Towards Generalizable Partially Relevant Video Retrieval

Woojin Jun, Cheol-Ho Cho, MinSeok Jung, SuBeen Lee, Jae-Pil Heo*
Sungkyunkwan University
{junwoojin, gersys, alstjr88, leesb7426, jaepilheo}@skku.edu

Abstract

Partially Relevant Video Retrieval (PRVR) aims to retrieve an untrimmed video that contains at least one segment relevant to a text query. Despite its practical motivation, existing PRVR methods are typically trained and evaluated on a single curated benchmark, limiting insight into whether they learn generalizable fine-grained text-video alignment. To examine this issue, we introduce a *multi-source-to-unseen-target* protocol, in which multiple source benchmarks are merged into a single training corpus without source labels and the trained model is evaluated on a held-out target benchmark. Under this protocol, existing PRVR methods generalize poorly to unseen targets despite leveraging the general image-text alignment of CLIP. Surprisingly, they even underperform zero-shot CLIP, suggesting that PRVR training can actively erode the transferable alignment inherited from CLIP. To address this limitation, we propose Generalization-Aware Learning (GAL) for PRVR, a two-branch framework that preserves CLIP-derived transferability while learning PRVR-specific fine-grained text-segment matching. GAL consists of an anchored branch that maintains CLIP's transferability and an adaptive branch that learns task-oriented segment-level alignment. We train these branches through conservative residual updates, per-sample branch specialization, and CLIP-text relation distillation, enabling complementary learning without overwriting the alignment structure provided by CLIP. Under the proposed protocol, GAL achieves state-of-the-art unseen-target retrieval with competitive source performance, showing that fine-grained matching can be learned without collapsing transferable alignment.

1 Introduction

With the rapid growth of video content across diverse platforms, Partially Relevant Video Retrieval (PRVR) [1–8] has emerged as a practical retrieval problem for natural-language access to untrimmed videos. Unlike conventional text-to-video retrieval, PRVR considers a video relevant if it contains at least one segment corresponding to the query.

Despite this practical motivation, existing PRVR methods are typically evaluated on single-source benchmarks curated under shared annotation rules, treating each as an isolated in-domain train-test setting. In real-world retrieval, however, videos and queries may originate from heterogeneous sources that differ not only in visual appearance but also in the semantic scope of queries themselves. Under such conditions, it remains unclear whether existing PRVR methods learn transferable text-video alignment or merely exploit benchmark-specific alignment patterns.

To examine PRVR under such heterogeneous retrieval scenarios, we introduce a *multi-source-to-unseen-target* protocol, in which multiple source benchmarks are merged into a single training corpus without using domain labels and the trained model is evaluated on a held-out target benchmark. Under

*Corresponding author

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

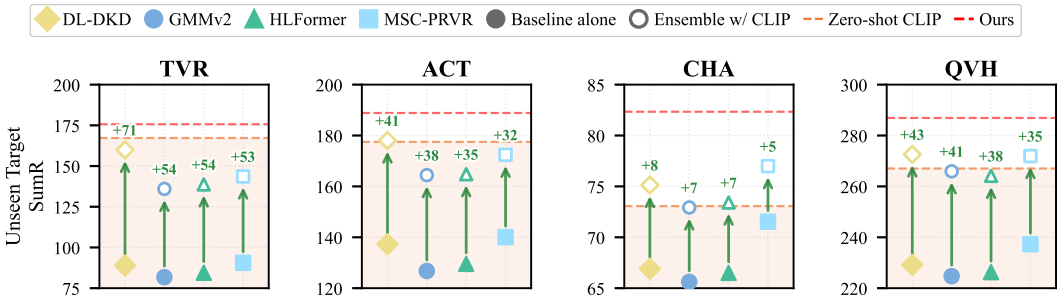


Figure 1: Multi-source-to-unseen-target analysis under a leave-one-dataset-out protocol. Each panel reports retrieval performance in SumR (the sum of R@{1, 5, 10, 100}) on one held-out target dataset after training on the remaining three datasets pooled as a single source corpus. Filled markers denote standalone PRVR baselines, while the orange dashed line marks zero-shot CLIP performance; the shaded region indicates performance below CLIP alone. Hollow markers and green arrows visualize gains from equal-weight post hoc score-level fusion with CLIP, with large gains suggesting that standalone PRVR training broadly overwrites the generalization ability of the frozen backbone. The red dashed line denotes our method, which outperforms the baselines.

this protocol, existing PRVR methods generalize poorly to unseen targets, despite leveraging the general image-text alignment of CLIP [9], as shown in Fig. 1. Surprisingly, they even underperform zero-shot CLIP, which is not fine-tuned for PRVR. This indicates that source-domain PRVR training not only fails to improve cross-source transferability, but can actively erode the transferable alignment inherited from CLIP.

A naïve way to recover cross-domain generalization is to ensemble the trained PRVR model with frozen CLIP at the score level. The frozen CLIP score can complement the PRVR model by providing generalizable image-text alignment that may be weakened during source-domain training. As shown in Fig. 1, this simple score-level ensembling substantially improves retrieval performance on unseen domains. However, it cannot serve as an actual remedy, since CLIP was not pre-trained for fine-grained video-text matching and its scores cannot be expected to consistently complement the segment-level alignment that the trained model has acquired, as shown in Fig. 3 (b). This motivates a training-time design that jointly learns PRVR-specific matching and preserves CLIP-derived transferability, without relying on a single representation in which task adaptation may drift from CLIP alignment or preservation may overly constrain task learning.

To this end, we propose Generalization-Aware Learning (GAL) for PRVR, a two-branch retrieval framework whose two branches are jointly trained to be complementary while retaining the generalization of CLIP: an *anchored* branch that preserves the generalization ability of CLIP and an *adaptive* branch that learns task-oriented segment-level matching. We wire both branches through Layer-Scaled Gated Residual (LSGR) updates initialized to closely preserve frozen CLIP features, providing a conservative starting state in which both branches inherit backbone knowledge before specialization. We then encourage complementary branch roles through Per-Sample Branch Specialization (PSBS), which adaptively assigns stronger learning signals to the branch that better explains each training sample, so that branch specialization emerges from data rather than being statically prescribed. Alongside PSBS, the anchored branch is kept close to the frozen backbone by deriving its text representation from the frozen CLIP text embedding through a skip-connected linear layer and reinforcing this alignment with a CLIP anchoring loss. For the adaptive branch, directly matching frozen CLIP features would overly restrict task-oriented learning; we therefore introduce CLIP-Text Relation Distillation (CTRD), which preserves CLIP-derived text-neighborhood structure without enforcing feature-level alignment. Together, these components make the two branches complementary while preserving the generalization ability of CLIP.

We evaluate the proposed method under a pooled multi-source-to-unseen-target protocol across four PRVR benchmarks, and achieve state-of-the-art unseen-domain retrieval while also surpassing prior baselines on source-domain performance. These results support a simple principle for PRVR domain generalization: effective training should preserve, rather than overwrite, the alignment structure that CLIP already provides.

Overall, our contributions are threefold. First, we formulate generalizable PRVR under a realistic pooled *multi-source-to-unseen-target* protocol and identify that existing PRVR training can erode the cross-domain transferability of CLIP, causing trained models to perform worse than zero-shot CLIP on unseen targets. Second, we propose GAL, a two-branch framework that preserves CLIP's general alignment while learning PRVR-specific fine-grained matching via conservative residual updates, per-sample branch specialization, and text-relation distillation. Third, under the multi-source-to-unseen-target protocol, GAL achieves state-of-the-art unseen-domain retrieval, while also surpassing baselines on source-domain performance.

2 Related Work

Partially Relevant Video Retrieval. Text-to-video retrieval [10–27] assumes whole-video relevance, which breaks down for untrimmed videos where queries typically correspond to only a short segment. Partially Relevant Video Retrieval (PRVR) [5, 28, 4, 29, 7, 8, 6] retrieves untrimmed videos that contain at least one segment matching a text query, requiring fine-grained correspondence between the query and only a subset of video segments. Existing methods build temporal modules on top of frozen CLIP features and improve segment-level matching through multi-scale clip modeling [28, 4, 30], hierarchical alignment [8, 7], or distillation from CLIP-based teachers [3]. In particular, prior CLIP-distillation methods use CLIP as a teacher to strengthen fine-grained PRVR alignment under standard PRVR settings, whereas we use CLIP-derived supervision to preserve transferable semantic structure during task adaptation. However, these works are evaluated on a single benchmark at a time, and PRVR benchmarks [31–34] differ not only in visual style but also in query semantic content, leaving the shared semantic anchors assumed by standard domain generalization largely absent. This motivates our pooled multi-source-to-unseen-target protocol and an approach that leverages the broad semantic coverage of frozen pre-trained backbone. We further discuss how our setting relates to video domain adaptation and generalization in Appendix D.

Preserving Pre-trained Knowledge during Fine-tuning. Fine-tuning CLIP-style zero-shot models can improve target-domain accuracy, sometimes at the cost of robustness under distribution shift [35]; more generally, fine-tuning can distort pre-trained features and hurt out-of-distribution performance [36]. A growing body of work mitigates this issue by keeping the fine-tuned model close to, or explicitly guided by, the pre-trained model, through weight-space ensembling [35], context-aware regularization [37], anchor-based auxiliary supervision [38], pre-trained-model-guided fine-tuning [39], and residual feature distillation for video tasks [40]. These methods primarily aim to balance pre-trained generalization and downstream adaptation within the fine-tuning objective, a strategy well aligned with tasks such as image classification, open-vocabulary recognition, and action recognition, where prediction can largely rely on global semantic alignment. PRVR, however, additionally requires fine-grained segment-level text-video correspondence beyond what frozen CLIP representations directly provide, creating a stronger tension between preserving general-purpose semantics and acquiring task-specific temporal grounding. We therefore explicitly separate these roles into co-trained branches with per-sample specialization, an architectural choice further informed by our analysis of the limitations of inference-time CLIP ensembling.

3 Method

The overall architecture of our GAL is illustrated in Fig. 2. GAL is a two-branch retrieval framework whose branches are co-trained to be complementary while retaining the generalization ability of CLIP [9]. To this end, we leverage an *anchored branch* that preserves CLIP's transferability, and an *adaptive branch* that learns task-oriented segment-level matching. The two branches use asymmetric query architectures and branch-specific training signals, so that their specialization emerges from data rather than being statically prescribed. Intuitively, one branch is kept close to CLIP so that the transferable alignment it already provides is not overwritten, while the other branch is free to learn the fine-grained segment-level matching that PRVR requires; their retrieval scores are combined at inference. The remainder of this section describes how both branches are initialized to inherit CLIP knowledge (Sec. 3.3), how each training sample is routed to the branch that explains it better (Sec. 3.4), and how each branch is regularized toward CLIP (Sec. 3.5).

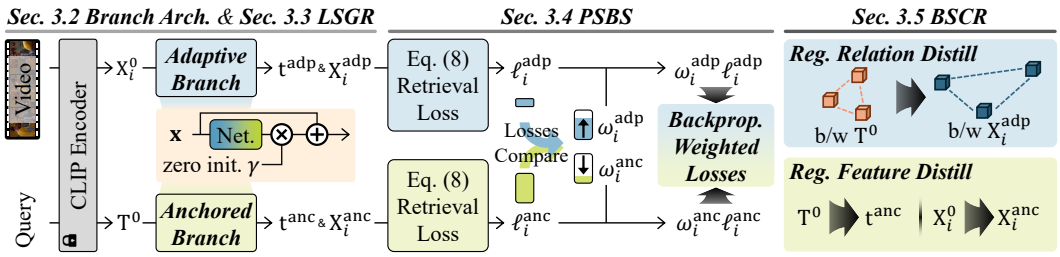


Figure 2: Overview of our method. Our method consists of two complementary branches: an anchored branch that preserves the generalization ability of frozen CLIP, and an adaptive branch that learns task-specific segment-level matching. Both branches are built upon CLIP features with layer-scaled gated residual updates (LSGR) for conservative adaptation. During training, per-sample branch specialization (PSBS) assigns stronger learning signals to the branch that better explains each sample, encouraging complementary behaviors. Additionally, branch-specific regularization enforces CLIP preservation in the anchored branch and CLIP-Text relation distillation in the adaptive branch.

3.1 Preliminaries and Notation

Let $\mathcal{D} = \{(q_i, v_i)\}_{i=1}^{N}$ denote the pooled multi-source training set, where q_i is a text query, v_i is its paired untrimmed video, and N is the number of training pairs. The frozen CLIP text encoder $E_{\text{text}}^{\text{CLIP}}$ produces a sequence of text-token features for q_i:

$$\mathbf{T}_i^0 = E_{\text{text}}^{\text{CLIP}}(q_i) = [\mathbf{t}_{i,\text{SOT}}^0, \mathbf{t}_{i,1}^0, \ldots, \mathbf{t}_{i,L_i^q}^0, \mathbf{t}_{i,\text{EOT}}^0], \qquad \mathbf{T}_i^0 \in \mathbb{R}^{(L_i^q+2)\times D}, \tag{1}$$

where L_i^q is the number of tokens, D denotes the CLIP embedding dimension, and we use $\mathbf{t}_i^0 = \mathbf{t}_{i,\text{EOT}}^0$ as the sentence embedding for short. For each video v_i, the CLIP visual encoder $E_{\text{vis}}^{\text{CLIP}}$ extracts features over L_i^v sampled visual units $v_{i,k}$:

$$\mathbf{X}_i^0 = [\mathbf{x}_{i,1}^0, \ldots, \mathbf{x}_{i,L_i^v}^0] \in \mathbb{R}^{L_i^v \times D}, \qquad \mathbf{x}_{i,k}^0 = E_{\text{vis}}^{\text{CLIP}}(v_{i,k}). \tag{2}$$

Throughout this section, the superscript 0 denotes frozen CLIP features.

