ours (OpenCLIP-M1)	40.8 / 30.0 / 13.2	78.0	30.7	38.5	15.2 / 9.6 / 3.6	34.0	20.4	16.6	52.6 / 38.2 / 14.8	90.3	30.4	45.3
Ours (OpenCLIP-M2)	41.1 / 31.1 / 13.8	79.2	31.4	**39.3**	15.9 / 10.0 / 3.8	35.3	21.0	**17.2**	53.5 / 38.9 / 15.4	91.2	30.7	**45.9**
Ours (OpenCLIP-M3)	39.2 / 29.2 / 12.4	76.0	30.4	37.4	14.5 / 9.3 / 3.4	32.7	20.4	16.1	52.8 / 38.1 / 14.9	90.5	30.2	45.3
Ours (OpenCLIP-M4)	39.6 / 29.2 / 13.5	77.7	29.5	37.9	14.8 / 9.4 / 3.5	33.1	20.2	16.2	53.0 / 38.4 / 15.0	90.8	30.3	45.5
Ours (OpenCLIP-M5)	39.1 / 28.8 / 12.4	73.7	30.1	36.8	14.4 / 9.1 / 3.3	31.9	19.9	15.7	52.3 / 37.7 / 14.7	89.8	29.9	44.8
Ours (CLIP-M1)	41.8 / 31.2 / 13.8	88.6	31.3	41.4	19.7 / 13.0 / 4.9	44.7	25.6	21.6	54.0 / 39.5 / 15.2	92.6	30.9	46.4
Ours (CLIP-M2)	42.6 / 30.9 / 14.5	88.0	31.5	**41.5**	20.1 / 13.3 / 5.1	45.2	25.9	**21.9**	54.1 / 39.6 / 15.3	92.8	31.0	**46.6**
Ours (CLIP-M3)	41.3 / 30.0 / 13.6	85.5	30.8	40.2	19.2 / 12.7 / 4.7	43.9	25.2	21.2	53.7 / 39.1 / 15.0	92.1	30.7	46.1
Ours (PECore-M1)	39.8 / 28.4 / 13.6	79.1	30.3	**38.2**	15.3 / 10.3 / 3.5	37.0	22.5	**17.7**	53.9 / 39.4 / 15.4	91.6	30.7	**46.2**
Ours (PECore-M2)	38.8 / 26.7 / 11.6	68.8	28.5	34.9	13.6 / 8.5 / 2.9	30.4	19.6	15.0	52.1 / 37.3 / 14.5	88.7	29.6	44.4
Ours (PECore-M3)	39.3 / 28.5 / 13.3	75.2	29.3	37.1	14.2 / 9.1 / 3.2	31.8	20.1	15.7	53.0 / 38.5 / 15.1	89.9	30.1	45.3

TABLE VII: **Stage Ablation studies:** For each dataset, we compare the Best model with FAES (Random-ASOT) and SMTS (Baseline-0). Columns report metrics as in Table VI . Drops from Best are shown in blue with downward arrows (↓).

Dataset	Ablated Stage	F1@10	F1@25	F1@50	Edit	Acc	Avg
50 Salads	None	41.84	31.19	13.84	88.58	31.30	41.81
	Stage1 (↓)	6.71 (↓35.13)	5.16 (↓26.03)	2.27 (↓11.57)	9.35 (↓79.23)	28.53 (↓2.77)	10.40 (↓31.41)
	Stage2 (↓)	0.17 (↓41.67)	0.08 (↓31.11)	0.03 (↓13.81)	0.88 (↓87.70)	5.48 (↓25.82)	3.01 (↓38.80)
GTEA	None	21.33	13.76	5.44	51.07	28.08	23.14
	Stage1 (↓)	1.38 (↓19.95)	0.60 (↓13.16)	0.25 (↓5.19)	17.39 (↓33.68)	17.30 (↓10.78)	7.38 (↓15.76)
	Stage2 (↓)	1.18 (↓20.15)	0.52 (↓13.24)	0.16 (↓5.28)	3.95 (↓47.12)	5.41 (↓22.67)	2.64 (↓20.50)
Breakfast	None	54.26	39.60	15.34	92.44	30.89	46.51
	Stage1 (↓)	12.58 (↓41.68)	8.36 (↓31.24)	2.85 (↓12.49)	18.87 (↓73.57)	26.08 (↓4.81)	13.75 (↓32.76)
	Stage2 (↓)	1.16 (↓53.10)	0.67 (↓38.93)	0.27 (↓15.07)	8.13 (↓84.31)	17.50 (↓13.39)	5.95 (↓40.56)

