PoseBridge: Bridging the Skeletonization Gap for Zero-Shot Skeleton-Based Action Recognition

Sanghyeon Lee **Jinwoo Kim** **Jong Taek Lee***
School of Computer Science and Engineering
Kyungpook National University, Daegu, South Korea
{hyeon1263,kk234556,jongtaeklee}@knu.ac.kr

Abstract

Zero-shot skeleton-based action recognition (ZSSAR) is typically treated as a skeleton-text alignment problem: encode joint-coordinate sequences, align them with language, and classify unseen actions. We argue that this alignment is often too late. Skeletons are not complete action observations, but compressed outputs of human pose estimation (HPE); by the time alignment begins, human-object interactions and pose-relative visual cues may no longer be explicit. We call this upstream semantic loss. To address it, we propose **PoseBridge**, an HPE-aware ZSSAR framework that bridges intermediate HPE representations to skeleton-text alignment. Rather than adding an RGB action branch or object detector, PoseBridge extracts pose-anchored semantic cues from the same HPE process that produces skeletons, then transfers them through skeleton-conditioned bridging and semantic prototype adaptation. Across NTU-RGB+D 60/120, PKU-MMD, and Kinetics-200/400, PoseBridge improves ZSSAR performance under the evaluated protocols. On the Kinetics-200/400 PURLS benchmark, which contains in-the-wild videos with diverse scenes and action contexts, PoseBridge achieves substantial gains over prior ZSSAR methods, outperforming the strongest compared baseline by 13.3–17.4 points across all eight splits. Code is available at https://github.com/hyeon0819/PoseBridge.

1 Introduction

Skeleton-based action recognition represents actions as compact body-joint trajectories, providing motion-centric cues less tied to appearance variation, background clutter, and illumination changes than raw RGB observations [35, 26, 6]. These properties are particularly attractive for zero-shot skeleton-based action recognition (ZSSAR), where models recognize unseen actions by transferring knowledge through shared motion patterns and textual action descriptions [10, 8, 37, 5, 15]. Most ZSSAR methods [10, 8, 36, 37, 5, 38] follow a skeletonize-then-align (S2A) paradigm: raw visual observations are first reduced to joint coordinates by a human pose estimation (HPE) model, and the resulting skeleton sequences are then aligned with text features in a shared semantic space.

However, this paradigm relies on a rarely examined assumption: skeletons preserve sufficient action semantics for language alignment. In practice, a skeleton is a compressed output of an upstream pose estimator rather than a complete action observation. When raw visual evidence is converted into joint coordinates, cues beyond body motion—such as object presence, human-object interaction, and pose-relative object relations—are no longer explicit. Object-sensitive benchmark actions expose this limitation: as shown in Fig. 1(A), "put on a hat/cap", "put on headphone", and "put on glasses" can share similar hand-to-head trajectories, but differ in the manipulated object and its configuration with the actor's hands and head. As a result, S2A may place similar skeleton features near multiple text

*Corresponding author.

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

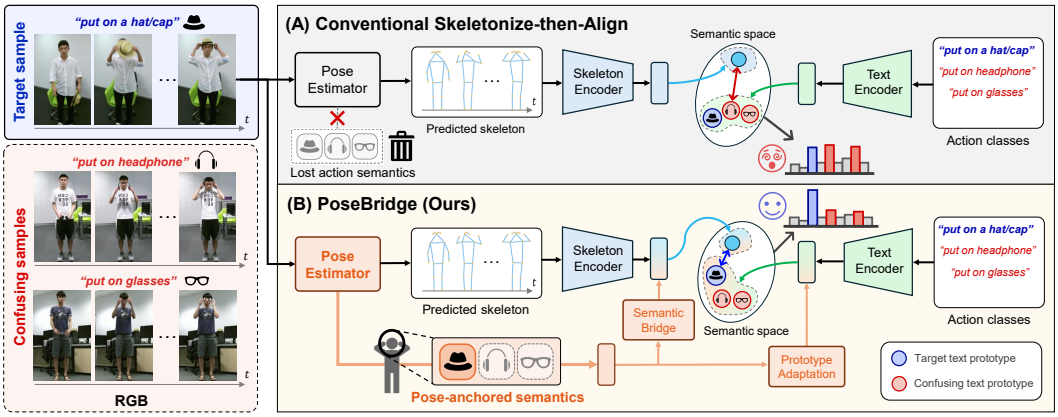

Figure 1: Motivation of PoseBridge. (A) Conventional S2A aligns text with joint-coordinate skeletons after skeletonization, where object, interaction, and pose-relative cues are not explicit. (B) PoseBridge routes HPE-side cues to the alignment space, disambiguating object-sensitive actions with similar skeleton trajectories.

prototypes, producing ambiguous zero-shot predictions. We call this mismatch between the semantics required by action text and the evidence retained in skeletons the skeletonization gap, and identify its source as upstream semantic loss before skeleton-text alignment begins.

A direct way to mitigate this gap is to augment skeletons with additional context. Language- or prompt-based context can enrich skeleton-text alignment with object and scene descriptions [2, 29], but such context is derived from language or skeleton-side inference rather than recovered from the visual evidence available inside the HPE stream. Another option is to introduce an RGB or object-centric visual pathway [19, 27]. However, this changes the modality protocol of ZSSAR and typically requires an additional visual representation beyond the HPE pass used to obtain skeletons, while also exposing the model to appearance or scene shortcuts. Reusing HPE features is closer to the skeleton extraction process [30], but naive feature fusion treats intermediate maps as generic visual features and does not ensure that the retained cues are action-relevant or organized around the human body. The key challenge is therefore not merely to reuse HPE features or append generic context after skeletonization, but to explicitly preserve action-relevant visual evidence within HPE and learn how it should be transferred to skeleton-based recognition. This shifts the focus from improving skeleton-text alignment after information has been discarded to addressing the upstream information bottleneck itself.

To address this challenge, we propose **PoseBridge**, a pose-anchored semantic bridge framework for ZSSAR. As shown in Fig. 1(B), PoseBridge intervenes before skeleton-text alignment by deriving bridge cues from intermediate HPE representations produced during skeleton extraction. Instead of treating HPE as a preprocessing step that only outputs joint coordinates, PoseBridge reinterprets its intermediate representations as pose-anchored semantic cues grounded in the body structure that defines the skeleton. These cues provide object- and pose-relative evidence that joint-coordinate skeletons do not explicitly retain. Importantly, PoseBridge uses cues from the same HPE process that produces skeletons, without an independent RGB action-recognition branch or external object detector.

PoseBridge consists of two coupled stages. First, it learns pose-anchored semantic cues within HPE. Multi-level HPE features are hierarchically refined so that fine-grained visual evidence from shallow layers is injected into deeper pose-structured representations, and body-aware pooling aggregates the refined responses around predicted joints. Auxiliary semantic alignment further encourages the pooled cues to preserve action-relevant evidence rather than generic appearance. Second, PoseBridge transfers these cues to ZSSAR. A skeleton-conditioned semantic bridge lets motion representations query temporal pose-anchored cues, while semantic prototype adaptation calibrates language prototypes toward the pose-semantic space estimated from seen classes. Together, these two stages establish a unified preservation-and-transfer framework, enabling action-relevant

visual evidence to be retained during skeleton extraction and effectively exploited for unseen-action recognition.

Experiments evaluate PoseBridge on NTU-RGB+D 60/120, PKU-MMD, and the broader Kinetics-200/400 PURLS benchmark, covering zero-shot learning (ZSL), generalized ZSL harmonic mean (GZSL-H), and random-split protocols. Beyond the commonly used NTU/PKU-style benchmarks, Kinetics further evaluates ZSSAR under greater class diversity and in-the-wild video conditions, where PoseBridge achieves particularly large improvements over previous methods evaluated under the PURLS protocol.