3.2 Branch Architecture

We define a skip-connected linear layer as $F(\mathbf{x}) = \mathbf{x} + f(\mathbf{x})$, where f is a learnable linear transformation whose output dimension matches that of $\mathbf{x}$. We denote the branch-specific skip-connected linear layers by $F_{\text{text}}^{\text{anc}}$, $F_{\text{text}}^{\text{adp}}$, $F_{\text{vis}}^{\text{anc}}$, and $F_{\text{vis}}^{\text{adp}}$, each with separate parameters.

Query representations. The architectural asymmetry between the two branches lies in the query representation. The anchored query branch is intentionally shallow, taking only the frozen CLIP EOT feature and applying a single skip-connected linear layer to produce the anchored query feature $\mathbf{t}_i^{\text{anc}}$:

$$\mathbf{t}_i^{\text{anc}} = F_{\text{text}}^{\text{anc}}(\mathbf{t}_i^0), \qquad \mathbf{t}_i^{\text{anc}} \in \mathbb{R}^D. \tag{3}$$

This lightweight design keeps the representation structurally close to frozen CLIP text, limiting drift from the generalizable feature space. In contrast, the adaptive query branch processes the full CLIP text-token sequence:

$$\mathbf{t}_i^{\text{adp}} = \text{Pool}\left(E_{\text{text}}^{\text{adp}}\left(F_{\text{text}}^{\text{adp}}\left(\mathbf{T}_i^0\right)\right)\right), \qquad \mathbf{t}_i^{\text{adp}} \in \mathbb{R}^D. \tag{4}$$

$E_{\text{text}}^{\text{adp}}$ is a Transformer query encoder with residual connections around the self-attention and feed-forward sublayers, and $\text{Pool}(\cdot)$ denotes the modular pooling commonly used in PRVR [28, 4, 2, 8]. This higher-capacity branch learns PRVR-specific representations for fine-grained matching.

Video representations. For video representations, the two branches use the same architectural form but separate parameters. Both branches receive the frozen CLIP visual sequence $\mathbf{X}_i^0$ and produce segment-level video representations:

$$\mathbf{X}_i^{\text{anc}} = E_{\text{vis}}^{\text{anc}}\left(F_{\text{vis}}^{\text{anc}}\left(\mathbf{X}_i^0\right)\right), \qquad \mathbf{X}_i^{\text{adp}} = E_{\text{vis}}^{\text{adp}}\left(F_{\text{vis}}^{\text{adp}}\left(\mathbf{X}_i^0\right)\right). \tag{5}$$

Here, E_{vis}^{b} ($b \in \{\text{anc}, \text{adp}\}$) adopts the same Transformer architecture as the query encoder, and $\mathbf{X}_i^b = [\mathbf{x}_{i,1}^b, \ldots, \mathbf{x}_{i,L_i^v}^b] \in \mathbb{R}^{L_i^v \times D}$ are the segment-level video representations.

Retrieval scoring. Each branch scores a query-video pair using maximum query-segment similarity:

$$s^{\text{anc}}(q_i, v_j) = \max_{1 \le k \le L_j^v} \cos\big(\mathbf{t}_i^{\text{anc}}, \mathbf{X}_{j,k}^{\text{anc}}\big), \qquad s^{\text{adp}}(q_i, v_j) = \max_{1 \le k \le L_j^v} \cos\big(\mathbf{t}_i^{\text{adp}}, \mathbf{X}_{j,k}^{\text{adp}}\big). \tag{6}$$

Here, s^{anc} and s^{adp} denote the anchored and adaptive retrieval scores, which are used by Per-Sample Branch Specialization during training (Sec. 3.4) and combined at inference (Sec. 3.6). The function $\cos(\cdot, \cdot)$ measures cosine similarity between two vectors.

3.3 Layer-Scaled Gated Residual Updates

To provide a conservative initial state in which both branches inherit CLIP knowledge before specialization, we replace each skip-connection with a layer-scaled gated residual update:

$$F_{\text{LSGR}}(\mathbf{x}) = \mathbf{x} + \boldsymbol{\gamma} \odot f(\mathbf{x}), \tag{7}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^D$ is a learnable channel-wise gate, and $\odot$ denotes element-wise multiplication. The gate $\boldsymbol{\gamma}$ is initialized to zero, so each learned module starts as an identity mapping that closely preserves frozen CLIP features at initialization. LSGR is applied to all skip-connected linear layers, as well as to the residual updates within all Transformer encoders.

3.4 Per-Sample Branch Specialization

The query-side asymmetry and layer-scaled gated residuals provide a foundation for branch divergence, but do not explicitly assign distinct roles. We therefore introduce Per-Sample Branch Specialization (PSBS), which assigns stronger learning signals to the branch that better explains each training sample. For a mini-batch of B paired samples $\{(q_i, v_i)\}_{i=1}^B$, we define the per-sample retrieval loss for branch $b \in \{\text{anc}, \text{adp}\}$ using in-batch contrastive learning:

$$\ell_i^b = -\log \frac{\exp(s^b(q_i, v_i)/\tau_{\text{ret}})}{\sum_{j=1}^B \exp(s^b(q_i, v_j)/\tau_{\text{ret}})}. \tag{8}$$

Here, s^b denotes either s^{anc} or s^{adp}, τ_{ret} is a retrieval temperature, and ℓ_i^b corresponds to the per-sample InfoNCE objective commonly used in PRVR. Then, PSBS computes a loss gap $\Delta_i = \ell_i^{\text{adp}} - \ell_i^{\text{anc}}$, which is treated as a variable for each sample when forming the branch weights so that gradients flow only through the weighted losses themselves. From this gap, the branch weights are defined as

$$w_i^{\text{anc}} = \sigma\left(\frac{\Delta_i}{\tau_{\text{psbs}}}\right), \qquad w_i^{\text{adp}} = \sigma\left(-\frac{\Delta_i}{\tau_{\text{psbs}}}\right), \tag{9}$$

where $\sigma(\cdot)$ is the sigmoid function and τ_{psbs} controls the softness of specialization. With these weights, PSBS-weighted retrieval objective is

$$\mathcal{L}_{\text{ret}} = \frac{1}{B} \sum_{i=1}^B \left(w_i^{\text{anc}} \ell_i^{\text{anc}} + w_i^{\text{adp}} \ell_i^{\text{adp}} \right). \tag{10}$$

Intuitively, this weighting concentrates per-sample gradients on the branch that already explains the sample better, so that the two branches gradually specialize on different regions of the training distribution. Through this per-sample weighting, branch specialization emerges from data rather than being statically prescribed.

3.5 Branch-Specific CLIP Regularization

To preserve the generalization ability of CLIP, we apply branch-specific regularization: feature-level anchoring for the anchored branch, and relational distillation for the adaptive branch.

3.5.1 CLIP Anchoring Loss

We dedicate the anchored branch to retaining the generalization ability of CLIP. While the skip-connected query design biases features toward frozen CLIP at initialization, this structural choice alone does not enforce such alignment under retrieval supervision. We therefore reinforce this

alignment with a feature-level anchoring loss, regularizing both anchored query and video features toward frozen CLIP through cosine distance:

$$\mathcal{L}_{\text{anc}} = \frac{1}{B}\sum_{i=1}^{B}\left[1-\cos(\mathbf{t}_i^{\text{anc}}, \mathbf{t}_i^0)\right] + \frac{1}{|\mathcal{M}|}\sum_{(i,k)\in\mathcal{M}}\left[1-\cos(\mathbf{x}_{i,k}^{\text{anc}}, \mathbf{x}_{i,k}^0)\right], \tag{11}$$

where $\mathcal{M}$ is the set of all non-padded frame positions in the batch.

3.5.2 CLIP-Text Relation Distillation

Since directly aligning adaptive features to frozen CLIP would overly restrict task-oriented learning, we introduce CLIP-Text Relation Distillation, which preserves CLIP-derived neighborhoods within the adaptive branch. We use frozen CLIP text rather than visual features as the semantic teacher, since queries describe semantic content while visual features additionally carry appearance factors such as scene style and lighting. As a result, CLIP text neighborhoods are less affected by domain-specific visual appearance than video neighborhoods. For each query q_i, we form a neighbor set $\mathcal{N}(i,\rho)$ in the frozen CLIP text space by selecting the top-ρ fraction of queries with the highest CLIP text similarity:

$$\mathcal{N}(i,\rho) = \arg\max_{\mathcal{S}} \sum_{j\in\mathcal{S}} \cos(\mathbf{t}_i^0, \mathbf{t}_j^0), \quad \text{s.t. } \mathcal{S}\subseteq\mathcal{E}_i,\ |\mathcal{S}| = \lfloor\rho|\mathcal{E}_i|\rfloor, \tag{12}$$

where $\rho\in[0,1]$ and $\mathcal{E}_i$ denotes the eligible index set excluding self-pairs and same-video pairs.

Given these neighbors, we penalize adaptive video pairs only when they are farther apart than their CLIP-text semantic relation allows by more than a margin, without forcing exact distance matching. This is realized through a one-sided squared hinge loss:

$$\mathcal{L}_{\text{ctrd}} = \frac{1}{B|\mathcal{N}(i,\rho)|}\sum_{i=1}^{B}\sum_{j\in\mathcal{N}(i,\rho)}\left[d(\hat{\mathbf{x}}_i^{\text{adp}}, \hat{\mathbf{x}}_j^{\text{adp}}) - d(\mathbf{t}_i^0, \mathbf{t}_j^0) - m\right]_+^2, \tag{13}$$

where $d(a,b) = 1-\cos(a,b)$, $\hat{\mathbf{x}}_i^{\text{adp}}$ is the adaptive segment feature selected by the max-pooling in Eq. (6) for (q_i, v_i), m is a margin, and $[\cdot]_+$ denotes the positive part.

3.6 Training Objective and Inference

The final training objective is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ret}} + \lambda_{\text{anc}}\mathcal{L}_{\text{anc}} + \lambda_{\text{ctrd}}\mathcal{L}_{\text{ctrd}}$, where λ_{anc} and λ_{ctrd} are loss weights. At inference time, the final retrieval score is a fixed weighted combination of the two branch scores, $s(q_i, v_j) = \alpha s^{\text{anc}}(q_i, v_j) + (1-\alpha)s^{\text{adp}}(q_i, v_j)$, where α is a fixed scalar.

4 Experiments

4.1 Experimental Setup

Datasets and metric. We evaluate on four PRVR benchmarks: TVR [31], ActivityNet Captions [34], Charades-STA [32], and QVHighlights [33]. Dataset details are provided in Appendix B. We use Recall at Q (R@Q) as the retrieval metric, measuring the percentage of queries whose correct match appears within the top-Q ranked results. To summarize recall across multiple thresholds, we report SumR, defined as the sum of R@Q over $Q\in 1, 5, 10, 100$. For comparisons with prior methods, we report R@1, R@10, and SumR, which directly measure top-ranked retrieval accuracy. For source-side evaluation and internal analyses, we report SumR as a compact retrieval metric.

Multi-source-to-unseen-target protocol. We evaluate generalizable PRVR under a leave-one-dataset-out multi-source-to-unseen-target protocol. Here, *split* denotes a leave-one-dataset-out configuration, where one dataset is held out as the unseen target and the remaining three datasets are used as sources. For each split, the three source datasets are pooled into a single source training corpus. The source corpus is constructed by concatenating all training text-video pairs without domain labels, per-domain balancing, or target-domain access. The held-out target is never used during training.

Table 1: Multi-source-to-unseen-target retrieval performance comparison on held-out targets. R@1, R@10 and SumR are reported for each unseen target dataset. Values are means over three seeds; subscripted $\pm$ denotes standard deviation. Best in **bold**, second-best underlined.

Method	TVR			ACT			CHA			QVH		
	R@1	R@10	SumR	R@1	R@10	SumR	R@1	R@10	SumR	R@1	R@10	SumR
ZS-CLIP	16.2	41.8	167.2	16.0	46.4	177.5	2.0	13.6	73.1	36.8	72.6	267.0
MS-SL	5.2$\pm$0.6	17.7$\pm$1.0	81.4$\pm$4.0	7.1$\pm$0.2	28.3$\pm$0.2	119.6$\pm$0.6	2.0$\pm$0.2	11.9$\pm$0.4	68.9$\pm$1.5	22.3$\pm$1.0	58.7$\pm$0.5	220.0$\pm$3.8
GMM	4.3$\pm$0.2	15.3$\pm$0.3	71.9$\pm$0.6	7.5$\pm$0.3	29.3$\pm$0.5	123.1$\pm$1.6	1.8$\pm$0.1	11.4$\pm$0.1	66.3$\pm$0.8	20.8$\pm$1.5	55.9$\pm$1.7	212.7$\pm$4.0
DL-DKD	6.2$\pm$0.4	19.8$\pm$1.0	89.4$\pm$5.1	9.1$\pm$0.4	33.5$\pm$0.3	136.2$\pm$0.6	2.3$\pm$0.3	12.3$\pm$0.7	69.0$\pm$1.5	25.9$\pm$0.3	61.3$\pm$1.2	230.8$\pm$2.2
GMMv2	5.2$\pm$0.2	17.6$\pm$0.5	80.9$\pm$2.0	8.1$\pm$0.3	30.3$\pm$0.4	126.2$\pm$1.6	1.8$\pm$0.2	10.9$\pm$0.5	65.2$\pm$0.4	23.4$\pm$0.4	60.3$\pm$0.6	225.3$\pm$1.1
HLF	5.4$\pm$0.2	18.3$\pm$0.2	83.6$\pm$1.3	8.1$\pm$0.1	30.8$\pm$0.7	128.3$\pm$1.8	2.0$\pm$0.1	11.8$\pm$0.5	67.6$\pm$1.1	24.0$\pm$0.2	60.8$\pm$0.3	226.6$\pm$0.2
MSC	6.0$\pm$0.2	20.0$\pm$0.3	89.9$\pm$0.6	9.4$\pm$0.1	34.4$\pm$0.3	138.9$\pm$1.0	2.1$\pm$0.2	12.6$\pm$0.4	71.9$\pm$1.4	27.3$\pm$0.7	64.5$\pm$0.4	239.0$\pm$0.8
Ours	**17.9**$\pm$0.1	**44.2**$\pm$0.1	**175.4**$\pm$0.4	**17.8**$\pm$0.1	**50.1**$\pm$0.1	**188.7**$\pm$0.3	**2.7**$\pm$0.1	**15.6**$\pm$0.4	**82.1**$\pm$0.4	**43.1**$\pm$0.8	**77.8**$\pm$0.3	**286.7**$\pm$0.2

Table 2: Source-domain retrieval performance comparison. Each source-domain score reports mean SumR over three source-containing splits and three seeds; subscripted $\pm$ denotes standard deviation over the resulting nine measurements. Src-Wise Avg. averages the four source-domain scores, and Src-Pool Avg. follows the protocol definition in Sec. 4.1. Best in **bold**, second-best underlined.