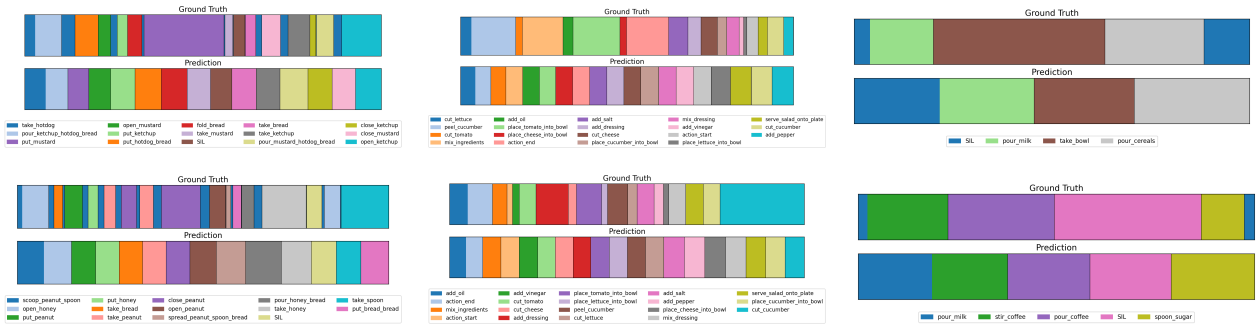


Fig. 3: **Qualitative results:** columns show segmentation results of our method on GTEA, 50 Salads, and Breakfast (left to right), with two examples each.

For instance, in Breakfast, the highest accuracy and F1 scores occur for shorter clips (< 60s), whereas performance progressively deteriorates for videos longer than 120s. GTEA and 50 Salads exhibit the same behavior, indicating that longer videos bring increased temporal variability and amplify error propagation in training-free segmentation. We

TABLE VIII: **Performance variation by video duration:** across different video length bins (in seconds) for Breakfast, GTEA, and 50 Salads datasets.

Breakfast	Acc	Edit	F1@10	F1@25	F1@50	Avg
0–60s	46.31	98.51	76.05	62.39	29.16	62.48
60–120s	40.99	96.28	63.98	50.89	20.26	54.48
≥120s	26.41	82.37	41.40	25.81	8.16	36.83
GTEA	Acc	Edit	F1@10	F1@25	F1@50	Avg
0–60s	27.75	50.65	24.38	17.14	8.38	25.66
60–120s	23.31	33.19	13.54	8.12	3.05	16.24
≥120s	16.82	12.99	6.25	2.27	0.57	7.78
50 Salads	Acc	Edit	F1@10	F1@25	F1@50	Avg
240–360s	30.57	84.37	42.15	31.39	13.73	40.44
360–480s	32.92	75.01	41.73	31.34	14.96	39.19
≥480s	23.40	76.11	28.31	19.88	7.41	31.02

TABLE IX: **Performance variation by number of action segments in ground truth:** Performance across bins of ground-truth segment counts for GTEA, Breakfast, and 50 Salads datasets.

GTEA	Acc	Edit	F1@10	F1@25	F1@50	Avg
20–29	28.23	72.77	48.55	33.53	16.18	39.85
30–39	23.12	42.79	19.11	11.78	4.78	20.32
40–49	22.19	15.14	10.06	5.95	2.27	11.12
Breakfast	Acc	Edit	F1@10	F1@25	F1@50	Avg
0–4	42.79	98.69	71.90	59.28	26.25	59.78
5–9	29.00	88.56	50.52	35.06	12.60	43.15
10–14	25.11	72.61	38.81	23.16	7.82	33.50
15–19	22.14	32.30	21.54	12.73	4.35	18.61
50 Salads	Acc	Edit	F1@10	F1@25	F1@50	Avg
15–19	25.73	83.77	32.79	23.86	10.06	35.24
20–24	32.74	77.48	44.37	33.12	14.08	40.36
25–29	32.00	65.57	38.57	24.62	14.18	34.99

further examine the impact of the number of ground-truth action segments per video (Table IX). In GTEA, where each video comprises many brief segments (mean ∼36), the model performs worse than in Breakfast, which has a mean of ∼5 segments. 50 Salads falls between these two cases. These findings indicate that the density of fine-grained action boundaries has a strong effect on performance, with tightly packed sequences of short actions being especially difficult.

b) : Table XI summarizes segment duration statistics. The mean segment duration in GTEA is only 1.94s, compared to 20.95s in Breakfast and 18.59s in 50 Salads. These results explain the difficulty in GTEA: the model must repeatedly localize boundaries within very short spans, leaving little room for temporal context aggregation. While datasets with longer segments allow the model to achieve better segmentation.