Our contributions are summarized as follows:

- We identify the skeletonization gap as a fundamental limitation of conventional skeletonize-then-align ZSSAR, revealing that action-relevant object, interaction, and pose-relative cues can become non-explicit before skeleton-text alignment begins. This motivates addressing semantic loss at its source rather than only refining downstream alignment.
- We propose PoseBridge, a pose-anchored semantic bridge framework that addresses this bottleneck through two coupled stages: semantic preservation within HPE via hierarchical refinement and body-aware pooling, and semantic transfer to ZSSAR via skeleton-conditioned bridging and prototype adaptation.
- PoseBridge achieves state-of-the-art ZSL accuracy across all standard NTU-RGB+D 60/120 splits, improves random-split GZSL-H by 11.9/12.9 points on NTU-RGB+D 120/PKU-MMD, and surpasses prior ZSSAR methods by 13.3–17.4 points across eight Kinetics-200/400 PURLS splits.

2 Related Work

Skeleton-text alignment for ZSSAR. Zero-shot skeleton-based action recognition (ZSSAR) recognizes unseen actions by aligning skeleton representations with textual class semantics [1, 24, 32, 38]. Early methods learn skeleton-text compatibility metrics, e.g., Relation Network [10], while later methods improve the semantic structure of this alignment. SynSE [8] decomposes action labels into verb and noun components, and SMIE [36] uses mutual information estimation for cross-modal matching. Recent methods further refine alignment granularity and robustness: PURLS [37] aligns part-level skeleton features with textual descriptions, SA-DVAE [16] separates action-relevant and irrelevant skeleton components, TDSM [5] denoises skeleton features toward descriptive latent representations, and Flora [3] improves semantic modeling with distribution-aware classification. These methods differ in alignment objectives and semantic representations, but largely operate after skeletonization: joint-coordinate sequences are treated as the primary observation, and semantic alignment is performed only after visual evidence has been compressed into skeletons.

Contextual and pose-estimation cues for skeleton recognition. Joint-coordinate skeletons encode body motion but do not explicitly represent objects, interactions, or scene context. Language-side methods enrich skeleton recognition with external or generated descriptions: Neuron [2] uses LLM-generated contextual side information, and SkeletonContext [29] injects prompt-based object and environment cues. These methods improve class-level semantics, but their context is text-derived and cannot directly recover sample-specific visual evidence made non-explicit during skeleton extraction. Other methods introduce visual information to complement skeletons, such as RGB-skeleton fusion in BSZSL [19] and vision-language alignment in SKI-Models [27]; however, such approaches change the modality protocol by adding an additional visual pathway. A related line uses pose-estimation-side representations rather than joint coordinates alone, including pose estimation maps [21], heatmap volumes [6], and intermediate HPE feature maps in PFMESR [30]. These studies highlight the value of HPE representations beyond joint coordinates. In particular, PFMESR reuses intermediate features from a pretrained HPE network for supervised action recognition. However, such reuse does not explicitly preserve visual-language cues for unseen actions or transfer them to zero-shot skeleton-text alignment. In contrast, PoseBridge learns pose-anchored semantic cues within HPE and transfers them through skeleton-conditioned attention and semantic prototype adaptation. Thus, our contribution lies in coupling upstream semantic preservation with downstream zero-shot transfer, rather than HPE feature reuse alone.

3 PoseBridge

3.1 Problem Setup, Recognition Protocol, and Overview

Given an input video $\mathbf{V} = \{\mathbf{I}_t\}_{t=1}^{T}$, a pose estimator Φ_{HPE} predicts joint coordinates $s_t = \Phi_{\text{HPE}}(\mathbf{I}_t)$, forming the skeleton sequence $\mathbf{S}_i = \{s_t\}_{t=1}^{T}$ for the i-th sample. ZSSAR is trained on seen-class samples $\mathcal{D}^s = \{(\mathbf{S}_i, y_i)\}_{i=1}^{N_s}$ with $y_i \in \mathcal{Y}^s$ and evaluated on disjoint unseen classes $\mathcal{Y}^u$, where $\mathcal{Y}^s \cap \mathcal{Y}^u = \varnothing$. In ZSL, predictions are restricted to $\mathcal{Y}^u$; in GZSL, they are made over $\mathcal{Y}^s \cup \mathcal{Y}^u$.

Recognition protocol. PoseBridge follows an HPE-aware ZSSAR protocol. Raw frames are accessed only through the same HPE process used to obtain skeletons. During ZSSAR training and inference, the HPE model is frozen and provides both skeleton sequence $\mathbf{S}_i$ and cached intermediate cue sequences $\mathbf{P}_i = \{\mathbf{p}_{i,t}\}_{t=1}^{T}$, where $\mathbf{p}_{i,t}$ denotes a frame-level pose-anchored cue. PoseBridge uses $\mathbf{S}_i$, $\mathbf{P}_i$, and text descriptions, explicitly leveraging RGB-derived HPE evidence without introducing a separate RGB action-recognition backbone or optical-flow branch. No unseen-class visual samples are used for training or prototype construction.

Typical skeletonize-then-align methods encode $\mathbf{S}_i$ as $\mathbf{z}_s = \mathcal{E}_s(\mathbf{S}_i)$ and align it with text prototypes $\mathbf{t}_c = \mathcal{E}_d(\mathbf{d}_c)$, where $\mathcal{E}_s$ and $\mathcal{E}_d$ denote the pretrained skeleton and text encoders. This alignment occurs after visual evidence has been compressed into joint coordinates. PoseBridge instead learns pose-anchored cues from intermediate HPE representations (Sec. 3.2) and transfers them to skeleton representations and prototypes (Sec. 3.3); training and inference are described in Sec. 3.4.

3.2 Learning Pose-Anchored Semantics within HPE

HPE representations contain action-relevant visual evidence that is often lost after skeleton extraction. To preserve it, we convert intermediate HPE features into pose-anchored semantics, as illustrated in Fig. 2. We refine multi-level features, aggregate them around the predicted body structure, and align them with action text semantics while preserving the original pose objective.

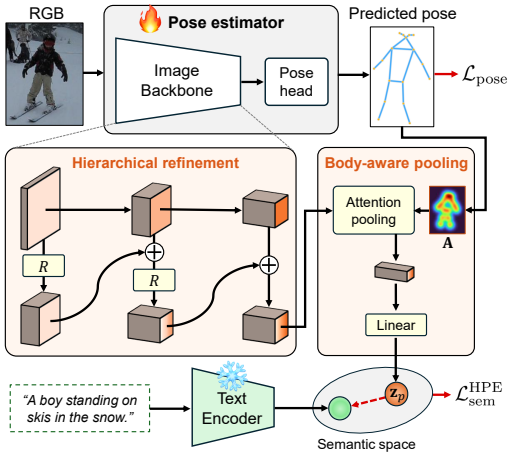

Figure 2: Overview of PoseBridge for learning pose-anchored semantics within HPE.

Hierarchical pose-structured refinement. Intermediate HPE layers provide complementary cues: shallow features retain local appearance and fine-grained spatial details, while deep features encode body configuration and keypoint structure. Using only the deepest feature can therefore discard visual evidence useful for actions requiring fine-grained visual distinctions. We address this by injecting shallower evidence into deeper pose features before semantic pooling.

Let $\{\mathbf{F}^l\}_{l=1}^{L}$ denote shallow-to-deep HPE features of frame $\mathbf{I}$, with $L = 3$ in our experiments. Starting from $\tilde{\mathbf{F}}^1 = \mathbf{F}^1$, each deeper feature is updated as:

$$\tilde{\mathbf{F}}^{l+1} = \mathbf{F}^{l+1} + \alpha\, \mathcal{R}^l\big(\psi^l(\mathcal{P}^l(\tilde{\mathbf{F}}^l))\big), \qquad l = 1, ..., L-1, \tag{1}$$

where $\mathcal{P}^l$ projects the channel dimension and ψ^l resizes the feature to the next-level resolution. $\mathcal{R}^l$ is a convolutional spatial refinement block applied before residual injection, and we fix $\alpha = 0.5$. The final feature $\tilde{\mathbf{F}}^L$ preserves fine-grained visual evidence while maintaining high-level pose structure.