Source	ZS CLIP	MS-SL	GMMFormer	DL-DKD	GMMF. v2	HLFormer	MSC-PRVR	Ours
TVR	167.2	248.8$_{\pm 1.1}$	243.5$_{\pm 1.5}$	248.8$_{\pm 1.5}$	257.6$_{\pm 1.8}$	257.0$_{\pm 1.8}$	**261.3**$_{\pm 1.7}$	247.3$_{\pm 1.6}$
ACT	177.5	193.7$_{\pm 2.0}$	190.9$_{\pm 2.6}$	197.5$_{\pm 2.6}$	197.0$_{\pm 2.5}$	197.1$_{\pm 2.8}$	203.6$_{\pm 2.9}$	**209.6**$_{\pm 1.7}$
CHA	73.1	98.0$_{\pm 1.4}$	92.1$_{\pm 2.8}$	92.3$_{\pm 2.1}$	97.6$_{\pm 2.0}$	97.3$_{\pm 1.9}$	**100.0**$_{\pm 1.8}$	94.9$_{\pm 1.0}$
QVH	267.0	252.6$_{\pm 8.2}$	242.2$_{\pm 8.7}$	261.0$_{\pm 6.2}$	257.6$_{\pm 6.9}$	259.6$_{\pm 5.9}$	267.6$_{\pm 6.5}$	**298.3**$_{\pm 1.7}$
Src-Wise Avg.	171.2	198.3	192.2	199.9	202.4	202.7	208.1	**212.5**
Src-Pool Avg.	155.8	201.8	196.9	202.9	206.6	206.3	**211.6**	210.2

For each trained method, we use three random seeds and aggregate performance according to the evaluation setting. For unseen-target evaluation, each per-target score averages three seeds, and *Unseen-target Avg.* further averages across the four held-out targets, yielding 12 measurements in total. For source-side evaluation, *Src-Wise Avg.* first computes the reported recall metric on each source validation set separately, then averages it across four splits, three source datasets per split, and three seeds yielding 36 measurements. *Src-Pool Avg.* first computes the reported recall metric on a pooled source validation corpus that merges all source validation sets within each split, then averages the results across four splits and three seeds yielding 12 measurements. For deterministic zero-shot CLIP, the seed dimension does not apply. Detailed aggregation rules are provided in Appendix A.

Baselines. We compare GAL-PRVR against six representative PRVR methods: MS-SL [28], GMMFormer [4], GMMFormer v2 [30], DL-DKD [3], MSC-PRVR [7], and HLFormer [8], all using the same frozen CLIP ViT-L/14 backbone with their default architectures and hyperparameters from the original papers. We additionally report a zero-shot CLIP baseline that ranks each query-video pair by the maximum cosine similarity between the query EOT feature and per-frame CLIP visual features.

Implementation details. Following standard PRVR practice [2, 5, 7], we use CLIP ViT-L/14 as the frozen backbone and sample up to 128 frames per video. We train all methods, including GAL, for up to 100 epochs at batch size 256 with three random seeds. All methods follow the same multi-source-to-unseen-target protocol and training infrastructure for fair comparison. For model selection, which is itself a non-trivial component of Domain-Generalization [41], we use source-pooled SumR on the pooled source validation set as the early-stopping criterion, with identical patience applied to all methods. For GAL, the Transformer encoders in both branches follow the GMMFormer v2 [30] architecture, except that the input projection is replaced with a skip-connected linear layer (Sec. 3.2) to preserve the CLIP feature dimension. We set $\rho = 0.4$, $m = 0.05$, $\tau_{\text{psbs}} = 1.0$, $\lambda_{\text{anc}} = 1.0$, $\lambda_{\text{ctrd}} = 0.5$, and $\alpha = 0.6$, fixed across all four leave-one-out splits without per-target tuning. Full training configurations are provided in Appendix B.

Table 3: Ablation study of the proposed components. Results are reported in SumR. Src-Wise Avg., Src-Pool Avg., and Unseen-target Avg. follow the protocol definition in Sec. 4.1. Values are means over three seeds; subscripted $\pm$ denotes standard deviation. Best in **bold**, second-best underlined.

	Method	Source (SumR)		Unseen Target (SumR)				
		Src-Wise Avg.	Src-Pool Avg.	TVR	ACT	CHA	QVH	Avg
	Zero-shot CLIP	171.2	155.8	167.2	177.5	73.1	267.0	171.2
(a)	Baseline	191.1	197.4	$52.2_{\pm 16.4}$	$115.5_{\pm 1.6}$	$66.8_{\pm 2.9}$	$233.2_{\pm 2.9}$	116.9
(b)	+ Anchored query path	192.9	196.7	$57.6_{\pm 23.8}$	$129.6_{\pm 11.2}$	$73.0_{\pm 0.9}$	$246.5_{\pm 2.6}$	126.7
(c)	+ LSGR	212.2	**211.7**	$171.5_{\pm 0.5}$	$182.1_{\pm 0.6}$	$79.2_{\pm 0.7}$	$279.8_{\pm 3.5}$	178.2
(d)	+ PSBS	211.9	210.2	$174.5_{\pm 1.4}$	$185.6_{\pm 0.7}$	$80.0_{\pm 1.1}$	$284.6_{\pm 2.2}$	181.2
(e)	+ CLIP Anchoring	212.0	210.3	$174.8_{\pm 0.5}$	$186.8_{\pm 0.9}$	$81.7_{\pm 0.7}$	$284.2_{\pm 1.0}$	181.9
(f)	+ CTRD (Full model)	**212.5**	210.2	$\mathbf{175.4}_{\pm 0.4}$	$\mathbf{188.7}_{\pm 0.3}$	$\mathbf{82.1}_{\pm 0.4}$	$\mathbf{286.7}_{\pm 0.2}$	**183.2**

4.2 Comparison with the State-of-the-Art

Multi-source-to-unseen-target retrieval performance. Tab. 1 reports unseen-target retrieval performance under the multi-source-to-unseen-target protocol. A consistent pattern emerges across the trained baselines. Despite being trained on multiple source datasets, they substantially underperform zero-shot CLIP on unseen targets, especially on TVR, where all trained baselines fall far below it in all recall metrics. This suggests that existing multi-source PRVR training can compromise cross-domain transferability inherited from the frozen CLIP backbone. In contrast, GAL consistently improves over CLIP and all trained baselines across all held-out targets and both recall cutoffs. Notably, GAL improves SumR over zero-shot CLIP by 8.2 on TVR, 11.2 on ACT, 9.0 on CHA, and 19.7 on QVH, with the largest gain on the most diverse target QVH. These results show that preserving CLIP-derived generalization while learning PRVR-specific matching is crucial for robust unseen-domain retrieval. Full per-Q recall results, including source-side and unseen-target evaluations, are provided in Appendix C, showing consistent trends across multiple recall thresholds.

Source-domain retrieval performance. We further report source-domain SumR in Tab. 2 to verify that the unseen-target gains in Tab. 1 are not obtained by sacrificing source-domain performance. GAL achieves the best Src-Wise Avg. and competitive Src-Pool Avg., showing robust performance under both separate source-domain evaluation and mixed-source retrieval. At the dataset level, GAL ranks first on ACT and QVH, with a particularly large margin on QVH. Notably, on QVH, the trained baselines barely match or even fall below zero-shot CLIP despite using QVH as a source, suggesting that source-domain training does not necessarily improve validation-domain transfer when the domain exhibits high diversity. On TVR and CHA, GAL trails the strongest source-side baseline, possibly due to benchmark-specific query characteristics that are less amenable to generalization, such as show-specific entities in TVR or short action-centric phrases in CHA. Overall, these results show that GAL improves unseen-target retrieval while retaining competitive source-domain performance.

4.3 Ablation Study

Component-wise ablation. In Tab. 3, we ablate the proposed components using SumR. The Baseline (a) substantially underperforms zero-shot CLIP on every unseen target, reproducing the failure mode of existing methods reported in Tab. 1. It follows the conventional multi-branch PRVR training scheme, in which each branch is optimized with its own retrieval loss and the two scores are fused only at inference. The anchored query path (b), a skip-connected linear layer over the frozen CLIP EOT feature, yields only a marginal and unstable gain, insufficient to recover transferability. Wiring both branches with LSGR (c) provides a conservative starting state in which both branches inherit CLIP knowledge before specialization, yielding a large jump in unseen performance and a sharp reduction in variance, while also improving source-side performance. Building on this state, PSBS (d) introduces per-sample branch specialization that encourages the two branches to take complementary roles, further improving unseen performance. Adding branch-specific CLIP regularization (e, f) further specializes each branch and improves unseen-target performance without sacrificing source-side performance. Together, these components yield the best Unseen-target Avg., with branch specialization and regularization driving the gains beyond the conservative learning.

Table 4: Single-branch control in SumR. All variants use a single seed (9527); values therefore differ slightly from the three-seed means in Tab. 3. Best in **bold**.

Model	Source (SumR)		Unseen Target (SumR)				
	Src-Wise Avg.	Src-Pool Avg.	TVR	ACT	CHA	QVH	Avg.
Single branch (bare)	193.8	199.2	56.0	123.9	65.9	229.9	118.9
+ LSGR	205.1	204.6	155.9	171.0	72.5	267.3	166.6
+ LSGR + CTRD	205.9	205.5	156.3	173.0	76.2	267.9	168.4
+ LSGR + CLIP Anchoring	204.9	201.9	171.2	188.5	79.2	283.1	180.5
GAL (two branches)	**212.8**	**210.4**	**175.7**	**188.9**	**82.3**	**286.8**	**183.4**

Table 5: Computational cost on a 1,000-video TVR gallery with 100 queries. Offline costs are incurred once when constructing the gallery; online costs are incurred per query. †A 768-d GMMFormer v2 is included as a width-matched control.

		Offline (ms/video)		Online (ms/query)			
Model	Params (M)	CLIP extract	Video enc.	Query enc.	Similarity	Total	GFLOPs
GMMFormer v2	30.5	2,107	0.64	0.03	0.03	0.06	2.6
GMMFormer v2 (768-d)†	119.7	2,107	1.49	0.07	0.04	0.11	10.2
MSC-PRVR	30.7	2,107	0.64	0.03	0.03	0.06	2.6
HLFormer	28.1	2,107	0.77	0.01	0.03	0.04	2.4
GAL	120.2	2,107	2.50	0.08	0.07	0.15	16.3

Single-branch control. Since LSGR yields the largest single gain, we test whether conservative CLIP-preserving adaptation alone explains the improvement using single-branch variants (Tab. 4). LSGR and CLIP anchoring yield a strong single branch, yet the two-branch GAL still improves every unseen target and both source averages, showing that separating CLIP preservation from task adaptation provides gains beyond conservative adaptation alone.

Computational cost. As shown in Tab. 5, GAL has more trainable parameters and FLOPs than the baselines, but raw CLIP extraction, shared by all methods, dominates the one-time offline cost, and its online latency remains 0.15 ms/query. The extra cost stems from the 768-d hidden size rather than the two branches: widening GMMFormer v2 to 768-d lowers its unseen-target Avg. from 124.7 to 102.5 (single seed), and all variants in Tab. 3 share the 768-d setting.

Hyperparameter sensitivity. Tab. 6 reports the sensitivity of GAL to the two CTRD hyperparameters, the neighbor ratio ρ (left) and the margin m (right), varying each while fixing the other at its default. Both Unseen-target Avg. and source-side Avg. remain within roughly 2 and 1 SumR points across all ρ and m values, indicating that GAL is robust to these choices and does not require per-target tuning. The defaults ($\rho = 0.4$, $m = 0.05$) sit near the best of both source and unseen target performance, and increasing ρ toward 1.0 slightly reduces unseen performance, supporting the use of a focused semantic neighborhood in CTRD over distilling from all available pairs.

4.4 Analysis

Branch specialization emerges from data. To verify that PSBS produces meaningful per-sample routing rather than a degenerate 0.5 assignment, we accumulate the anchored-branch weight w_i^{anc} as a per-query exponential moving average over training and inspect its distribution on the TVR-target run, separated by the source dataset of each training query, as shown in Fig. 3a. The distribution is clearly bimodal, with one mode near $w^{\mathrm{anc}} \approx 0$ corresponding to adaptive specialization and another near $w^{\mathrm{anc}} \approx 0.5$ corresponding to balanced co-training; almost no mass falls above 0.6. This shows that PSBS actually exercises per-sample routing. The routing is also strongly source-dependent, even though PSBS observes no source labels at training time: Charades queries align largely with the adaptive-specialization mode, QVHighlights queries cluster around the balanced mode, and ActivityNet queries occupy both modes. This data-driven, source-aware partitioning supports our claim that branch specialization emerges from data rather than being statically prescribed.