4) Limitations and Future Work: While we perform our zero-shot open-vocabulary action segmentation on 3 prominent action segmentation datasets, future work can explore

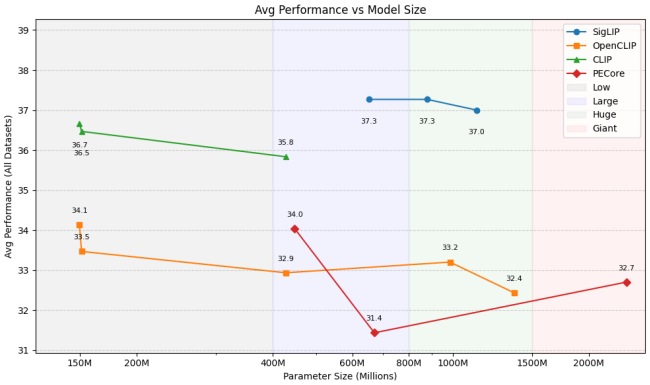


Fig. 4: **Performance vs Model Size.** Models are grouped by family (SigLIP, CLIP, OpenCLIP, PECore). Shaded regions indicate parameter size bins: Low (≤400M), Large (400–800M), Huge (800–1500M), and Giant (>1500M).

TABLE X: Action-type confusion on GTEA (SigLIP-M1).

GT Action Type Accuracy (%)	take 41	put 36	open 35	close 35	pour 37	scoop 21	spread 41	stir 54	fold 37	shake 1	SIL 22
Top Misclassified (%)	put 12	take 17	take 21	take 29	take 15	take 28	put 29	open 15	SIL 24	open 31	put 29

TABLE XI: **Statistics of segment durations** across datasets.

Dataset	Min (s)	Max (s)	Mean (s)
GTEA	0.07	44.73	1.94
Breakfast	0.07	386.00	20.95
50 Salads	0.03	138.47	18.59

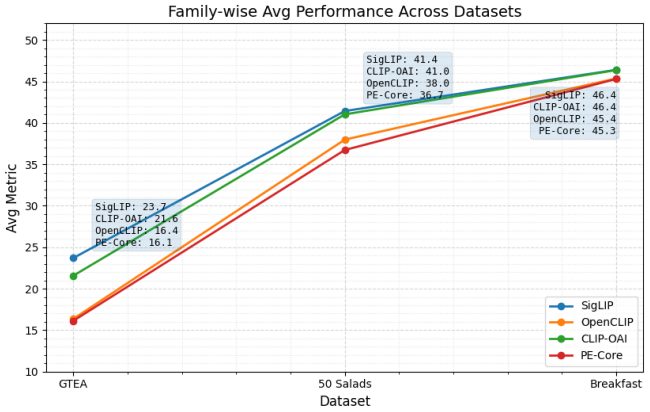


Fig. 5: **VLM Family average of Avg Metric:** across datasets (GTEA, 50 Salads, Breakfast). Each line is a VLM family.

action segmentation datasets. Further, future works can dive deeper into prompt engineering in order to improve the performance. Also enhancing the temporal modeling ability of the optimal transport algorithm is important for such training-free action segmentation pipelines.

VI. Conclusion

We presented *OVTAS*, a training-free, zero-shot pipeline for open-vocabulary temporal action segmentation using pre-trained VLMs. Our two-stage design—Frame–Action Em-

bedding Similarity (FAES) followed by Similarity-Matrix Temporal Segmentation (SMTS)—delivers encouraging performance across standard TAS benchmarks without any task-specific supervision. A comprehensive study of 14 VLMs reveals that semantic alignment quality strongly depends on action-motion discrimination. We release VLM embeddings for all 14 models across three datasets.

Acknowledgements: We acknowledge the Feddersen Distinguished Professorship Funds. This work was also supported by NSF under the Partnership for Innovation: Technology Transfer (PFI-TT) 2329804. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency.

REFERENCES

[1] G. Ding, F. Sener, and A. Yao, "Temporal action segmentation: An analysis of modern techniques," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 2, pp. 1011–1030, 2023.

[2] L. Romeo, R. Marani, A. G. Perri, and J. Gall, "Multi-modal temporal action segmentation for manufacturing scenarios," *Engineering Applications of Artificial Intelligence*, vol. 148, p. 110320, 2025.

[3] G. De Rossi, M. Minelli, S. Roin, F. Falezza, A. Sozzi, F. Ferraguti, F. Setti, M. Bonfè, C. Secchi, and R. Muradore, "A first evaluation of a multi-modal learning system to control surgical assistant robots via action segmentation," *IEEE Transactions on Medical Robotics and Bionics*, vol. 3, no. 3, pp. 714–724, 2021.

[4] X. Chen, W. Chen, D. Lee, Y. Ge, N. Rojas, and P. Kormushev, "A backbone for long-horizon robot task understanding," *IEEE Robotics and Automation Letters*, 2025.

[5] L. Okamoto and P. Parmar, "Hierarchical neurosymbolic approach for comprehensive and explainable action quality assessment," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 3204–3213, 2024.

[6] C. Ju, T. Han, K. Zheng, Y. Zhang, and W. Xie, "Prompting visual-language models for efficient video understanding," in *European conference on computer vision*, pp. 105–124, Springer, 2022.