Body-aware semantic pooling. The refined feature map $\tilde{\mathbf{F}}^L$ contains responses over the full spatial map, but useful action semantics should be selected around the actor. We therefore build a body attention map $\mathbf{A}$ from the predicted joint probabilities: joint probabilities are converted into joint heatmaps, averaged over joints, and normalized to form a body-aware spatial prior. The frame-level pose-anchored cue $\mathbf{p}$ is then obtained by body-aware weighted pooling:

$$\mathbf{p} = W_p \left(\frac{\sum_{h,w} \mathbf{A}(h,w)\, \tilde{\mathbf{F}}^L(h,w)}{\sum_{h,w} \mathbf{A}(h,w) + \varepsilon} \right). \tag{2}$$

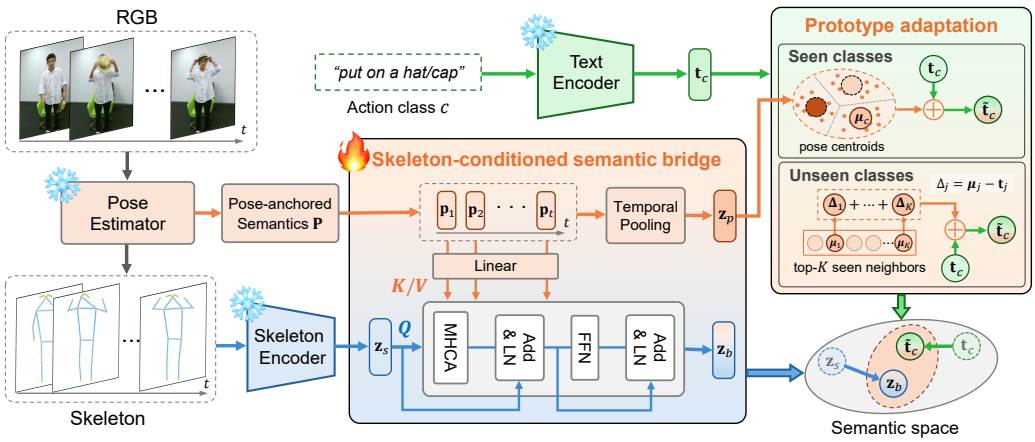

Figure 3: Overview of PoseBridge in the ZSSAR. The pose-anchored semantics extracted from HPE are used for both the skeleton-conditioned semantic bridge and semantic prototype adaptation.

Here, h, w index spatial locations in $\tilde{\mathbf{F}}^L$, W_p projects the pooled feature into the semantic space, and ε is used for numerical stability. Body-aware pooling encourages $\mathbf{p}$ to emphasize visual evidence near the predicted body structure, including local appearance cues around action-relevant body parts.

Action-semantic alignment. To align pose-anchored semantics with language, we use image-level captions from MS COCO [17] as weak image-language supervision. Each paired description $\mathbf{d}$ is encoded by a pretrained CLIP text encoder $\mathcal{E}_d$ and projected into the same embedding space, yielding $\mathbf{t} = W_d\mathcal{E}_d(\mathbf{d})$. Given a batch of B pose-text pairs, we optimize a CLIP-style symmetric contrastive loss:

$$\mathcal{L}_{\text{sem}}^{\text{HPE}} = -\frac{1}{2B}\sum_{i=1}^{B}\left[\log\frac{\exp(\mathbf{p}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^{B}\exp(\mathbf{p}_i^\top \mathbf{t}_j/\tau)} + \log\frac{\exp(\mathbf{p}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^{B}\exp(\mathbf{p}_j^\top \mathbf{t}_i/\tau)}\right], \tag{3}$$

where $\mathbf{p}_i$ and $\mathbf{t}_i$ are L2-normalized embeddings of i-th sample, and $\tau = 0.07$. This contrastive objective aligns pose-anchored cues with textual semantics while separating different samples. The pose estimator is trained with both pose and semantic losses:

$$\mathcal{L}_{\text{HPE}} = \mathcal{L}_{\text{pose}} + \lambda_{\text{hpe}}\,\mathcal{L}_{\text{sem}}^{\text{HPE}}, \tag{4}$$

where $\mathcal{L}_{\text{pose}}$ is the original pose estimation loss and λ_{hpe} balances the two objectives. After training, the frozen HPE model extracts skeleton sequences $\mathbf{S}_i$ and pose-anchored semantic sequences $\mathbf{P}_i = \{\mathbf{p}_{i,t}\}_{t=1}^{T}$, which are passed to the ZSSAR stage as bridge evidence.

3.3 Bridging Pose-Anchored Semantics to Skeleton-Text Alignment

As illustrated in Fig. 3, we use the pose-anchored semantics learned within HPE to bridge skeleton representations and text prototypes in ZSSAR. The frozen HPE model extracts both the skeleton sequence $\mathbf{S}_i$ and the pose-anchored semantic sequence $\mathbf{P}_i$. PoseBridge uses $\mathbf{P}_i$ for two purposes: enriching the skeleton representation through a skeleton-conditioned semantic bridge and adapting text prototypes with pose-semantic cues. Thus, PoseBridge recovers action-relevant semantics from the skeleton extraction pipeline itself, without introducing an additional RGB recognition branch.

Skeleton-conditioned semantic bridge. Given the skeleton sequence $\mathbf{S}_i$, we first obtain the skeleton representation as $\mathbf{z}_s = W_s(\mathcal{E}_s(\mathbf{S}_i))$, where $\mathcal{E}_s$ is the skeleton encoder and W_s projects the skeleton feature into the semantic embedding space. We then use $\mathbf{z}_s$ as a query and the temporal pose-anchored semantic sequence $\mathbf{P}_i$ as keys and values:

$$\mathbf{z}_b = \text{LN}_2(\mathbf{h}_i + \text{FFN}(\mathbf{h}_i)), \qquad \mathbf{h}_i = \text{LN}_1(\mathbf{z}_s + \sigma(\gamma_b)\odot \text{MHCA}(\mathbf{z}_s, W_b\mathbf{P}_i)), \tag{5}$$

where W_b projects pose-anchored semantics into the bridge embedding space, MHCA denotes multi-head cross-attention, LN is layer normalization, σ is the sigmoid function, and γ_b is a learnable element-wise gate. This injects temporal pose-anchored semantics into the skeleton representation, producing $\mathbf{z}_b$ that remains grounded in skeleton motion while shifting toward a more semantically

aligned space. Thus, the bridge makes complementary pose-anchored visual evidence accessible to skeleton-text matching, enabling the model to distinguish actions that are ambiguous from joint-coordinate motion alone.

Semantic prototype adaptation. Even after enriching the query with the skeleton-conditioned bridge, the target prototypes remain purely language-defined and may still be misaligned with the pose-semantic space. We therefore adapt text prototypes while keeping them as the primary anchors for zero-shot generality. From the temporal pose-anchored semantic sequence $\mathbf{P}_i$, we obtain a video-level pose-semantic representation $\mathbf{z}_{p,i}$ using attention-based temporal pooling. This converts frame-level HPE semantics into class-level pose evidence for prototype adaptation.

Let $\mathbf{t}_c$ denote the text prototype of class c. For each seen class, we compute the pose-semantic centroid from its training samples and use it for residual prototype adaptation:

$$\tilde{\mathbf{t}}_c = (1-\rho)\mathbf{t}_c + \rho\boldsymbol{\mu}_c, \qquad \boldsymbol{\mu}_c = \frac{1}{|\mathcal{D}_c^s|}\sum_{1\le i\le|\mathcal{D}_c^s|}\mathbf{z}_{p,i}, \quad c\in\mathcal{Y}^s, \tag{6}$$

where $\mathcal{D}_c^s$ denotes the training samples of seen class c, and ρ is a fixed adaptation coefficient. This preserves the generality of the language prototype while correcting it toward the pose-semantic space.