Branch complementarity at inference. We examine whether the inference-time combination of the specialized branches is complementary rather than arbitrary score averaging. Fig. 3b sweeps the

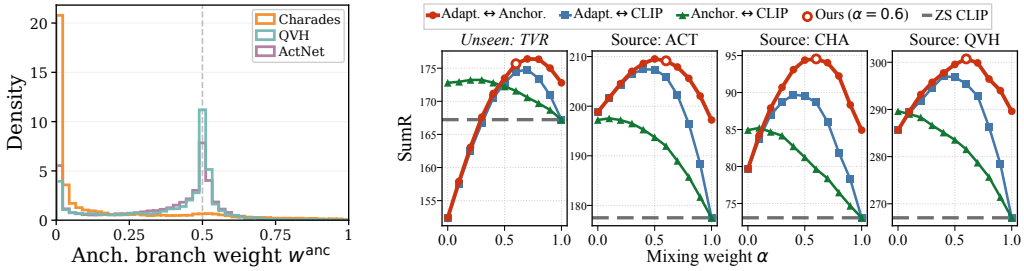
Table 6: Hyperparameter sensitivity of GAL on the two CTRD hyperparameters. We report SumR with a single seed; source-side and unseen-target averages follow the aggregation rules in Sec. 4.1, except that the seed dimension is not averaged. Defaults are shown in **bold**.

(a) Neighbor ratio ρ sensitivity

ρ	Source (SumR)		Unseen (SumR)				
	Src-Wise	Src-Pool	TVR	ACT	CHA	QVH	Avg.
0.2	212.2	210.1	175.7	188.1	83.4	288.6	183.9
0.4	**212.8**	**210.4**	**175.7**	**188.9**	**82.3**	**286.8**	**183.4**
0.6	211.7	209.8	176.1	189.1	79.8	288.1	183.3
0.8	212.7	210.7	175.8	188.7	78.3	285.0	181.9
1.0	212.2	210.1	174.0	187.6	81.7	284.8	182.0

(b) Margin m sensitivity

m	Source		Unseen
	Src-Wise	Src-Pool	Avg.
0	212.6	210.6	183.7
0.05	**212.8**	**210.4**	**183.4**
0.1	211.9	209.2	183.6



(a) PSBS weight w^{anc} distribution

(b) Branch complementarity under inference-time score mixing

Figure 3: Analyses of the trained two-branch model: branch specialization emerges from data during training (left), and the resulting branches reinforce each other at inference (right).

mixing weight α under SumR for three score pairs, denoted as "left ↔ right", on the unseen target and three source domains. Here, α weights the right score: $\alpha = 0$ uses only the left score and $\alpha = 1$ only the right; The open circle on the red curve marks our default $\alpha = 0.6$ used by GAL. Mixing the adaptive and anchored branches (red) yields a broad near-best plateau across all domains with the default point inside it, indicating complementary rather than redundant signals. In contrast, mixing either branch with frozen zero-shot CLIP (blue, green) reproduces the post-hoc CLIP failure mode in Fig. 1: increasing the CLIP weight improves the unseen target but can degrade source domains, since CLIP is not trained for fine-grained text-video matching. Thus, joint training makes the two branches mutually reinforcing, avoiding the unseen–source trade-off induced by frozen CLIP mixing. Further analyses on source-dataset scaling and preservation of the CLIP neighborhood structure are provided in Appendices E and F.

5 Conclusion & Limitation

Conclusion. We formulate generalizable Partially Relevant Video Retrieval (PRVR) under a multi-source-to-unseen-target protocol and show that existing PRVR training erodes the cross-domain transferability of CLIP. To address this, we propose GAL, a two-branch framework whose anchored and adaptive branches are co-trained to be complementary while retaining the generalization of CLIP, through conservative initialization, per-sample branch specialization, and CLIP-text relation distillation. Across four PRVR benchmarks, GAL achieves state-of-the-art unseen-domain retrieval while retaining competitive source-domain performance.

Limitations. GAL does not lead on every source domain: on TVR and CHA, baselines benefit more from source-fitting on benchmark-specific queries such as show-specific entities or short action-centric phrases. This points to adaptive mixing between source-fitting and CLIP preservation as an interesting direction for future work, building on our method.

Acknowledgements

This work was supported in part by MSIT/IITP (No. RS-2022-II220680, RS-2020-II201821, RS-2019-II190421, RS-2024-00459618, RS-2024-00360227, RS-2026-25549037, RS-2025-25442569), MSIT/NIPA (No. RS-2026-25621569), MSIT/NRF (No. RS-2024-00357729), and KNPA/KIPoT (No. RS-2025-25393280).

References

[1] Xianke Chen, Daizong Liu, Xun Yang, Xirong Li, Jianfeng Dong, Meng Wang, and Xun Wang. Prvr: Partially relevant video retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[2] WonJun Moon, Cheol-Ho Cho, Woojin Jun, Taeoh Kim, Inwoong Lee, Dongyoon Wee, Minho Shim, and Jae-Pil Heo. Prototypes are balanced units for efficient and effective partially relevant video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21789–21799, 2025.

[3] Jianfeng Dong, Minsong Zhang, Zheng Zhang, Xianke Chen, Daizong Liu, Xiaoye Qu, Xun Wang, and Baolong Liu. Dual learning with dynamic knowledge distillation for partially relevant video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11302–11312, 2023.

[4] Yuting Wang, Jinpeng Wang, Bin Chen, Ziyun Zeng, and Shu-Tao Xia. Gmmformer: Gaussian-mixture-model based transformer for efficient partially relevant video retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5767–5775, 2024.

[5] Cheol-Ho Cho, WonJun Moon, Woojin Jun, MinSeok Jung, and Jae-Pil Heo. Ambiguity-restrained text-video representation learning for partially relevant video retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 2500–2508, 2025.

[6] WooJin Jun, WonJun Moon, Cheol-Ho Cho, MinSeok Jung, and Jae-Pil Heo. Bridging the semantic granularity gap between text and frame representations for partially relevant video retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4166–4174, 2025.

[7] WonJun Moon, MinSeok Jung, Gilhan Park, Tae-Young Kim, Cheol-Ho Cho, Woojin Jun, and Jae-Pil Heo. Mitigating semantic collapse in partially relevant video retrieval. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[8] Jun Li, Jinpeng Wang, Chaolei Tan, Niu Lian, Long Chen, Yaowei Wang, Min Zhang, Shu-Tao Xia, and Bin Chen. Enhancing partially relevant video retrieval with hyperbolic learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23074–23084, 2025.

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[10] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022.

[11] Yuqi Liu, Pengfei Xiong, Luhui Xu, Shengming Cao, and Qin Jin. Ts2-net: Token shift and selection transformer for text-video retrieval. In *European conference on computer vision*, pages 319–335. Springer, 2022.

[12] Siteng Huang, Biao Gong, Yulin Pan, Jianwen Jiang, Yiliang Lv, Yuyuan Li, and Donglin Wang. Vop: Text-video co-operative prompt tuning for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6565–6574, 2023.

[13] Haoran Tang, Meng Cao, Jinfa Huang, Ruyang Liu, Peng Jin, Ge Li, and Xiaodan Liang. Muse: Mamba is efficient multi-scale learner for text-video retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7238–7246, 2025.

[14] Wentao Ma, Qingchao Chen, Tongqing Zhou, Shan Zhao, and Zhiping Cai. Using multimodal contrastive knowledge distillation for video-text retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[15] Peiyan Guan, Renjing Pei, Bin Shao, Jianzhuang Liu, Weimian Li, Jiaxi Gu, Hang Xu, Songcen Xu, Youliang Yan, and Edmund Y Lam. Pidro: Parallel isomeric attention with dynamic routing for text-video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11164–11173, 2023.

[16] Bangxiang Lan, Ruobing Xie, Ruixiang Zhao, Xingwu Sun, Zhanhui Kang, Gang Yang, and Xirong Li. Hybrid-tower: Fine-grained pseudo-query interaction and generation for text-to-video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24497–24506, 2025.

[17] Zengrong Lin, Zheng Wang, Tianwen Qian, Pan Mu, Sixian Chan, and Cong Bai. Neighborretr: Balancing hub centrality in cross-modal retrieval. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9263–9273, 2025.

[18] Chan Hur, Jeong-hun Hong, Dong-hun Lee, Dabin Kang, Semin Myeong, Sang-hyo Park, and Hyeyoung Park. Narrating the video: boosting text-video retrieval via comprehensive utilization of frame-level captions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24077–24086, 2025.

[19] Huakai Lai, Guoxin Xiong, Huayu Mai, Xiang Liu, and Tianzhu Zhang. Rethinking noisy video-text retrieval via relation-aware alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9231–9241, 2025.

[20] Arun Reddy, Alexander Martin, Eugene Yang, Andrew Yates, Kate Sanders, Kenton Murray, Reno Kriz, Celso M De Melo, Benjamin Van Durme, and Rama Chellappa. Video-colbert: Contextualized late interaction for text-to-video retrieval. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19691–19701, 2025.

[21] Peng Jin, Hao Li, Zesen Cheng, Kehan Li, Xiangyang Ji, Chang Liu, Li Yuan, and Jie Chen. Diffusionret: Generative text-video retrieval with diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2470–2481, 2023.

[22] Kaibin Tian, Ruixiang Zhao, Zijie Xin, Bangxiang Lan, and Xirong Li. Holistic features are almost sufficient for text-to-video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17138–17147, 2024.

[23] Jiamian Wang, Guohao Sun, Pichao Wang, Dongfang Liu, Sohail Dianat, Majid Rabbani, Raghuveer Rao, and Zhiqiang Tao. Text is mass: Modeling as stochastic embedding for text-video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16551–16560, 2024.

[24] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. Cap4video: What can auxiliary captions do for text-video retrieval? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10704–10713, 2023.

[25] Renjing Pei, Jianzhuang Liu, Weimian Li, Bin Shao, Songcen Xu, Peng Dai, Juwei Lu, and Youliang Yan. Clipping: Distilling clip-based models with a student base for video-language retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18983–18992, 2023.

[26] Jian Xiao, Zijie Song, Jialong Hu, Hao Cheng, Jia Li, Zhenzhen Hu, and Richang Hong. Rebalancing contrastive alignment with bottlenecked semantic increments in text-video retrieval. *arXiv preprint arXiv:2505.12499*, 2025.

[27] Jiamian Wang, Pichao Wang, Dongfang Liu, Qiang Guan, Sohail Dianat, Majid Rabbani, Raghuveer Rao, and Zhiqiang Tao. Diffusion-inspired truncated sampler for text-video retrieval. *Advances in Neural Information Processing Systems*, 37:3882–3906, 2024.

[28] Jianfeng Dong, Xianke Chen, Minsong Zhang, Xun Yang, Shujie Chen, Xirong Li, and Xun Wang. Partially relevant video retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 246–257, 2022.

[29] Yi Pan, Yujia Zhang, Michael Kampffmeyer, and Xiaoguang Zhao. Propy: Building interactive prompt pyramids upon clip for partially relevant video retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 519–533, 2025.

[30] Yuting Wang, Jinpeng Wang, Bin Chen, Tao Dai, Ruisheng Luo, and Shu-Tao Xia. Gmmformer v2: An uncertainty-aware framework for partially relevant video retrieval. *arXiv preprint arXiv:2405.13824*, 2024.

[31] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 447–463. Springer, 2020.

[32] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275, 2017.

[33] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858, 2021.

[34] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017.

[35] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7959–7971, 2022.

[36] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

[37] Xiaofeng Mao, Yufeng Chen, Xiaojun Jia, Rong Zhang, Hui Xue, and Zhao Li. Context-aware robust fine-tuning. *International Journal of Computer Vision*, 132(5):1685–1700, 2024.

[38] Jinwei Han, Zhiwen Lin, Zhongyisun Sun, Yingguo Gao, Ke Yan, Shouhong Ding, Yuan Gao, and Gui-Song Xia. Anchor-based robust finetuning of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26919–26928, 2024.

[39] Sibo Wang, Jie Zhang, Zheng Yuan, and Shiguang Shan. Pre-trained model guided fine-tuning for zero-shot adversarial robustness. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24502–24511, 2024.

[40] Xiaohu Huang, Hao Zhou, Kun Yao, and Kai Han. Froster: Frozen clip is a strong teacher for open-vocabulary action recognition. *arXiv preprint arXiv:2402.03241*, 2024.

[41] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[42] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6320–6329. IEEE, 2019.

[43] Aadarsh Sahoo, Rutav Shah, Rameswar Panda, Kate Saenko, and Abir Das. Contrast and mix: Temporal contrastive video domain adaptation with background mixing. *Advances in neural information processing systems*, 34:23386–23400, 2021.

[44] Zhiyu Yao, Yunbo Wang, Jianmin Wang, Philip S Yu, and Mingsheng Long. Videodg: Generalizing temporal relations in videos to novel domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7989–8004, 2021.

A Detailed Evaluation Protocol and Aggregation

We provide a detailed description of the multi-source-to-unseen-target protocol and the aggregation rules used in the main tables.

Leave-one-dataset-out splits. We evaluate on four PRVR benchmarks: TVR [31], ActivityNet Captions [34], Charades-STA [32], and QVHighlights [33]. We construct four leave-one-dataset-out splits. In each split, one dataset is held out as the unseen target, and the remaining three datasets are used as source datasets. For example, when TVR is the held-out target, ActivityNet Captions, Charades-STA, and QVHighlights are used as source datasets. Throughout this section, a split refers to one such leave-one-dataset-out configuration.

Training and model selection. For each split, the source training corpus is constructed by concatenating all training query-video pairs from the three source datasets. We do not use domain labels, per-domain balancing, or any target-domain data during training. Model selection is performed using only source validation data. Specifically, we merge the validation queries and videos from all source datasets into a single pooled source validation set. Early stopping and checkpoint selection are based on the validation metric measured on this pooled source validation set. The held-out target dataset is used only for final evaluation.