[7] Y. Yu, C. Cao, Y. Zhang, Q. Lv, L. Min, and Y. Zhang, "Building a multi-modal spatiotemporal expert for zero-shot action recognition with clip," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, pp. 9689–9697, 2025.

[8] M. Xu and S. Gould, "Temporally consistent unbalanced optimal transport for unsupervised action segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14618–14627, 2024.

[9] J. Li and S. Todorovic, "Action shuffle alternating learning for unsupervised action segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12628–12636, 2021.

[10] G. Ding and A. Yao, "Leveraging action affinity and continuity for semi-supervised temporal action segmentation," in *European Conference on Computer Vision*, pp. 17–32, Springer, 2022.

[11] A. Richard, H. Kuehne, A. Iqbal, and J. Gall, "Neuralnetwork-viterbi: A framework for weakly supervised video learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7386–7395, 2018.

[12] H. Khan, S. Haresh, A. Ahmed, S. Siddiqui, A. Konin, M. Z. Zia, and Q.-H. Tran, "Timestamp-supervised action segmentation with graph convolutional networks," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10619–10626, IEEE, 2022.

[13] S. Swetha, H. Kuehne, Y. S. Rawat, and M. Shah, "Unsupervised discriminative embedding for sub-action learning in complex activities," in *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 2588–2592, IEEE, 2021.

[14] A. Richard, H. Kuehne, and J. Gall, "Weakly supervised action learning with rnn based fine-to-coarse modeling," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 754–763, 2017.

[15] J. Li and S. Todorovic, "Set-constrained viterbi for set-supervised action segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10820–10829, 2020.

[16] Y. Ishikawa, S. Kasai, Y. Aoki, and H. Kataoka, "Alleviating over-segmentation errors by detecting action boundaries," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2322–2331, 2021.

[17] H. Kuehne, A. Richard, and J. Gall, "Weakly supervised learning of actions from transcripts," *Computer Vision and Image Understanding*, vol. 163, pp. 78–89, 2017.

[18] Z. Wang, Z. Gao, L. Wang, Z. Li, and G. Wu, "Boundary-aware cascade networks for temporal action segmentation," in *European Conference on Computer Vision*, pp. 34–51, Springer, 2020.

[19] F. Spurio, E. Bahrami, G. Francesca, and J. Gall, "Hierarchical vector quantization for unsupervised action segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, pp. 6996–7005, 2025.

[20] Z. Li, Y. Abu Farha, and J. Gall, "Temporal action segmentation from timestamp supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8365–8374, 2021.

[21] A. Richard, H. Kuehne, and J. Gall, "Action sets: Weakly supervised action segmentation without ordering constraints," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 5987–5996, 2018.

[22] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Fine-grained action segmentation using the semi-supervised action gan," *Pattern Recognition*, vol. 98, p. 107039, 2020.

[23] E. Bueno-Benito, B. T. Vecino, and M. Dimiccoli, "Leveraging triplet loss for unsupervised action segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4922–4930, 2023.

[24] Z. Wang, H. Chen, X. Li, C. Liu, Y. Xiong, J. Tighe, and C. Fowlkes, "Sscap: Self-supervised co-occurrence action parsing for unsupervised temporal action segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1819–1828, 2022.

[25] Z. Song, K. Chen, P. Wang, M. Song, and N. Zheng, "Unsupervised action segmentation via multi-scale temporal-interaction enhancement," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[26] P. A. Knight, "The sinkhorn–knopp algorithm: convergence and applications," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 1, pp. 261–275, 2008.

[27] A. Fathi, X. Ren, and J. M. Rehg, "Learning to recognize objects in egocentric activities," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3281–3288, 2011.

[28] S. Stein and S. J. McKenna, "Combining embedded accelerometers with computer vision for recognizing food preparation activities," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, pp. 729–738, 2013.

[29] H. Kuehne, A. Arslan, and T. Serre, "The language of actions: Recovering the syntax and semantics of goal-directed human activities," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 780–787, 2014.

[30] S. Sarfraz, N. Murray, V. Sharma, A. Diba, L. Van Gool, and R. Stiefelhagen, "Temporally-weighted hierarchical clustering for unsupervised action segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11225–11234, 2021.

[31] D. Bolya, P.-Y. Huang, P. Sun, J. H. Cho, A. Madotto, C. Wei, T. Ma, J. Zhi, J. Rajasegaran, H. Rasheed, *et al.*, "Perception encoder: The best visual embeddings are not at the output of the network," *arXiv preprint arXiv:2504.13181*, 2025.

[32] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

[33] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PmLR, 2021.

[34] G. Ilharco, M. Wortsman, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, *et al.*, "Openclip," *Zenodo*, 2021.