For unseen classes, no pose samples are available. We therefore transfer the text-to-pose displacement from semantically related seen classes. Let $\mathcal{N}_K(c)$ denote the top-K nearest seen classes to unseen class c in the text prototype space. The unseen prototype is adapted as:

$$\tilde{\mathbf{t}}_c = \mathbf{t}_c + \rho\sum_{j\in\mathcal{N}_K(c)}\omega_{cj}(\boldsymbol{\mu}_j - \mathbf{t}_j), \qquad \omega_{cj} = \frac{\exp(\mathbf{t}_c^\top\mathbf{t}_j/\tau_a)}{\sum_{k\in\mathcal{N}_K(c)}\exp(\mathbf{t}_c^\top\mathbf{t}_k/\tau_a)}, \quad c\in\mathcal{Y}^u. \tag{7}$$

All centroids and adapted prototypes are L2-normalized. Since unseen adaptation uses only residual shifts from nearby seen classes without accessing unseen visual or pose samples, it preserves the zero-shot constraint while making text prototypes more compatible with the pose-semantic space.

3.4 Training Objective and Inference

Training objective. ZSSAR is trained on seen classes only. We optimize three representations with different roles: the skeleton representation $\mathbf{z_s}$, the pose-semantic representation $\mathbf{z_p}$, and the bridge representation $\mathbf{z_b}$. While $\mathbf{z_s}$ and $\mathbf{z_p}$ serve as training anchors for shaping the motion and pose-semantic spaces, $\mathbf{z}_b$ is used as the final representation for zero-shot matching. Let $\mathbf{T}^s = [\mathbf{t}_c]_{c\in\mathcal{Y}^s}$ denote the seen-class text prototype matrix. We define three basic objectives:

$$\mathcal{L}_{\mathrm{cls}}(\mathbf{z}) = \mathrm{CE}(\mathcal{C}(\mathbf{z}), y), \quad \mathcal{L}_{\mathrm{sem}}(\mathbf{z}) = \mathrm{CE}(\mathbf{z}\,(\mathbf{T}^s)^\top, y), \quad \mathcal{L}_{\mathrm{con}}(\mathbf{z}) = \mathrm{SupCon}(\mathbf{z}, y), \tag{8}$$

where $\mathcal{C}$ is a seen-class classifier, CE denotes cross-entropy loss, and SupCon denotes supervised contrastive loss [14], which promotes intra-class compactness and inter-class separation.

For the skeleton and pose anchors, $r\in\{s,p\}$, we use all three losses: $\mathcal{L}_r = \lambda_r^{\mathrm{cls}}\mathcal{L}_{\mathrm{cls}}(\mathbf{z}_r) + \lambda_r^{\mathrm{sem}}\mathcal{L}_{\mathrm{sem}}(\mathbf{z}_r) + \lambda_r^{\mathrm{con}}\mathcal{L}_{\mathrm{con}}(\mathbf{z}_r)$. For the bridge representation, we omit the closed-set classifier and use only semantic matching and contrastive learning: $\mathcal{L}_b = \lambda_b^{\mathrm{sem}}\mathcal{L}_{\mathrm{sem}}(\mathbf{z}_b) + \lambda_b^{\mathrm{con}}\mathcal{L}_{\mathrm{con}}(\mathbf{z}_b)$. This avoids forcing the inference representation $\mathbf{z}_b$ into seen-class classifier boundaries, keeping it better suited for prototype matching with unseen classes. To couple the skeleton and pose-semantic representations, we further use cosine consistency and pose-guided distribution distillation:

$$\mathcal{L}_{\mathrm{align}} = \lambda_{\mathrm{s2p}}\,(1-\cos(\mathbf{z}_s,\mathbf{z}_p)) + \lambda_{\mathrm{kd}}\,\mathrm{KL}\left(\mathrm{softmax}\Big(\frac{\mathbf{z}_p(\mathbf{T}^s)^\top}{\tau_d}\Big)^{\mathrm{sg}}\middle\|\mathrm{softmax}\Big(\frac{\mathbf{z}_s(\mathbf{T}^s)^\top}{\tau_d}\Big)\right), \tag{9}$$

where KL denotes Kullback–Leibler divergence loss, τ_d is the distillation temperature, and sg denotes stop-gradient on the pose-guided teacher distribution. The final objective is $\mathcal{L}_{\mathrm{ZSSAR}} = \mathcal{L}_s + \mathcal{L}_p + \mathcal{L}_b + \mathcal{L}_{\mathrm{align}}$. Overall, this objective shapes $\mathbf{z}_s$ and $\mathbf{z}_p$ into discriminative semantic anchors, while training $\mathbf{z}_b$ as an open prototype-matching representation for unseen-class transfer.

Inference. At inference, we use the bridge representation $\mathbf{z}_b$ as the final query and match it with the adapted prototypes $\tilde{\mathbf{t}}_c$. Since both are L2-normalized, similarity is computed by dot product, and ZSL/GZSL predictions are computed as:

$$\hat{y}_{\mathrm{zsl}} = \arg\max_{c\in\mathcal{Y}^u}\mathbf{z}_b^\top\tilde{\mathbf{t}}_c, \qquad \hat{y}_{\mathrm{gzsl}} = \arg\max_{c\in\mathcal{Y}^s\cup\mathcal{Y}^u}\big(\mathbf{z}_b^\top\tilde{\mathbf{t}}_c - \kappa\,\mathbf{1}[c\in\mathcal{Y}^s]\big), \tag{10}$$

Table 1: Comparison on NTU-RGB+D 60/120 under standard ZSL/GZSL splits. † denotes methods using an additional RGB pathway with a pretrained CLIP visual encoder. The best results are highlighted in **red**, and the second-best results are blue.

Method	Venue	NTU-RGB+D 60 (Xsub)								NTU-RGB+D 120 (Xsub)							
		55/5 Split				48/12 Split				110/10 Split				96/24 Split			
		ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL		
		Acc	*S*	*U*	*H*	*Acc*	*S*	*U*	*H*	*Acc*	*S*	*U*	*H*	*Acc*	*S*	*U*	*H*
ReViSE [9]	ICCV 2017	53.9	74.2	34.7	29.2	17.5	62.4	20.8	31.2	55.0	48.7	44.8	46.7	32.4	49.7	25.1	33.3
JPoSE [31]	ICCV 2019	64.8	64.4	50.3	56.5	28.8	60.5	20.6	30.8	51.9	47.7	46.4	47.0	32.4	38.6	22.8	28.7
CADA-VAE [24]	CVPR 2019	76.8	69.4	61.8	65.4	29.0	51.3	27.0	35.4	59.5	47.2	19.8	48.4	35.8	41.1	34.1	37.3
SynSE [8]	ICIP 2021	75.8	61.3	56.9	59.0	33.3	52.2	27.9	36.3	62.7	52.5	57.6	54.9	38.7	56.4	32.2	41.0
SMIE [36]	ACMMM 2023	78.0	-	-	-	40.2	-	-	-	65.7	-	-	-	45.3	-	-	-
PURLS [37]	CVPR 2024	79.2	-	-	-	41.0	-	-	-	72.0	-	-	-	52.0	-	-	-
SA-DVAE [16]	ECCV 2024	82.4	62.3	70.8	66.3	41.4	50.2	36.9	42.6	68.8	61.1	59.8	60.4	46.1	58.8	35.8	44.5
STAR [1]	ACMMM 2024	81.4	69.0	69.9	69.4	45.1	62.7	37.0	46.6	63.3	59.9	52.7	56.1	44.3	51.2	36.9	42.9
SCoPLe [38]	CVPR 2025	84.1	69.6	71.9	70.8	53.0	54.5	61.8	57.9	74.5	63.5	61.1	62.3	52.2	53.3	51.2	52.2
Neuron [2]	CVPR 2025	86.9	69.1	73.8	71.4	62.7	61.6	56.8	59.1	71.5	67.6	59.5	63.3	57.1	67.5	44.4	53.6
TDSM [5]	ICCV 2025	86.5	-	-	-	56.0	-	-	-	74.2	-	-	-	65.1	-	-	-
FS-VAE [32]	ICCV 2025	86.9	77.0	74.5	75.7	57.2	56.2	48.6	52.1	74.4	59.2	67.9	63.3	62.5	57.8	51.9	54.7
BSZSL† [19]	ESWA 2025	83.0	85.1	72.0	78.0	53.0	73.9	36.5	48.8	77.7	80.8	71.4	**75.8**	56.1	72.1	53.8	61.6
SKI-VLM† [27]	AAAI 2025	82.2	-	-	-	52.0	-	-	-	77.5	-	-	-	59.3	-	-	-
PoseBridge	**This work**	**88.8**	82.0	84.0	**83.0**	**73.2**	75.6	61.9	**68.0**	**80.3**	71.3	65.0	68.0	**70.9**	66.1	59.4	**62.6**