Retrieval metrics. For each evaluation set, we compute Recall at Q (R@Q), which measures the percentage of queries whose ground-truth video appears within the top-Q ranked results. For comparisons with prior methods, Tab. 1 reports R@1, R@10 and SumR on held-out target datasets. Here, SumR denotes the sum of R@Q over $Q \in \{1, 5, 10, 100\}$. For source-domain evaluation, Tab. 2 reports SumR. For internal analyses, including ablation studies, hyperparameter sensitivity, and branch mixing analysis, we report SumR. Full recall results are provided in Appendix C when applicable.

Unseen-target scores in Tab. 1. Tab. 1 reports unseen-target performance only. For each held-out target dataset and each random seed, we compute R@1, R@10 and SumR on the validation set of the held-out target. The individual target columns report the mean over three seeds for the corresponding held-out target split. For example, the TVR columns report performance when TVR is held out as the unseen target and the model is trained on ActivityNet Captions, Charades-STA, and QVHighlights. The Avg. columns average the corresponding recall metric over the four held-out targets and three seeds. For deterministic zero-shot CLIP, the seed dimension does not apply, so the all columns average four target-dataset measurements.

Source-domain scores in Tab. 2. Tab. 2 reports source-domain SumR. Each dataset-level source score is computed when that dataset is used as a source domain. A dataset appears as a source dataset in three leave-one-dataset-out splits. For example, TVR is used as a source when ActivityNet Captions, Charades-STA, or QVHighlights is held out as the target. Therefore, each dataset-level source score averages three source-containing splits and three random seeds, resulting in nine SumR measurements for trained methods. The Src-Wise Avg. row averages the four dataset-level source scores. The Src-Pool Avg. row measures source-side performance under a mixed-source validation corpus. For each split and each random seed, we form a pooled source validation corpus by merging the validation queries and videos from all source datasets. We then compute SumR on this pooled source validation corpus. Since there are four splits and three random seeds, Src-Pool Avg. averages 12 measurements for trained methods. For deterministic zero-shot CLIP, the seed dimension does not apply. Accordingly, dataset-level source scores average three source-containing split measurements, Src-Wise Avg. averages 12 source-dataset measurements, and Src-Pool Avg. averages four split-level measurements.

Ablation scores in Tab. 3. Tab. 3 reports SumR for the ablation study. The source-side columns follow the same Src-Wise Avg. and Src-Pool Avg. aggregation rules as Tab. 2. The individual unseen-target columns report the mean SumR over three seeds for each held-out target split. The unseen-target Avg. column averages SumR over the four held-out targets and three seeds.

Hyperparameter sensitivity in Tab. 6. Tab. 6 reports single-seed SumR for hyperparameter sensitivity analysis. The source-side and unseen-target averages follow the same aggregation rules

as Tab. 3, except that the seed dimension is not averaged. Thus, Src-Wise Avg. averages over four splits and three source datasets per split, Src-Pool Avg. averages over four split-level pooled-source measurements, and the unseen-target Avg. averages over four held-out targets.

Standard deviations. For trained methods, subscripted $\pm$ values denote standard deviation. In Tab. 1, the standard deviation of each target-specific score is computed over three random seeds. In Tab. 2, the standard deviation of each dataset-level source score is computed over the nine measurements from three source-containing splits and three random seeds. For aggregate rows or columns such as Src-Wise Avg., Src-Pool Avg., and unseen-target Avg., we report the averaged score and omit standard deviations for readability unless otherwise specified.

B Additional Implementation Details

Datasets. To evaluate PRVR methods, we use four large-scale datasets: TVR, ActivityNet Captions, QVHighlights, and Charades.

TVR [31] consists of 19,524 videos collected from five TV shows, where each video is paired with five language descriptions. Following previous works [4, 7, 8, 28], we use 87,175 and 17,435 moment-text pairs for training and testing, respectively.

ActivityNet Captions [34] contains 9,043 training and 4,430 testing YouTube videos, with 33,721 and 15,753 text queries, respectively, averaging 3.7 descriptions per video.

QVHighlights [33] contains diverse user-generated lifestyle videos captured from various viewpoints and devices. Since dense annotations are unavailable for the official test split, we use the original training and validation sets for PRVR evaluation [7]. the dataset contains 2,214 training videos and 474 testing videos while preserving the original 7,218 and 1,550 text queries, respectively, yielding an average of 3.3 queries per video.

Charades [32] contains 6,670 videos with 16,128 sentence descriptions, where each video is associated with approximately 2.4 moment-text pairs on average. Compared to the other datasets, Charades contains relatively short and repetitive text queries with substantial semantic overlap across different videos. Specifically, the average query length is only 6.2 words, which is significantly shorter than TVR (12.2), ActivityNet Captions (13.6), and QVHighlights (10.5) [2].

Therefore, these datasets exhibit substantially different characteristics in terms of video domains, query styles, and semantic distributions. Consequently, conventional single-dataset evaluation settings are insufficient for assessing the generalization capability of PRVR methods [4, 7, 8, 28], as models optimized for a specific dataset often fail to maintain robust performance across other video domains.

Training Configuration For all components inherited from GMMFormer v2 [30], we use its default training configuration without modification, while excluding the optimal matching loss and the revamped query diverse loss. The Transformer encoders use hidden dimension 768, 4 attention heads, and dropout 0.2 on both inputs and residual sublayers. Consistent with prior PRVR methods, for videos exceeding 128 frames, we apply uniform binning with mean pooling within each bin to produce a fixed-length input. Unlike GMMFormer v2's 32-frame clip-level / 128-frame frame-level dual-branch design, our two video branches share the same 128-frame frame-level input; the dual-branch distinction comes only from separate query-attention paths (anchored vs adaptive).

Baseline Hyperparameters For each baseline method, we adopt the authors' published hyperparameters, modifying only the batch size to 256 for consistency with GAL. When a baseline provides dataset-specific configurations (e.g., separate configs for TVR, ActivityNet, Charades, QVHighlights), we uniformly use the TVR configuration across all four leave-one-out splits, as no per-target tuning is permitted under our cross-dataset generalization protocol.

Hyperparameter search. For GAL, hyperparameters are selected on the source-pooled validation set of a single leave-one-out split (ACT-as-target) and then fixed across all four splits without per-target adjustment. The unseen target of any split is never used in the selection of any hyperparameter. For GAL, we set $K = 0.4$ and $m = 0.05$ for CTRD, while $\tau_{\mathrm{psbs}}, \lambda_{\mathrm{anc}}, \lambda_{\mathrm{ctrd}}$ follow standard defaults and $\alpha = 0.6$; all hyperparameters are fixed across all four splits without per-target tuning.

Hardware and reproducibility. All experiments are run on a single NVIDIA RTX PRO 6000 Blackwell GPU (96 GB) with an AMD EPYC 9755 128-core CPU. We use three random seeds (9527, 1234, 2026) for all methods and report mean ± standard deviation across seeds.

C Detailed Results

Results across all recall cutoffs extending Tab. 1 To complement the main comparison in Tab. 1, we provide full per-Q recall results for each retrieval metric. Specifically, we report separate tables for R@1, R@5, R@10, R@100, and SumR in Tabs. 7-11. While the main table comparison focuses on unseen-target performance, these appendix tables additionally include the source-side metrics from the multi-source-to-unseen-target protocol in Sec. 4.1: Src-Wise Avg and Src-Pool Avg. This allows us to compare existing methods under the same aggregation protocol across both source and unseen-target evaluations. Overall, these full per-Q results are consistent with the main comparison, showing that GAL improves unseen-target retrieval while maintaining competitive source-side performance across cutoff levels.

Source-domain results across all recall cutoffs extending Tab. 2 We additionally report source-domain retrieval performance for each recall cutoff level in Tabs. 12-15. While the main paper summarizes source-side performance using SumR, these appendix tables provide a more fine-grained analysis across R@1, R@5, R@10, and R@100. The results show that GAL maintains competitive or superior source-domain retrieval performance consistently across different recall thresholds, while simultaneously achieving substantially stronger unseen-target retrieval performance. These observations further support that preserving CLIP-derived transferability does not necessarily require sacrificing source-domain retrieval quality.

Table 7: Multi-source-to-unseen-target retrieval performance comparison on held-out targets. We report R@1; Src-Wise Avg., Src-Pool Avg., and Unseen-target Avg. follow the protocol definition in Sec. 4.1. Values are means over three seeds; subscripted ± denotes standard deviation. Best in **bold**, second-best underlined.

Method	Source (R@1)		Unseen Target (R@1)				
	Src-Wise Avg.	Src-Pool Avg.	TVR	ACT	CHA	QVH	Avg.
Zero-shot CLIP	17.8	14.2	16.2	16.0	2.0	36.8	17.8
MS-SL	20.2	20.2	$5.2_{\pm 0.6}$	$7.1_{\pm 0.2}$	$2.0_{\pm 0.2}$	$22.3_{\pm 1.0}$	9.1
GMMFormer	18.9	19.4	$4.3_{\pm 0.2}$	$7.5_{\pm 0.3}$	$1.8_{\pm 0.1}$	$20.8_{\pm 1.5}$	8.6
DL-DKD	21.1	20.9	$6.2_{\pm 0.4}$	$9.1_{\pm 0.4}$	$2.3_{\pm 0.3}$	$25.9_{\pm 0.3}$	10.9
GMMFormer v2	21.4	21.5	$5.2_{\pm 0.2}$	$8.1_{\pm 0.3}$	$1.8_{\pm 0.2}$	$23.4_{\pm 0.4}$	9.6
HLFormer	21.5	21.4	$5.4_{\pm 0.2}$	$8.1_{\pm 0.1}$	$2.0_{\pm 0.1}$	$24.0_{\pm 0.2}$	9.9
MSC-PRVR	22.9	22.8	$6.0_{\pm 0.2}$	$9.4_{\pm 0.1}$	$2.1_{\pm 0.2}$	$27.3_{\pm 0.7}$	11.2
Ours	**25.5**	**23.3**	$\mathbf{17.9}_{\pm 0.1}$	$\mathbf{17.8}_{\pm 0.1}$	$\mathbf{2.7}_{\pm 0.1}$	$\mathbf{43.1}_{\pm 0.8}$	**20.4**

Table 8: Multi-source-to-unseen-target retrieval performance comparison on held-out targets. We report R@5; Src-Wise Avg., Src-Pool Avg., and Unseen-target Avg. follow the protocol definition in Sec. 4.1. Values are means over three seeds; subscripted ± denotes standard deviation. Best in **bold**, second-best underlined.

Method	Source (R@5)		Unseen Target (R@5)				
	Src-Wise Avg.	Src-Pool Avg.	TVR	ACT	CHA	QVH	Avg.
Zero-shot CLIP	34.8	30.3	33.5	35.1	8.1	62.4	34.8
MS-SL	41.4	42.3	$12.7_{\pm 0.8}$	$19.4_{\pm 0.2}$	$7.1_{\pm 0.2}$	$46.9_{\pm 1.8}$	21.5
GMMFormer	39.8	41.0	$10.6_{\pm 0.4}$	$20.4_{\pm 0.6}$	$6.9_{\pm 0.3}$	$44.4_{\pm 1.6}$	20.6
DL-DKD	42.2	42.8	$14.6_{\pm 1.0}$	$23.9_{\pm 0.2}$	$7.6_{\pm 0.2}$	$50.5_{\pm 0.5}$	24.1
GMMFormer v2	42.8	43.8	$12.4_{\pm 0.4}$	$21.3_{\pm 0.3}$	$6.4_{\pm 0.2}$	$49.0_{\pm 0.6}$	22.3
HLFormer	43.1	43.9	$13.2_{\pm 0.3}$	$21.7_{\pm 0.4}$	$7.2_{\pm 0.2}$	$49.2_{\pm 0.2}$	22.8
MSC-PRVR	44.6	**45.5**	$14.4_{\pm 0.4}$	$24.5_{\pm 0.3}$	$7.6_{\pm 0.1}$	$53.1_{\pm 0.4}$	24.9
Ours	**46.3**	45.2	$\mathbf{35.4}_{\pm 0.2}$	$\mathbf{38.2}_{\pm 0.3}$	$\mathbf{9.5}_{\pm 0.2}$	$\mathbf{68.5}_{\pm 0.6}$	**37.9**

Table 9: Multi-source-to-unseen-target retrieval performance comparison on held-out targets. We report R@10; Src-Wise Avg., Src-Pool Avg., and Unseen-target Avg. follow the protocol definition in Sec. 4.1. Values are means over three seeds; subscripted $\pm$ denotes standard deviation. Best in **bold**, second-best underlined.

Method	Source (R@10)		Unseen Target (R@10)				
	Src-Wise Avg.	Src-Pool Avg.	TVR	ACT	CHA	QVH	Avg.
Zero-shot CLIP	43.6	39.3	41.8	46.4	13.6	72.6	43.6
MS-SL	52.0	53.7	$17.7_{\pm 1.0}$	$28.3_{\pm 0.2}$	$11.9_{\pm 0.4}$	$58.7_{\pm 0.5}$	29.1
GMMFormer	50.3	52.2	$15.3_{\pm 0.3}$	$29.3_{\pm 0.5}$	$11.4_{\pm 0.1}$	$55.9_{\pm 1.7}$	28.0
DL-DKD	52.6	54.0	$19.8_{\pm 1.0}$	$33.5_{\pm 0.3}$	$12.3_{\pm 0.7}$	$61.3_{\pm 1.2}$	31.7
GMMFormer v2	53.5	55.2	$17.6_{\pm 0.5}$	$30.3_{\pm 0.4}$	$10.9_{\pm 0.5}$	$60.3_{\pm 0.6}$	29.8
HLFormer	53.5	55.1	$18.3_{\pm 0.2}$	$30.8_{\pm 0.7}$	$11.8_{\pm 0.5}$	$60.8_{\pm 0.3}$	30.4
MSC-PRVR	55.1	**56.7**	$20.0_{\pm 0.3}$	$34.4_{\pm 0.3}$	$12.6_{\pm 0.4}$	$64.5_{\pm 0.4}$	32.9
Ours	**55.8**	55.8	$\mathbf{44.2}_{\pm 0.1}$	$\mathbf{50.1}_{\pm 0.1}$	$\mathbf{15.6}_{\pm 0.4}$	$\mathbf{77.8}_{\pm 0.3}$	**46.9**

Table 10: Multi-source-to-unseen-target retrieval performance comparison on held-out targets. We report R@100; Src-Wise Avg., Src-Pool Avg., and Unseen-target Avg. follow the protocol definition in Sec. 4.1. Values are means over three seeds; subscripted $\pm$ denotes standard deviation. Best in **bold**, second-best underlined.