where κ is a seen-class calibration coefficient that mitigates seen-class bias in GZSL, following prior work [2]. Thus, inference explicitly couples the two sides of our bridge: $\mathbf{z}_b$ carries pose-anchored semantics recovered from the skeleton extraction process, while $\tilde{\mathbf{t}}$ provides skeleton-compatible semantic targets, jointly reducing the skeleton-text semantic gap.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate PoseBridge on NTU-RGB+D 60 [25], NTU-RGB+D 120 [20], PKU-MMD [18], and Kinetics-200/400 [12, 37]. NTU-RGB+D 60/120 and PKU-MMD are controlled RGB-D skeleton action benchmarks, while Kinetics-200/400 provides in-the-wild videos with diverse scenes and action contexts. For all datasets, we follow the standard seen/unseen class splits used in prior ZSSAR works [8, 36, 37]. Detailed dataset descriptions are provided in Appendix B.

Evaluation protocol. For zero-shot learning (ZSL), we report top-1 accuracy on unseen classes. For generalized zero-shot learning (GZSL), the prediction space contains both seen and unseen classes. Following prior works, we report the seen-class accuracy S, the unseen-class accuracy U, and their harmonic mean H, which measures the balance between seen and unseen recognition.

Implementation details. We use RTMPose-s [11] as the 2D pose estimator with RTMDet-nano [22] for person detection in the top-down HPE pipeline. The HPE model is trained with the original pose estimation objective and our action-semantic alignment objective on the MS COCO dataset [17], and is then frozen to extract both estimated 2D skeleton sequences and pose-anchored semantics used in the ZSSAR stage. Following common ZSSAR evaluation protocols for fair comparison, we adopt Shift-GCN [4] as the skeleton encoder and use the pretrained CLIP text encoder [23] to obtain semantic prototypes from action descriptions. All models are trained only on seen classes and evaluated on disjoint unseen classes. Detailed implementation settings are provided in Appendix A, and hyperparameter/design choice analyses are presented in Appendix D.

4.2 Comparison with State-of-the-Art

Evaluation on standard split benchmark. We compare PoseBridge with state-of-the-art ZSSAR methods on the standard NTU-RGB+D 60/120 Xsub protocols, following [8]. As shown in Table 1, PoseBridge achieves the best ZSL performance across all settings and substantially improves GZSL performance. The gains are especially clear under harder splits with fewer seen and more unseen classes: on NTU-RGB+D 60 under the 48/12 split, PoseBridge improves the previous best ZSL

Table 2: Comparison on NTU-RGB+D 60/120 and PKU-MMD under random ZSL/GZSL splits.

Method	Venue	NTU-RGB+D 60 (Xsub)		NTU-RGB+D 120 (Xsub)		PKU-MMD (Xsub)	
		55/5 Split		110/10 Split		46/5 Split	
		ZSL	GZSL (*H*)	ZSL	GZSL (*H*)	ZSL	GZSL (*H*)
ReViSE [9]	ICCV 2017	60.9	60.3	44.9	40.3	59.3	49.8
JPoSE [31]	ICCV 2019	59.4	60.1	46.7	43.7	57.2	51.6
CADA-VAE [24]	CVPR 2019	61.8	66.4	45.2	45.6	60.7	45.8
SynSE [8]	ICIP 2021	64.2	67.5	47.3	43.5	53.9	49.5
SMIE [36]	ACMMM 2023	65.1	-	46.4	-	60.8	-
SA-DVAE [16]	ECCV 2024	84.2	75.3	50.7	47.5	66.5	54.7
SCoPLe [38]	CVPR 2025	83.7	77.7	53.3	54.1	71.4	54.9
TDSM [5]	ICCV 2025	88.9	-	69.5	-	70.8	-
FS-VAE [32]	ICCV 2025	-	-	-	-	71.2	59.0
PoseBridge	**This work**	**90.5**	**78.2**	**74.8**	**66.0**	**77.5**	**71.9**

Table 3: Results on Kinetics-200/400 under the PURLS [37] ZSL benchmark. * indicates the use of multiple text prompts; otherwise, a single text prompt is used.

Method	Venue	Kinetics-200				Kinetics-400			
		180/20 Split	160/40 Split	140/60 Split	120/80 Split	360/40 Split	320/80 Split	300/100 Split	280/120 Split
DeVISE [7]	NeurIPS 2013	22.2	12.3	8.0	5.7	18.4	10.2	9.5	8.3
ReViSE [9]	ICCV 2017	25.0	13.3	8.1	6.2	20.8	11.8	9.5	8.2
PURLS [37]	CVPR 2024	26.0	15.9	10.2	7.8	22.5	15.1	11.4	11.0
PURLS* [37]	CVPR 2024	32.2	22.6	12.0	11.8	34.5	24.3	17.0	14.3
TDSM [5]	ICCV 2025	38.2	24.4	15.3	13.1	38.9	26.2	18.5	16.1
PoseBridge	**This work**	**55.6**	**38.3**	**28.6**	**26.4**	**52.3**	**39.6**	**32.9**	**32.1**

accuracy by 10.5 points and the GZSL harmonic mean by 8.9 points. The strong results on NTU-RGB+D 120 under the 96/24 split further confirm the effectiveness of pose-anchored semantics for semantically diverse unseen actions.

Notably, PoseBridge outperforms recent RGB-assisted methods in almost all settings, despite using no additional RGB recognition branch. Instead, it recovers action-relevant visual semantics from intermediate HPE representations already computed during skeleton extraction, showing that the gain comes from better exploiting the skeletonization process rather than adding heavy RGB-based modules. To address potential concerns about skeleton input formats, Appendix C provides a controlled comparison using the same RTMPose-extracted 2D skeletons, where PoseBridge remains superior, along with additional Xview and Xset results.

Evaluation on random split benchmark. Table 2 reports results under the random split protocol introduced by SMIE [36], where three randomly selected seen/unseen splits are evaluated and averaged. PoseBridge again delivers consistent improvements across NTU-RGB+D 60, NTU-RGB+D 120, and PKU-MMD datasets. These results demonstrate that the effectiveness of PoseBridge is consistent across different seen-unseen partitions, indicating that the proposed semantic bridge improves ZSSAR beyond a specific split configuration.

Evaluation on Kinetics-200/400 dataset. We further evaluate PoseBridge on the more challenging Kinetics-200/400 benchmarks following PURLS [37]. Unlike the controlled NTU-RGB+D and PKU-MMD benchmarks, Kinetics contains in-the-wild videos with diverse scenes, viewpoints, subjects, and action contexts. As shown in Table 3, PoseBridge substantially outperforms prior methods across all splits. In particular, PoseBridge improves Kinetics-200 under the 180/20 split by 17.4 points, from 38.2% to 55.6%, and Kinetics-400 under the 280/120 split by 16.0 points, from 16.1% to 32.1%. These substantial gains demonstrate the effectiveness of preserving pose-anchored visual evidence for zero-shot recognition, particularly when joint-coordinate motion alone is insufficient to distinguish actions.

Computational cost. PoseBridge achieves a strong accuracy-efficiency trade-off among recent ZSSAR approaches. As shown in Table 4, it achieves the highest ZSL accuracy across all four standard NTU-RGB+D splits while maintaining the fastest throughput under the recognition-stage evaluation protocol. This comparison measures the ZSSAR recognition stage after skeleton and cue extraction, rather than the complete video-to-action pipeline. For all methods, trainable/total

Table 4: Accuracy-efficiency comparison on NTU-RGB+D 60/120 at the ZSSAR recognition stage. We report ZSL accuracy under standard splits, with parameters, GFLOPs, and throughput.

Method	Venue	NTU-RGB+D 60 (Xsub)		NTU-RGB+D 120 (Xsub)		Trainable / Total #Params (M)	GFLOPs (G)	Throughput (clips/s)
		55/5 Split	48/12 Split	110/10 Split	96/24 Split			
Neuron [2]	CVPR 2025	86.9	62.7	71.5	57.1	2.3 / 126.7	7.5	1165.4
TDSM [5]	ICCV 2025	86.5	56.0	74.2	65.1	260.4 / 384.2	10.6	772.5
FS-VAE [32]	ICCV 2025	86.9	57.2	74.4	62.5	**0.2 / 64.4**	**3.7**	1237.0
SKI-VLM [27]	AAAI 2025	82.2	52.0	77.5	59.3	2.8 / 215.9	281.0	22.6
PoseBridge	**This work**	**88.8**	**73.2**	**80.3**	**70.9**	3.8 / 128.2	7.3	**1398.0**

parameters, GFLOPs, and throughput are measured with the skeleton encoder and text encoder included on a single NVIDIA RTX A6000 GPU. Compared with SKI-VLM [27], which relies on an additional RGB visual encoder and incurs substantially larger GFLOPs, PoseBridge recovers action-relevant semantics from intermediate HPE representations with a much more compact recognition pipeline. TDSM has the largest parameter count due to its diffusion-based backbone and text-conditioning components. Overall, PoseBridge improves zero-shot recognition while preserving the efficiency of a skeleton-based pipeline. Component-wise costs for the full PoseBridge pipeline, from pose estimation to ZSSAR prediction, are provided in Appendix D.3.

4.3 Ablation Study

Component ablation. We analyze the contribution of each PoseBridge component in Table 5. The baseline uses only skeleton sequences from the original pose estimator, without HPE intermediate features or pose-anchored semantics. On the ZSSAR side, the skeleton-conditioned semantic bridge (SB) improves performance by injecting temporal pose-anchored cues, while prototype adaptation (PA) further aligns text prototypes with the pose-semantic space. On the HPE side, hierarchical refinement (HR) and body-aware pooling (BP) provide complementary gains by preserving and selecting body-aware semantics. The full model performs best, confirming the benefit of combining HPE-side semantic preservation with ZSSAR-side semantic transfer.

Table 5: Ablation study of PoseBridge components on NTU-RGB+D 60 (48/12 split).

HPE		ZSSAR		NTU-RGB+D 60 (Xsub, 48/12 Split)	
HR	BP	SB	PA	ZSL	GZSL *(H)*
-	-	-	-	54.7	52.0
-	-	-	✓	56.6	57.3
-	-	✓	-	60.4	60.7
-	-	✓	✓	61.4	61.7
-	✓	✓	✓	64.0	63.2
✓	-	✓	✓	64.8	63.8
✓	✓	-	✓	66.9	63.9
✓	✓	✓	-	71.0	66.5
✓	✓	✓	✓	**73.2**	**68.0**

Table 6: Matched raw-video comparisons on NTU-RGB+D 60 (48/12) and Kinetics-200 (120/80). Except for the RGB-only variant, all methods use skeleton inputs. PoseBridge-HPE denotes our learned HPE cues, and PoseBridge-ZSSAR comprises semantic bridging and prototype adaptation. Efficiency is measured over 16-frame clips.

	Configuration		ZSL (%)		End-to-end efficiency		
Variant	**Visual cue source**	**Integration**	**NTU60**	**K200**	**Params**	**GFLOPs**	**Throughput**
A	None	Skeleton-only	54.7	19.1	132.0	23.47	238.6
B	VideoCLIP	RGB-only	47.2	27.8	212.7	77.28	112.4
C	VideoCLIP	Simple fusion	59.6	**29.6**	220.1	94.09	79.0
D	Vanilla HPE	Simple fusion	57.3	19.3	132.2	23.47	235.2
E	PoseBridge-HPE	Simple fusion	62.3	22.3	135.6	27.92	233.0
F	Vanilla HPE	PoseBridge-ZSSAR	61.4	19.2	134.7	23.49	233.6
G	**PoseBridge-HPE**	**PoseBridge-ZSSAR**	**73.2**	26.4	138.1	27.94	232.8

Matched controls. To isolate the effects of visual cues and integration strategies, Table 6 compares skeleton-only recognition (A), large-scale pretrained VideoCLIP [33] as a separate RGB branch (B), and its fusion with skeleton features (C), alongside HPE-based variants (D–G). Vanilla HPE cues reuse intermediate features learned solely for pose estimation, whereas PoseBridge-HPE explicitly learns pose-anchored semantics. Simple fusion concatenates skeleton and visual features followed

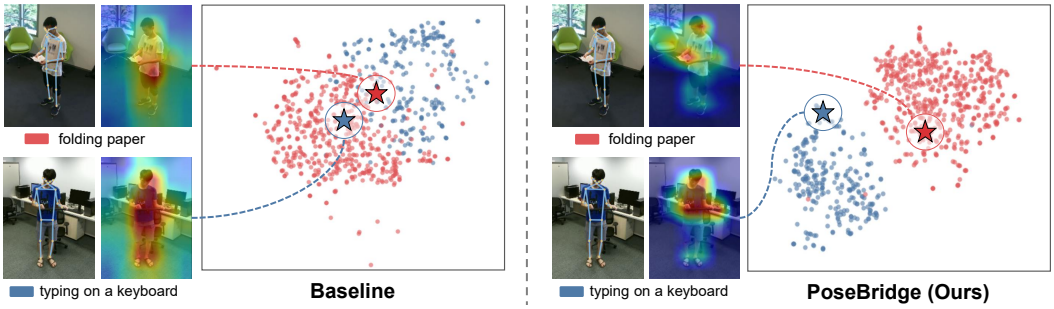

Figure 4: Qualitative comparison between the baseline and PoseBridge on the confusing action pair "folding paper" and "typing on a keyboard". We visualize predicted skeletons, HPE Grad-CAM maps, and t-SNE embeddings of zero-shot matching features.

by a two-layer MLP. Under identical fusion, PoseBridge-HPE improves NTU60/K200 accuracy by 5.0/3.0 points (D to E), while PB-ZSSAR further improves it to 73.2/26.4 (E to G), demonstrating complementary semantic preservation and transfer. Compared with VideoCLIP fusion (C), PoseBridge achieves higher NTU60 accuracy (73.2 vs. 59.6), though lower on K200 (26.4 vs. 29.6), while requiring 3.4× fewer GFLOPs and achieving 2.9× higher throughput without a separate RGB backbone.

4.4 Qualitative Analysis

We qualitatively verify that PoseBridge recovers semantic cues that are largely missing from skeleton coordinates alone. In Fig. 4, we compare a baseline trained only with the original pose estimation objective and our HPE model trained with action-semantic alignment, hierarchical refinement, and body-aware pooling. For the confusing action pair "folding paper" and "typing on a keyboard", Grad-CAM shows that the baseline mainly responds to the overall human body, reflecting its bias toward keypoint localization. In contrast, our pose-anchored feature focuses more strongly on action-relevant regions, such as the keyboard, the paper, and hand-object interaction areas. The t-SNE plots show a consistent trend: baseline features are highly mixed, while PoseBridge yields much clearer separation between the two classes. These results show that PoseBridge captures action-relevant semantics during HPE and transfers them effectively to skeleton-text matching in zero-shot recognition.