Method	Source (R@100)		Unseen Target (R@100)				
	Src-Wise Avg.	Src-Pool Avg.	TVR	ACT	CHA	QVH	Avg.
Zero-shot CLIP	75.1	72.0	75.7	80.0	49.3	95.3	75.1
MS-SL	84.7	85.6	$45.8_{\pm 1.6}$	$64.9_{\pm 0.6}$	$47.9_{\pm 1.1}$	$92.1_{\pm 0.6}$	62.7
GMMFormer	83.2	84.3	$41.7_{\pm 0.5}$	$65.9_{\pm 0.3}$	$46.2_{\pm 0.9}$	$91.6_{\pm 0.2}$	61.3
DL-DKD	84.0	85.2	$48.7_{\pm 2.7}$	$69.7_{\pm 0.3}$	$46.7_{\pm 0.8}$	$93.1_{\pm 0.5}$	64.6
GMMFormer v2	84.8	85.9	$45.7_{\pm 1.0}$	$66.6_{\pm 0.8}$	$46.0_{\pm 1.0}$	$92.6_{\pm 0.8}$	62.7
HLFormer	84.7	85.8	$46.7_{\pm 1.0}$	$67.7_{\pm 0.7}$	$46.7_{\pm 0.4}$	$92.7_{\pm 0.1}$	63.4
MSC-PRVR	**85.5**	**86.6**	$49.6_{\pm 0.4}$	$70.5_{\pm 0.5}$	$49.6_{\pm 1.0}$	$94.0_{\pm 0.3}$	65.9
Ours	84.9	86.0	$\mathbf{78.0}_{\pm 0.3}$	$\mathbf{82.7}_{\pm 0.1}$	$\mathbf{54.4}_{\pm 0.3}$	$\mathbf{97.4}_{\pm 0.2}$	**78.1**

Table 11: Multi-source-to-unseen-target retrieval performance comparison. We report SumR, the sum of R@$\{1, 5, 10, 100\}$; Src-Wise Avg., Src-Pool Avg., and Unseen-target Avg. follow the protocol definition in Sec. 4.1. Values are means over three seeds; subscripted $\pm$ denotes standard deviation. Best in **bold**, second-best underlined.

Method	Source (SumR)		Unseen Target (SumR)				
	Src-Wise Avg.	Src-Pool Avg.	TVR	ACT	CHA	QVH	Avg.
Zero-shot CLIP	171.2	155.8	167.2	177.5	73.1	267.0	171.2
MS-SL	198.3	201.8	$81.4_{\pm 4.0}$	$119.6_{\pm 0.6}$	$68.9_{\pm 1.5}$	$220.0_{\pm 3.8}$	122.5
GMMFormer	192.2	196.9	$71.9_{\pm 0.6}$	$123.1_{\pm 1.6}$	$66.3_{\pm 0.8}$	$212.7_{\pm 4.0}$	118.5
DL-DKD	199.9	202.9	$89.4_{\pm 5.1}$	$136.2_{\pm 0.6}$	$69.0_{\pm 1.5}$	$230.8_{\pm 2.2}$	131.4
GMMFormer v2	202.4	206.6	$80.9_{\pm 2.0}$	$126.2_{\pm 1.6}$	$65.2_{\pm 0.4}$	$225.3_{\pm 1.1}$	124.4
HLFormer	202.7	206.3	$83.6_{\pm 1.3}$	$128.3_{\pm 1.8}$	$67.6_{\pm 1.1}$	$226.6_{\pm 0.2}$	126.5
MSC-PRVR	208.1	**211.6**	$89.9_{\pm 0.6}$	$138.9_{\pm 1.0}$	$71.9_{\pm 1.4}$	$239.0_{\pm 0.8}$	134.9
Ours	**212.5**	210.2	$\mathbf{175.4}_{\pm 0.4}$	$\mathbf{188.7}_{\pm 0.3}$	$\mathbf{82.1}_{\pm 0.4}$	$\mathbf{286.7}_{\pm 0.2}$	**183.2**

Table 12: Source-domain retrieval performance comparison. Recall@1 (R@1) is reported for each dataset when it is used as a source domain. Each dataset-level source score is averaged over the three splits where the dataset is used as a source and three random seeds; subscripted $\pm$ denotes standard deviation. Src-Wise Avg. averages the four dataset-level source scores, and Src-Pool Avg. follows the protocol definition in Sec. 4.1. Best in **bold**, second-best underlined.

Source	ZS CLIP	MS-SL	GMMFormer	DL-DKD	GMMF. v2	HLFormer	MSC-PRVR	Ours
TVR	16.2	$31.1_{\pm 0.4}$	$29.8_{\pm 0.5}$	$31.6_{\pm 0.5}$	$33.5_{\pm 0.6}$	$33.5_{\pm 0.8}$	**34.9**$_{\pm 0.8}$	$31.5_{\pm 0.4}$
ACT	16.0	$16.3_{\pm 0.5}$	$16.1_{\pm 0.7}$	$17.4_{\pm 0.8}$	$17.1_{\pm 0.8}$	$17.1_{\pm 0.8}$	$18.6_{\pm 0.8}$	**21.2**$_{\pm 0.3}$
CHA	2.0	$3.5_{\pm 0.2}$	$3.1_{\pm 0.5}$	$3.2_{\pm 0.3}$	$3.6_{\pm 0.3}$	$3.4_{\pm 0.4}$	**3.7**$_{\pm 0.3}$	$3.5_{\pm 0.2}$
QVH	36.8	$29.8_{\pm 2.7}$	$26.5_{\pm 3.0}$	$32.2_{\pm 2.4}$	$31.2_{\pm 2.6}$	$31.9_{\pm 2.3}$	$34.5_{\pm 2.2}$	**45.9**$_{\pm 0.7}$
Src-Wise Avg.	17.8	20.2	18.9	21.1	21.4	21.5	22.9	**25.5**
Src-Pool Avg.	14.2	20.2	19.4	20.9	21.5	21.4	22.8	**23.3**

Table 13: Source-domain retrieval performance comparison. Recall@5 (R@5) is reported for each dataset when it is used as a source domain. Each dataset-level source score is averaged over the three splits where the dataset is used as a source and three random seeds; subscripted $\pm$ denotes standard deviation. Src-Wise Avg. averages the four dataset-level source scores, and Src-Pool Avg. follows the protocol definition in Sec. 4.1. Best in **bold**, second-best underlined.

Source	ZS CLIP	MS-SL	GMMFormer	DL-DKD	GMMF. v2	HLFormer	MSC-PRVR	Ours
TVR	33.5	$56.8_{\pm 0.4}$	$54.9_{\pm 0.6}$	$56.6_{\pm 0.6}$	$59.5_{\pm 0.7}$	$59.3_{\pm 0.6}$	**60.6**$_{\pm 0.5}$	$56.2_{\pm 0.5}$
ACT	35.1	$39.2_{\pm 0.7}$	$38.6_{\pm 0.9}$	$40.6_{\pm 1.0}$	$40.3_{\pm 0.9}$	$40.5_{\pm 1.0}$	$42.5_{\pm 1.0}$	**44.3**$_{\pm 0.6}$
CHA	8.1	$12.0_{\pm 0.3}$	$11.1_{\pm 0.7}$	$11.1_{\pm 0.8}$	$12.2_{\pm 0.7}$	$12.2_{\pm 0.7}$	**12.8**$_{\pm 0.5}$	$11.9_{\pm 0.3}$
QVH	62.4	$57.6_{\pm 2.7}$	$54.7_{\pm 2.9}$	$60.5_{\pm 1.9}$	$59.3_{\pm 2.5}$	$60.3_{\pm 2.0}$	$62.6_{\pm 2.2}$	**72.8**$_{\pm 0.6}$
Src-Wise Avg.	34.8	41.4	39.8	42.2	42.8	43.1	44.6	**46.3**
Src-Pool Avg.	30.3	42.3	41.0	42.8	43.8	43.9	**45.5**	45.2

Table 14: Source-domain retrieval performance comparison. Recall@10 (R@10) is reported for each dataset when it is used as a source domain. Each dataset-level source score is averaged over the three splits where the dataset is used as a source and three random seeds; subscripted $\pm$ denotes standard deviation. Src-Wise Avg. averages the four dataset-level source scores, and Src-Pool Avg. follows the protocol definition in Sec. 4.1. Best in **bold**, second-best underlined.

Source	ZS CLIP	MS-SL	GMMFormer	DL-DKD	GMMF. v2	HLFormer	MSC-PRVR	Ours
TVR	41.8	$67.3_{\pm 0.4}$	$65.6_{\pm 0.5}$	$67.0_{\pm 0.5}$	$69.9_{\pm 0.5}$	$69.6_{\pm 0.5}$	**70.7**$_{\pm 0.4}$	$66.3_{\pm 0.5}$
ACT	46.4	$52.5_{\pm 0.6}$	$51.7_{\pm 0.7}$	$53.7_{\pm 0.8}$	$53.7_{\pm 0.7}$	$53.7_{\pm 0.9}$	$55.7_{\pm 0.8}$	**56.8**$_{\pm 0.6}$
CHA	13.6	$19.4_{\pm 0.6}$	$17.8_{\pm 0.6}$	$17.9_{\pm 0.9}$	$19.5_{\pm 0.7}$	$19.4_{\pm 0.8}$	**20.3**$_{\pm 0.8}$	$18.8_{\pm 0.4}$
QVH	72.6	$68.8_{\pm 2.5}$	$66.2_{\pm 2.2}$	$71.7_{\pm 1.7}$	$70.9_{\pm 1.9}$	$71.3_{\pm 1.6}$	$73.7_{\pm 2.0}$	**81.4**$_{\pm 0.8}$
Src-Wise Avg.	43.6	52.0	50.3	52.6	53.5	53.5	55.1	**55.8**
Src-Pool Avg.	39.3	53.7	52.2	54.0	55.2	55.1	**56.7**	55.8

Table 15: Source-domain retrieval performance comparison. Recall@100 (R@100) is reported for each dataset when it is used as a source domain. Each dataset-level source score is averaged over the three splits where the dataset is used as a source and three random seeds; subscripted $\pm$ denotes standard deviation. Src-Wise Avg. averages the four dataset-level source scores, and Src-Pool Avg. follows the protocol definition in Sec. 4.1. Best in **bold**, second-best underlined.

Source	ZS CLIP	MS-SL	GMMFormer	DL-DKD	GMMF. v2	HLFormer	MSC-PRVR	Ours
TVR	75.7	$93.7_{\pm 0.1}$	$93.3_{\pm 0.1}$	$93.5_{\pm 0.1}$	$94.7_{\pm 0.2}$	$94.6_{\pm 0.1}$	**95.0**$_{\pm 0.1}$	$93.3_{\pm 0.2}$
ACT	80.0	$85.7_{\pm 0.3}$	$84.6_{\pm 0.4}$	$85.8_{\pm 0.2}$	$85.8_{\pm 0.2}$	$85.8_{\pm 0.2}$	$86.8_{\pm 0.3}$	**87.3**$_{\pm 0.2}$
CHA	49.3	$63.1_{\pm 0.5}$	$60.1_{\pm 1.3}$	$60.1_{\pm 0.7}$	$62.3_{\pm 0.7}$	$62.4_{\pm 0.5}$	**63.3**$_{\pm 0.5}$	$60.7_{\pm 0.4}$
QVH	95.3	$96.4_{\pm 0.5}$	$94.9_{\pm 0.8}$	$96.7_{\pm 0.5}$	$96.2_{\pm 0.4}$	$96.1_{\pm 0.4}$	$96.8_{\pm 0.3}$	**98.2**$_{\pm 0.2}$
Src-Wise Avg.	75.1	84.7	83.2	84.0	84.8	84.7	**85.5**	84.9
Src-Pool Avg.	72.0	85.6	84.3	85.2	85.9	85.8	**86.6**	86.0

D Relation to Video Domain Adaptation and Generalization

Our multi-source-to-unseen-target protocol is related to, but distinct from, video domain adaptation (DA) and video domain generalization (DG). Video DA methods [42, 43] typically access unlabeled target-domain videos during training, which is incompatible with our strictly target-free protocol. Video DG [44] is closer to our setting, since it also evaluates on unseen domains; however, representative video DG methods are formulated as closed-set classification with a fixed label space shared across source and target domains. PRVR has no such shared label space: relevance is defined by an arbitrary free-form query, and only an unknown temporal portion of a video may match it. Moreover, PRVR benchmarks differ in query vocabulary, semantic granularity, and annotation style. Consequently, class-discriminative or domain-invariant objectives cannot be directly transferred to PRVR without reformulation.

Our focus also differs from conventional visual domain shift. Frozen CLIP already transfers well across PRVR benchmarks (Tab. 1), whereas PRVR-specific training can weaken this transferable text–visual alignment while learning finer segment-level matching. GAL addresses this trade-off by preserving general CLIP semantics in one branch while adapting to query-conditioned segment matching in the other. Although this principle may extend to other tasks, the scoring mechanism and branch-specific objectives of GAL are tailored to PRVR.

E Scaling the Number of Source Datasets

Table 16: Unseen-target SumR with respect to the number of training sources, averaged over all source combinations of each size. Single seed (9527).