5 Conclusion

We presented PoseBridge, an HPE-aware framework for ZSSAR that addresses a fundamental limitation of skeletonize-then-align approaches: the loss of action-relevant visual evidence during skeletonization. Rather than treating HPE solely as preprocessing, PoseBridge preserves pose-anchored semantics from intermediate HPE representations and transfers them to skeleton-text alignment through a skeleton-conditioned semantic bridge and prototype adaptation. Experiments on NTU-RGB+D 60/120, PKU-MMD, and Kinetics-200/400 demonstrate consistent improvements, particularly on challenging splits and in-the-wild videos. Importantly, PoseBridge achieves these gains without a separate RGB action-recognition backbone, highlighting HPE as an effective upstream semantic source. Overall, PoseBridge demonstrates the importance of preserving visual semantics before skeleton-text alignment for advancing zero-shot recognition.

Acknowledgments

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2025-25425223), and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2025-25416432). This work was also supported by the Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT (MSIT, Korea) & Gwangju Metropolitan City.

References

[1] Yang Chen, Jingcai Guo, Tian He, Xiaocheng Lu, and Ling Wang. Fine-grained side information guided dual-prompts for zero-shot skeleton action recognition. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 778–786, 2024.

[2] Yang Chen, Jingcai Guo, Song Guo, and Dacheng Tao. Neuron: Learning context-aware evolving representations for zero-shot skeleton action recognition. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8721–8730, 2025.

[3] Yang Chen, Miaoge Li, Zhijie Rao, Deze Zeng, Song Guo, and Jingcai Guo. Learning by neighbor-aware semantics, deciding by open-form flows: Towards robust zero-shot skeleton action recognition. *arXiv preprint arXiv:2511.09388*, 2025.

[4] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 183–192, 2020.

[5] Jeonghyeok Do and Munchurl Kim. Bridging the skeleton-text modality gap: Diffusion-powered modality alignment for zero-shot skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12757–12768, October 2025.

[6] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2969–2978, 2022.

[7] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013.

[8] Pranay Gupta, Divyanshu Sharma, and Ravi Kiran Sarvadevabhatla. Syntactically guided generative embeddings for zero-shot skeleton action recognition. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 439–443. IEEE, 2021.

[9] Yao-Hung Hubert Tsai, Liang-Kang Huang, and Ruslan Salakhutdinov. Learning robust visual-semantic embeddings. In *Proceedings of the IEEE International conference on Computer Vision*, pages 3571–3580, 2017.

[10] Bhavan Jasani and Afshaan Mazagonwalla. Skeleton based zero shot action recognition in joint pose-language semantic space. *arXiv preprint arXiv:1911.11344*, 2019.

[11] Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. Rtmpose: Real-time multi-person pose estimation based on mmpose. *arXiv preprint arXiv:2303.07399*, 2023.

[12] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[13] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

[14] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[15] Jidong Kuang, Hongsong Wang, Chaolei Han, Yang Zhang, and Jie Gui. Zero-shot skeleton-based action recognition with dual visual-text alignment. *Pattern Recognition*, page 112342, 2025.

[16] Sheng-Wei Li, Zi-Xiang Wei, Wei-Jie Chen, Yi-Hsin Yu, Chih-Yuan Yang, and Jane Yung-jen Hsu. Sa-dvae: Improving zero-shot skeleton-based action recognition by disentangled variational autoencoders. In *European conference on computer vision*, pages 447–462. Springer, 2024.

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[18] Chunhui Liu, Yueyu Hu, Yanghao Li, Sijie Song, and Jiaying Liu. Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv preprint arXiv:1703.07475*, 2017.

[19] Hongjie Liu, Yingchun Niu, Kun Zeng, Chun Liu, Mengjie Hu, and Qing Song. Beyond-skeleton: Zero-shot skeleton action recognition enhanced by supplementary rgb visual information. *Expert Systems with Applications*, 273:126814, 2025.

[20] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701, 2019.

[21] Mengyuan Liu and Junsong Yuan. Recognizing human actions as the evolution of pose estimation maps. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1159–1168, 2018.

[22] Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. Rtmdet: An empirical study of designing real-time object detectors. *arXiv preprint arXiv:2212.07784*, 2022.

[23] 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.

[24] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 54–57, 2019.

[25] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[26] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12026–12035, 2019.

[27] Arkaprava Sinha, Dominick Reilly, Francois Bremond, Pu Wang, and Srijan Das. Ski models: Skeleton induced vision-language embeddings for understanding activities of daily living. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 6931–6939, 2025.

[28] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019.

[29] Ning Wang, Tieyue Wu, Naeha Sharif, Farid Boussaid, Guangming Zhu, Lin Mei, Mohammed Bennamoun, et al. Skeletoncontext: Skeleton-side context prompt learning for zero-shot skeleton-based action recognition. *arXiv preprint arXiv:2603.29692*, 2026.

[30] Hao Wen, Zhe-Ming Lu, Fengli Shen, Ziqian Lu, Yangming Zheng, and Jialin Cui. Enhancing skeleton-based action recognition with feature maps from pose estimation networks. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, page 2024EAP1162, 2025.

[31] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 450–459, 2019.

[32] Wenhan Wu, Zhishuai Guo, Chen Chen, Hongfei Xue, and Aidong Lu. Frequency-semantic enhanced variational autoencoder for zero-shot skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11122–11131, 2025.

[33] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6787–6800, 2021.

[34] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in neural information processing systems*, 35: 38571–38584, 2022.

[35] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[36] Yujie Zhou, Wenwen Qiang, Anyi Rao, Ning Lin, Bing Su, and Jiaqi Wang. Zero-shot skeleton-based action recognition via mutual information estimation and maximization. In *Proceedings of the 31st ACM international conference on multimedia*, pages 5302–5310, 2023.

[37] Anqi Zhu, Qiuhong Ke, Mingming Gong, and James Bailey. Part-aware unified representation of language and skeleton for zero-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18761–18770, 2024.

[38] Anqi Zhu, Jingmin Zhu, James Bailey, Mingming Gong, and Qiuhong Ke. Semantic-guided cross-modal prompt learning for skeleton-based zero-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13876–13885, 2025.

Appendix

This appendix provides additional details, results, and analyses that support the main paper. Section A describes the implementation details of the HPE and ZSSAR stages. Section B summarizes the datasets, seen/unseen split settings, and action descriptions. Section C reports additional performance comparisons, including challenging splits, Xview/Xset protocols, controlled comparisons with estimated 2D skeleton inputs, and the effect of semantic alignment on pose estimation. Section D analyzes key design choices, including hyperparameters, component-wise computational costs, and different pose estimators. Section E provides confusion matrix comparisons and additional analyses of object-sensitive recognition, representation quality, robustness, semantic supervision, and statistical reliability. Section F presents additional qualitative results, including t-SNE and Grad-CAM visualizations. Section G discusses limitations and broader impacts.

A Implementation Details

A.1 HPE Details

We use RTMPose-s [11] as the 2D pose estimator and train it on MS COCO [17]. For the top-down pose estimation pipeline, we employ RTMDet-nano [22] for person detection, followed by RTMPose-s for keypoint estimation. Both the skeleton coordinates and pose-anchored semantic cues are extracted from this shared HPE pipeline. Following the standard RTMPose-s setting, we use an input resolution of 256×192 and train the model for 420 epochs with AdamW. The base learning rate is set to 4×10^{-3}, the weight decay is set to 0, and the learning rate is adjusted with a linear warm-up followed by cosine annealing. The model is optimized with the original SimCC-based pose estimation loss and our action-semantic alignment loss described in Sec. 3.2.

For action-semantic alignment, we use COCO captions as image-level textual supervision. We encode each caption using a frozen CLIP ViT-B/32 text encoder [23] and map the resulting feature to the pose-semantic embedding space with a lightweight projection head. The projection dimension is set to 512, and the semantic loss weight is set to $\lambda_{\mathrm{hpe}} = 0.1$. The CLIP-style contrastive loss for semantic alignment is applied from the beginning of training.