Method	# Sources	TVR	ACT	CHA	QVH	Avg.
Zero-shot CLIP	–	167.2	177.5	73.1	267.0	171.2
GMMFormer v2	1	33.1	44.0	38.8	146.7	65.7
	2	64.4	103.7	64.0	196.6	107.2
	3	81.8	126.7	65.6	224.7	124.7
GAL	1	172.2	185.1	77.8	279.8	178.7
	2	174.3	188.0	79.9	283.4	181.4
	3	175.7	188.9	82.3	286.8	183.4

We evaluate how unseen-target performance changes as the number of source datasets increases from one to three. For each held-out target and each source count, we average SumR over all source combinations of that size. For example, when TVR is held out, the one-source result averages ACT, CHA, and QVH; the two-source result averages ACT+CHA, ACT+QVH, and CHA+QVH; and the three-source result uses ACT+CHA+QVH. Due to the combinatorial number of configurations, we use a single seed (9527) and compare with GMMFormer v2. As shown in Tab. 16, GAL improves monotonically with the number of sources on all four unseen targets and already outperforms zero-shot CLIP on every target with a single source. GMMFormer v2 also benefits from broader source coverage but remains below zero-shot CLIP on every target even with all three sources. These results show that GAL preserves strong transferability under limited source supervision and further benefits from additional heterogeneous training data.

F Preservation of the CLIP Neighborhood Structure

Table 17: Preservation of the frozen-CLIP neighborhood structure (kNN∩CLIP@10, %; higher means more preserved) on the ActivityNet-held-out split.

Representation	GAL (anchored)	GAL (adaptive)	GMMFormer v2	MSC-PRVR
Query	94.0	49.6	40.8	48.3
GT-moment segment	91.5	72.4	52.8	53.5

To examine how training changes the learned representation space, we measure how well each model preserves the neighborhood structure of frozen CLIP. Using the ActivityNet-held-out split, we analyze query features and the features of annotated ground-truth moments, where each moment feature is obtained by mean-pooling its frame features. The moment annotations are not used during training or retrieval; they are used only in this post-hoc analysis to isolate the query-relevant portion of each video. For each sample, we retrieve its 10 nearest neighbors by cosine similarity both in the frozen CLIP space and in a learned representation space, and compute the fraction of neighbors shared by the two sets (kNN∩CLIP@10), averaged over all samples. For example, 94.0% means that, on average, 9.4 of the 10 frozen-CLIP neighbors remain neighbors after training. As shown in Tab. 17, the anchored branch preserves most of the CLIP neighborhood structure, whereas GMMFormer v2 and MSC-PRVR substantially alter it. The adaptive branch shows intermediate preservation, consistent with the intended separation between CLIP preservation and PRVR-specific adaptation.

G Qualitative Analysis

Table 18: Representative source-domain failure cases. Each entry is the rank of the ground-truth video (lower is better).

				GAL		
Dataset	**Query**	**ZS CLIP**	**GMMv2**	**Anch.**	**Adapt.**	**Fused**
TVR	Beckett nods after Ryan tells her what she is going to do.	400	8	347	150	221
TVR	Dr. Cameron shows Dr. Forman the repeated x ray.	794	10	463	71	135
CHA	person they stand up	1199	7	761	636	680
CHA	the person sits down in a chair	741	3	636	231	351

Source-domain failure cases. Although GAL achieves the best Src-Wise Avg. and a competitive Src-Pool Avg., it trails stronger source-fitted baselines on TVR and Charades (Tab. 2). We inspect queries for which GMMFormer v2 ranks the ground-truth video within the top 10 but GAL fails, and report representative cases in Tab. 18. The TVR cases involve show-specific entities and narrative relations, where identifying the target video depends not only on the depicted action but also on character identities and interaction context that recur within the benchmark. The poor zero-shot CLIP ranks, together with the strong GMMFormer v2 ranks, suggest that source fitting can effectively exploit these recurring benchmark-specific cues. The Charades cases show a different pattern: short, low-specificity action queries such as "person they stand up" can plausibly match many videos depicting the same coarse action, and stronger source fitting may capture subtle temporal or contextual regularities that distinguish the labeled video from otherwise similar candidates. These cases refine our limitation: preserving transferable alignment improves overall and unseen-target performance, but may be less effective for queries that depend heavily on benchmark-specific entities or fine-grained in-domain distinctions.

Table 19: Representative training queries grouped by their EMA anchored-branch weight w^{anc}.

Route	$\boldsymbol{w^{\mathrm{anc}}}$	**Query**
Anchored-favored	0.825	People are watching him next to a pond.
Anchored-favored	0.780	Words are then shown on the screen.
Adaptive-favored	0.085	The man serves the ball and scores a point.
Adaptive-favored	0.174	More women are shown running down the track and landing into the sand pit as others watch.
Balanced	0.468	They sit at the table and hold each other's hands, prepared to arm wrestle.
Balanced	0.492	He then uses a brush to polish the suede material before spraying a white substance onto the shoe.

Per-sample branch routing. To complement Fig. 3a, we qualitatively inspect training queries according to their EMA anchored-branch weight w^{anc}: high values favor the anchored branch, low values favor the adaptive branch, and values near 0.5 indicate balanced supervision. As shown

in Tab. 19, anchored-favored queries tend to describe visually salient states or scenes that can be recognized from representative frames, whereas adaptive-favored queries more often involve fine-grained motion or temporal interactions. Balanced queries combine recognizable visual content with action-level information. PSBS can thus be viewed as inducing a soft, sample-dependent assignment between the two branches; rather than learning an explicit discrete cluster, the assignment arises continuously from the relative per-sample retrieval losses.

H Per-Method Breakdown

To provide a detailed analysis of each method under the multi-source-to-unseen-target protocol, we report per-dataset breakdowns for all methods (see Tabs. 20–26), providing per-src and pooled-src breakdowns alongside the unseen-target results.

For each unseen target, we evaluate retrieval performance under three roles: (i) *per-src*, where retrieval is performed separately within each source dataset; (ii) *pooled-src*, where queries and videos from all source datasets are merged into a single retrieval pool; and (iii) *unseen*, where the held-out target dataset is evaluated without being seen during training. We report Recall at $Q \in \{1, 5, 10, 100\}$ and their sum (SumR). All results are averaged over three random seeds, with the standard deviation computed across seeds.

Table 20: Per-domain breakdown of GMMFormer v2 under the real-world (multi-source to unseen) setting. We report Recall at $Q \in \{1, 5, 10, 100\}$ and SumR across per-source, pooled-source, and unseen domains.

Target	Domain	Role	R@1	R@5	R@10	R@100	SumR
TVR	ACT	per-src	16.6±0.6	39.7±0.3	53.2±0.1	85.6±0.1	195.1±1.0
TVR	CHA	per-src	3.3±0.2	11.8±0.6	18.9±0.4	61.8±0.6	95.9±1.3
TVR	QVH	per-src	30.6±0.5	59.4±0.9	71.1±0.9	96.6±0.0	257.7±1.4
TVR	combined	pooled-src	14.7±0.5	35.2±0.1	47.5±0.2	81.5±0.1	178.9±0.6
TVR	TVR	unseen	5.2±0.2	12.4±0.4	17.6±0.5	45.7±1.0	80.9±2.0
ACT	TVR	per-src	33.6±0.7	59.6±0.8	69.7±0.5	94.7±0.3	257.7±2.3
ACT	CHA	per-src	3.7±0.3	12.0±0.6	19.7±0.5	62.5±1.0	97.8±1.9
ACT	QVH	per-src	28.7±1.1	56.5±0.5	68.7±0.5	95.8±0.3	249.7±1.4
ACT	combined	pooled-src	26.1±0.5	48.1±0.5	57.9±0.3	87.1±0.1	219.2±1.1
ACT	ACT	unseen	8.1±0.3	21.3±0.3	30.3±0.4	66.6±0.8	126.2±1.6
CHA	TVR	per-src	33.9±0.2	60.0±0.4	70.3±0.2	94.7±0.0	258.9±0.4
CHA	ACT	per-src	18.1±0.1	41.5±0.2	54.5±0.2	86.1±0.3	200.1±0.6
CHA	QVH	per-src	34.4±0.1	61.9±1.1	72.9±0.5	96.1±0.4	265.4±0.9
CHA	combined	pooled-src	24.5±0.2	48.9±0.2	60.8±0.0	89.3±0.1	223.5±0.5
CHA	CHA	unseen	1.8±0.2	6.4±0.2	10.9±0.5	46.0±1.0	65.2±0.4
QVH	TVR	per-src	33.1±0.7	59.0±0.5	69.6±0.5	94.7±0.2	256.3±1.7
QVH	ACT	per-src	16.8±0.3	39.9±0.2	53.2±0.5	85.8±0.1	195.7±0.8
QVH	CHA	per-src	3.7±0.3	12.8±0.5	20.1±0.7	62.7±0.2	99.1±1.6
QVH	combined	pooled-src	20.9±0.1	43.2±0.1	54.7±0.2	85.8±0.1	204.6±0.3
QVH	QVH	unseen	23.4±0.4	49.0±0.6	60.3±0.6	92.6±0.8	225.3±1.1

Table 21: Per-domain breakdown of GMMFormer under the real-world (multi-source to unseen) setting. We report Recall at $Q \in \{1, 5, 10, 100\}$ and SumR across per-source, pooled-source, and unseen domains.

Target	Domain	Role	R@1	R@5	R@10	R@100	SumR
TVR	ACT	per-src	15.5±0.3	38.0±0.3	51.2±0.1	84.2±0.1	188.9±0.2
TVR	CHA	per-src	2.7±0.2	10.3±0.6	17.2±0.3	58.8±0.3	89.0±1.0
TVR	QVH	per-src	26.8±0.8	54.5±1.6	66.4±1.4	95.3±0.5	243.0±4.1
TVR	combined	pooled-src	13.5±0.3	33.3±0.2	45.2±0.1	79.6±0.2	171.6±0.6
TVR	TVR	unseen	4.3±0.2	10.6±0.4	15.3±0.3	41.7±0.5	71.9±0.6
ACT	TVR	per-src	29.5±0.4	55.0±0.1	65.7±0.1	93.4±0.2	243.6±0.4
ACT	CHA	per-src	3.2±0.5	11.2±0.2	17.8±0.4	60.0±0.8	92.2±1.2
ACT	QVH	per-src	23.1±1.3	51.6±0.5	63.7±0.5	93.9±0.3	232.2±1.5
ACT	combined	pooled-src	22.8±0.4	44.4±0.1	54.2±0.0	85.4±0.2	206.7±0.3
ACT	ACT	unseen	7.5±0.3	20.4±0.6	29.3±0.5	65.9±0.3	123.1±1.6
CHA	TVR	per-src	30.3±0.3	55.3±0.6	66.0±0.4	93.3±0.1	245.0±1.2
CHA	ACT	per-src	17.0±0.3	39.7±0.2	52.6±0.3	85.0±0.2	194.3±0.7
CHA	QVH	per-src	29.7±0.7	57.9±0.8	68.5±0.3	95.4±0.2	251.5±1.7
CHA	combined	pooled-src	22.3±0.1	45.8±0.2	57.6±0.1	88.0±0.1	213.7±0.2
CHA	CHA	unseen	1.8±0.1	6.9±0.3	11.4±0.1	46.2±0.9	66.3±0.8
QVH	TVR	per-src	29.4±0.1	54.4±0.5	65.0±0.4	93.2±0.1	242.1±0.9
QVH	ACT	per-src	15.7±0.1	38.0±0.1	51.2±0.1	84.6±0.2	189.6±0.1
QVH	CHA	per-src	3.5±0.3	11.8±0.2	18.6±0.1	61.4±1.0	95.2±0.7
QVH	combined	pooled-src	19.0±0.0	40.4±0.2	51.8±0.2	84.3±0.1	195.6±0.4
QVH	QVH	unseen	20.8±1.5	44.4±1.6	55.9±1.7	91.6±0.2	212.7±4.0

Table 22: Per-domain breakdown of HLFormer under the real-world (multi-source to unseen) setting. We report Recall at $Q \in \{1, 5, 10, 100\}$ and SumR across per-source, pooled-source, and unseen domains.

Target	Domain	Role	R@1	R@5	R@10	R@100	SumR
TVR	ACT	per-src	16.5±0.1	39.9±0.2	53.1±0.4	85.7±0.2	195.3±0.5
TVR	CHA	per-src	3.1±0.3	11.7±0.5	18.8±0.8	61.8±0.5	95.4±1.1
TVR	QVH	per-src	31.7±0.7	60.3±0.6	71.1±0.9	96.5±0.2	259.6±1.6
TVR	combined	pooled-src	14.5±0.0	35.5±0.2	47.5±0.4	81.6±0.2	179.1±0.4
TVR	TVR	unseen	5.4±0.2	13.2±0.3	18.3±0.2	46.7±1.0	83.6±1.3
ACT	TVR	per-src	33.2±0.9	59.1±0.4	69.6±0.4	94.5±0.1	256.4±1.5
ACT	CHA	per-src	3.5±0.1	11.9±0.1	19.0±0.2	62.8±0.2	97.1±0.4
ACT	QVH	per-src	29.4±1.4	58.0±0.6	69.7±0.9	95.8±0.3	252.9±1.8
ACT	combined	pooled-src	25.9±0.7	48.0±0.3	57.7±0.3	87.0±0.1	218.4±1.1
ACT	ACT	unseen	8.1±0.1	21.7±0.4	30.8±0.7	67.7±0.7	128.3±1.8
CHA	TVR	per-src	34.3±0.4	59.9±0.4	70.2±0.3	94.6±0.1	259.0±0.9
CHA	ACT	per-src	18.1±0.1	41.7±0.2	54.8±0.4	86.1±0.1	200.7±0.5
CHA	QVH	per-src	34.5±0.3	62.6±0.5	73.1±0.1	96.1±0.2	266.4±0.6
CHA	combined	pooled-src	24.6±0.2	49.0±0.3	60.8±0.2	89.2±0.1	223.7±0.3
CHA	CHA	unseen	2.0±0.1	7.2±0.2	11.8±0.5	46.7±0.4	67.6±1.1
QVH	TVR	per-src	33.1±0.2	58.8±0.4	69.1±0.3	94.6±0.1	255.6±0.9
QVH	ACT	per-src	16.7±0.1	39.8±0.1	53.2±0.3	85.6±0.1	195.2±0.1
QVH	CHA	per-src	3.6±0.5	12.9±0.4	20.3±0.3	62.5±0.2	99.4±1.1
QVH	combined	pooled-src	20.8±0.0	43.0±0.1	54.6±0.3	85.5±0.1	203.9±0.3
QVH	QVH	unseen	24.0±0.2	49.2±0.2	60.8±0.3	92.7±0.1	226.6±0.2

Table 23: Per-domain breakdown of MS-SL under the real-world (multi-source to unseen) setting. We report Recall at $Q \in \{1, 5, 10, 100\}$ and SumR across per-source, pooled-source, and unseen domains.

Target	Domain	Role	R@1	R@5	R@10	R@100	SumR
TVR	ACT	per-src	16.2±0.2	39.3±0.4	52.6±0.4	85.5±0.2	193.7±1.0
TVR	CHA	per-src	3.5±0.3	11.9±0.4	19.2±0.7	62.6±0.4	97.0±1.5
TVR	QVH	per-src	30.1±0.6	58.5±0.4	69.5±0.2	96.5±0.2	254.6±0.9
TVR	combined	pooled-src	14.4±0.2	34.9±0.4	47.1±0.4	81.6±0.3	178.0±1.0
TVR	TVR	unseen	5.2±0.6	12.7±0.8	17.7±1.0	45.8±1.6	81.4±4.0
ACT	TVR	per-src	31.2±0.3	57.1±0.3	67.6±0.4	93.7±0.2	249.6±0.7
ACT	CHA	per-src	3.4±0.1	11.9±0.2	19.0±0.4	63.2±0.2	97.6±0.6
ACT	QVH	per-src	26.6±0.6	54.1±0.4	65.8±0.5	95.9±0.4	242.4±1.1
ACT	combined	pooled-src	24.2±0.2	46.2±0.2	56.0±0.3	86.5±0.2	212.9±0.4
ACT	ACT	unseen	7.1±0.2	19.4±0.2	28.3±0.2	64.9±0.6	119.6±0.6
CHA	TVR	per-src	31.4±0.3	56.9±0.1	67.4±0.2	93.6±0.1	249.4±0.4
CHA	ACT	per-src	16.8±0.0	39.9±0.2	53.1±0.2	86.1±0.2	195.9±0.2
CHA	QVH	per-src	32.7±0.7	60.1±1.0	71.1±1.1	96.8±0.2	260.7±2.4
CHA	combined	pooled-src	22.7±0.2	46.7±0.1	58.7±0.2	88.8±0.1	217.1±0.4
CHA	CHA	unseen	2.0±0.2	7.1±0.2	11.9±0.4	47.9±1.1	68.9±1.5
QVH	TVR	per-src	30.8±0.4	56.3±0.3	66.9±0.3	93.7±0.1	247.6±0.8
QVH	ACT	per-src	15.9±0.5	38.4±0.3	51.9±0.3	85.4±0.1	191.5±0.8
QVH	CHA	per-src	3.6±0.2	12.3±0.3	19.9±0.3	63.5±0.3	99.4±0.6
QVH	combined	pooled-src	19.6±0.4	41.4±0.0	53.1±0.2	85.3±0.1	199.3±0.6
QVH	QVH	unseen	22.3±1.0	46.9±1.8	58.7±0.5	92.1±0.6	220.0±3.8

Table 24: Per-domain breakdown of MSC-PRVR under the real-world (multi-source to unseen) setting. We report Recall at $Q \in \{1, 5, 10, 100\}$ and SumR across per-source, pooled-source, and unseen domains.

Target	Domain	Role	R@1	R@5	R@10	R@100	SumR
TVR	ACT	per-src	17.8±0.2	41.7±0.1	55.0±0.3	86.5±0.3	200.9±0.3
TVR	CHA	per-src	3.6±0.3	12.4±0.1	19.6±0.1	62.8±0.3	98.5±0.3
TVR	QVH	per-src	34.3±0.5	62.2±0.1	73.9±0.6	96.8±0.0	267.2±1.0
TVR	combined	pooled-src	15.9±0.1	37.1±0.1	49.2±0.2	82.3±0.2	184.5±0.2
TVR	TVR	unseen	6.0±0.2	14.4±0.4	20.0±0.3	49.6±0.4	89.9±0.6
ACT	TVR	per-src	34.6±1.0	60.4±0.5	70.7±0.4	95.0±0.0	260.8±1.9
ACT	CHA	per-src	3.6±0.3	12.5±0.3	19.9±0.2	63.5±0.4	99.4±0.6
ACT	QVH	per-src	32.2±0.3	60.2±0.7	71.5±0.5	96.5±0.4	260.4±0.9
ACT	combined	pooled-src	27.1±0.7	49.1±0.3	58.8±0.2	87.6±0.1	222.7±1.1
ACT	ACT	unseen	9.4±0.1	24.5±0.3	34.4±0.3	70.5±0.5	138.9±1.0
CHA	TVR	per-src	35.4±0.7	61.1±0.3	71.1±0.2	95.1±0.2	262.7±1.3
CHA	ACT	per-src	19.5±0.1	43.8±0.1	56.8±0.4	87.2±0.1	207.3±0.5
CHA	QVH	per-src	37.0±1.2	65.2±0.4	75.7±1.0	97.1±0.2	275.2±2.0
CHA	combined	pooled-src	26.0±0.3	50.8±0.2	62.5±0.1	90.2±0.1	229.5±0.5
CHA	CHA	unseen	2.1±0.2	7.6±0.1	12.6±0.4	49.6±1.0	71.9±1.4
QVH	TVR	per-src	34.7±0.5	60.3±0.4	70.4±0.3	95.0±0.2	260.3±1.3
QVH	ACT	per-src	18.5±0.3	42.0±0.1	55.4±0.1	86.7±0.1	202.5±0.5
QVH	CHA	per-src	3.9±0.1	13.4±0.2	21.3±0.4	63.6±0.5	102.2±0.8
QVH	combined	pooled-src	22.4±0.1	44.8±0.1	56.3±0.0	86.4±0.1	209.9±0.3
QVH	QVH	unseen	27.3±0.7	53.1±0.4	64.5±0.4	94.0±0.3	239.0±0.8

Table 25: Per-domain breakdown of DL-DKD under the real-world (multi-source to unseen) setting. We report Recall at $Q \in \{1, 5, 10, 100\}$ and SumR across per-source, pooled-source, and unseen domains.

Target	Domain	Role	R@1	R@5	R@10	R@100	SumR
TVR	ACT	per-src	16.4±0.3	39.9±0.3	52.9±0.3	85.8±0.1	195.1±0.5
TVR	CHA	per-src	2.9±0.3	10.4±0.6	17.4±0.5	60.3±0.3	90.9±1.5
TVR	QVH	per-src	30.8±0.7	59.9±0.1	71.6±0.4	96.9±0.3	259.2±0.8
TVR	combined	pooled-src	14.4±0.2	35.1±0.3	46.9±0.3	81.3±0.2	177.6±0.4
TVR	TVR	unseen	6.2±0.4	14.6±1.0	19.8±1.0	48.7±2.7	89.4±5.1
ACT	TVR	per-src	31.7±0.3	57.0±0.3	67.4±0.3	93.6±0.1	249.6±0.8
ACT	CHA	per-src	3.3±0.2	11.0±0.5	17.6±0.9	59.5±0.7	91.3±0.7
ACT	QVH	per-src	30.5±0.8	58.8±0.6	69.9±0.9	96.1±0.5	255.2±1.7
ACT	combined	pooled-src	24.9±0.2	46.3±0.2	55.9±0.3	85.7±0.2	212.8±0.6
ACT	ACT	unseen	9.1±0.4	23.9±0.2	33.5±0.3	69.7±0.3	136.2±0.6
CHA	TVR	per-src	32.0±0.3	56.8±0.6	67.2±0.4	93.5±0.1	249.5±1.1
CHA	ACT	per-src	18.3±0.2	41.9±0.1	54.6±0.2	86.0±0.1	200.8±0.6
CHA	QVH	per-src	35.3±1.1	62.9±0.5	73.5±0.8	97.0±0.2	268.7±2.4
CHA	combined	pooled-src	23.8±0.1	47.9±0.3	59.6±0.1	88.8±0.2	220.1±0.2
CHA	CHA	unseen	2.3±0.3	7.6±0.2	12.3±0.7	46.7±0.8	69.0±1.5
QVH	TVR	per-src	31.1±0.2	56.0±0.7	66.5±0.4	93.5±0.1	247.2±1.3
QVH	ACT	per-src	17.4±0.5	40.1±0.2	53.4±0.1	85.7±0.3	196.6±0.7
QVH	CHA	per-src	3.4±0.1	11.9±0.3	18.9±0.7	60.5±0.6	94.8±1.3
QVH	combined	pooled-src	20.4±0.3	42.1±0.1	53.5±0.1	85.0±0.2	201.0±0.4
QVH	QVH	unseen	25.9±0.3	50.5±0.5	61.3±1.2	93.1±0.5	230.8±2.2

Table 26: Per-domain breakdown of our method under the real-world (multi-source to unseen) setting. We report Recall at $Q \in \{1, 5, 10, 100\}$ and SumR across per-source, pooled-source, and unseen domains.

Target	Domain	Role	R@1	R@5	R@10	R@100	SumR
TVR	ACT	per-src	21.2±0.1	44.5±0.6	57.0±0.4	87.4±0.2	210.0±1.2
TVR	CHA	per-src	3.5±0.1	12.0±0.1	18.8±0.5	60.7±0.4	94.9±0.8
TVR	QVH	per-src	46.5±0.4	73.2±0.3	82.4±0.2	98.2±0.1	300.3±0.8
TVR	combined	pooled-src	19.0±0.1	39.9±0.4	51.0±0.4	82.6±0.3	192.5±1.0
TVR	TVR	unseen	17.9±0.1	35.4±0.2	44.2±0.1	78.0±0.3	175.4±0.4
ACT	TVR	per-src	31.7±0.4	56.5±0.2	66.6±0.2	93.4±0.1	248.3±0.7
ACT	CHA	per-src	3.4±0.1	11.7±0.2	18.5±0.3	60.3±0.2	93.9±0.5
ACT	QVH	per-src	45.8±0.5	72.4±0.8	80.7±0.3	98.0±0.1	296.9±1.0
ACT	combined	pooled-src	26.4±0.3	47.4±0.1	56.6±0.2	86.0±0.1	216.4±0.6
ACT	ACT	unseen	17.8±0.1	38.2±0.3	50.1±0.1	82.7±0.1	188.7±0.3
CHA	TVR	per-src	31.5±0.3	56.3±0.3	66.4±0.2	93.3±0.2	247.4±0.9
CHA	ACT	per-src	21.6±0.2	44.7±0.1	57.2±0.3	87.4±0.2	210.9±0.7
CHA	QVH	per-src	45.5±1.1	72.8±0.3	81.2±0.5	98.3±0.2	297.7±1.0
CHA	combined	pooled-src	25.7±0.3	49.7±0.2	60.9±0.3	89.7±0.2	225.9±1.0
CHA	CHA	unseen	2.7±0.1	9.5±0.2	15.6±0.4	54.4±0.3	82.1±0.4
QVH	TVR	per-src	31.2±0.6	55.8±0.8	66.0±0.9	93.2±0.3	246.2±2.5
QVH	ACT	per-src	20.9±0.3	43.7±0.5	56.3±0.6	87.2±0.3	208.0±1.5
QVH	CHA	per-src	3.7±0.2	12.0±0.5	19.0±0.2	61.1±0.5	95.7±0.8
QVH	combined	pooled-src	22.2±0.4	43.7±0.6	54.7±0.6	85.6±0.2	206.2±1.7
QVH	QVH	unseen	43.1±0.8	68.5±0.6	77.8±0.3	97.4±0.2	286.7±0.2

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the proposed methods and their empirical improvements, which are supported by the experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses several limitations. GAL does not consistently outperform baselines on every source domain: on TVR and CHA, existing methods benefit more from source-fitting to benchmark-specific characteristics. In addition, GAL requires more trainable parameters and FLOPs than the baselines due to its wider hidden dimension, although its online latency remains small.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present formal theoretical results such as theorems or proofs, focusing instead on methodological design and empirical evaluation.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides detailed descriptions of datasets, model configurations, hyper-parameters, and evaluation settings sufficient to reproduce the main experimental results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The code is not included with the paper. We plan to release the code publicly to support reproducibility. Detailed descriptions of the datasets, evaluation protocol, and hyperparameters are provided in Sec. 4 and Appendices A–B.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides sufficient experimental details such as datasets, model configurations, and key hyper-parameters, which are enough to understand and interpret the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report mean and variance over multiple runs to reflect the variability of the experimental results. In cases where the variance is excessively large and difficult to present clearly, we report only the mean values for readability.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specifies the compute environment, including CPU, GPU, and memory, which is sufficient to understand the hardware setup used for the experiments in implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper involves standard retrieval model development and evaluation on publicly available datasets, with no ethical concerns or violations.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The paper focuses on a general video retrieval method for improving retrieval accuracy, without direct application to societal decision-making or deployment, so it does not present clear or specific societal impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not introduce or release high-risk assets such as generative models or sensitive datasets, so safeguards for misuse are not applicable.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites the original sources of datasets and models, thereby properly crediting and respecting their terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce or release new datasets, models, or code assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or experiments with human subjects, relying only on existing benchmark datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve research with human subjects, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method does not involve LLMs as a component of the proposed approach.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.