For the pose-anchored semantic cue extractor, we use $L = 3$ feature levels by default. The hierarchical refinement gate is fixed to $\alpha = 0.5$. After training, the HPE model is frozen and used to extract both 2D skeleton sequences and pose-anchored semantic cues for the ZSSAR stage. Additional analyses on design choices, including different pose estimators such as HRNet [28] and ViTPose [34], are reported in Appendix D.

A.2 ZSSAR Details

We use Shift-GCN [4] as the skeleton encoder and the pretrained CLIP ViT-L/14 text encoder [23] for semantic prototype extraction. The CLIP text encoder is frozen, and action descriptions are encoded and cached before training. The skeleton feature dimension is 256, the pose-semantic cue dimension is 512, and the shared embedding dimension is 512. The skeleton projector uses two MLP layers, while the pose adapter uses one MLP layer. The dropout ratio is set to 0.1.

The skeleton-conditioned semantic bridge uses multi-head cross-attention with 4 heads. For each video, we use 16 temporal pose-anchored cues. The skeleton representation is used as the query, and the temporal pose-anchored cues are used as keys and values. For prototype adaptation, we use the top-$K = 5$ nearest seen classes in the text-prototype space. The prototype adaptation coefficient is set to $\rho = 0.2$.

We train PoseBridge with AdamW for 30 epochs. The batch size is 128, the test batch size is 256, the learning rate is 1×10^{-3}, and the weight decay is 2×10^{-3}. We use 5 warm-up epochs followed by cosine annealing with a minimum learning rate of 1×10^{-6}. Gradient clipping is applied with a maximum norm of 1.0. We also use an exponential moving average (EMA) with decay 0.999 starting from epoch 5. All experiments are conducted on a single NVIDIA RTX A6000 GPU.

For the loss weights, we set $\lambda_s^{\mathrm{cls}} = 1.0$, $\lambda_s^{\mathrm{sem}} = 1.5$, $\lambda_s^{\mathrm{con}} = 1.5$, $\lambda_p^{\mathrm{cls}} = 0.5$, $\lambda_p^{\mathrm{sem}} = 0.5$, $\lambda_p^{\mathrm{con}} = 0.3$, $\lambda_{\mathrm{s2p}} = 0.3$, $\lambda_{\mathrm{kd}} = 1.0$, $\lambda_b^{\mathrm{sem}} = 1.0$, and $\lambda_b^{\mathrm{con}} = 0.5$. The supervised contrastive temperature is set to

0.07, and the distillation temperature is set to 4.0. Additional analyses on design choices are reported in Appendix D.

B Dataset Details

B.1 Dataset Details

NTU-RGB+D 60. NTU-RGB+D 60 [25] is a large-scale RGB-D action recognition benchmark containing 56,880 samples from 60 action classes performed by 40 subjects. The dataset was captured by three Kinect V2 cameras and provides multi-modal observations, including RGB videos, depth maps, infrared sequences, and 3D skeleton annotations. The standard evaluation protocols are cross-subject (Xsub) and cross-view (Xview), which evaluate generalization across different subjects and camera viewpoints, respectively. For zero-shot evaluation, we follow the seen/unseen class splits adopted in prior ZSSAR works [8, 36, 37].

NTU-RGB+D 120. NTU-RGB+D 120 [20] extends NTU-RGB+D 60 to 114,480 samples from 120 action classes. It is also captured with Kinect V2 sensors and provides RGB, depth, infrared, and skeleton modalities. Compared with NTU-RGB+D 60, it contains a larger action vocabulary and more diverse acquisition setups, making it a more challenging benchmark for zero-shot generalization. The standard evaluation protocols are cross-subject (Xsub) and cross-setup (Xset), where Xset evaluates generalization across different camera and environment setups. We follow the standard ZSSAR seen/unseen splits used in previous works [8, 36, 37].

PKU-MMD. PKU-MMD [18] is a multi-modal RGB-D benchmark for human action understanding captured with Kinect v2 sensors. Following prior ZSSAR protocols [8, 36, 37], we use the commonly adopted 51-class setting, which contains 1,076 long video sequences and nearly 20,000 action instances performed by 66 subjects from three camera views. The standard evaluation protocols are cross-subject (Xsub) and cross-view (Xview).

Kinetics-200/400. Kinetics-400 [12] is a large-scale in-the-wild video action recognition dataset collected from YouTube videos. Unlike the NTU-RGB-D 60/120 and PKU-MMD datasets above, Kinetics contains larger variations in scene, viewpoint, subject appearance, camera motion, and action context. Following the PURLS benchmark [37], Kinetics-200 is constructed from the first 200 classes of Kinetics-400 and evaluated under multiple seen/unseen splits. We additionally evaluate on Kinetics-400 using analogous splits to test scalability under a larger action vocabulary.

Skeleton extraction. For all datasets, we extract 2D skeletons from RGB frames using RTMPose [11]. We also obtain pose-anchored semantic cues from the same pose estimation process.

Auxiliary HPE training data. We train the 2D pose estimator on MS COCO [17]. In addition to the original pose estimation supervision, we use image-level textual descriptions from COCO captions as semantic supervision for the proposed action-semantic alignment loss. Specifically, the pose-anchored semantic cue extracted from the HPE representation is aligned with the corresponding caption feature encoded by the pretrained CLIP text encoder [23]. MS COCO is used only for training the HPE model and is not used for ZSSAR training.

B.2 Seen/Unseen Split Details

We follow the seen/unseen class splits used in prior ZSSAR protocols for fair comparison. For NTU-RGB+D 60, we evaluate the 55/5 and 48/12 splits; for NTU-RGB+D 120, we evaluate the 110/10 and 96/24 splits. For PKU-MMD, we use the standard 46/5 split adopted in previous ZSSAR works. For Kinetics-200, we follow the PURLS benchmark [37] and evaluate the 180/20, 160/40, 140/60, and 120/80 splits. For Kinetics-400, we use analogous 360/40, 320/80, 300/100, and 280/120 splits. All models are trained only on the seen classes of each split, and no samples from unseen classes are used during training.

B.3 Action Description Details

We use LLM-generated action descriptions to construct text prototypes for ZSSAR. Following the semantic setting introduced by SMIE [36], each action category label is expanded into a richer action description using ChatGPT. For example, the category label "drink water" is expanded into "the

Table 7: Results under more challenging seen/unseen splits on NTU-RGB+D 60 and NTU-RGB+D 120. The best results are highlighted in **red**, and the second-best results are blue.

Method	Venue	NTU-RGB+D 60 (Xsub)		NTU-RGB+D 120 (Xsub)	
		40/20 Split	30/30 Split	80/40 Split	60/60 Split
ReViSE [9]	ICCV 2017	24.3	14.8	19.5	8.3
JPoSE [31]	ICCV 2019	20.1	12.4	13.7	7.7
CADA-VAE [24]	CVPR 2019	16.2	11.5	10.6	5.7
SynSE [8]	ICIP 2021	19.9	12.0	13.6	7.7
PURLS [37]	CVPR 2024	31.1	23.5	28.4	19.6
SCoPLe [38]	CVPR 2025	32.0	18.2	25.3	15.7
TDSM [5]	ICCV 2025	36.1	25.9	37.0	27.2
PoseBridge	**This work**	**41.4**	**37.2**	**43.7**	**30.4**

Table 8: Comparison on NTU-RGB+D 60 under the Xview protocol and NTU-RGB+D 120 under the Xset protocol (standard seen/unseen splits). We report ZSL accuracy and GZSL seen accuracy S, unseen accuracy U, and harmonic mean H.

Method	Venue	NTU-RGB+D 60 (Xview)								NTU-RGB+D 120 (Xset)							
		55/5 Split				48/12 Split				110/10 Split				96/24 Split			
		ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL		
		Acc	*S*	*U*	*H*	*Acc*	*S*	*U*	*H*	*Acc*	*S*	*U*	*H*	*Acc*	*S*	*U*	*H*
ReViSE [9]	ICCV 2017	54.4	25.8	29.3	27.4	17.2	34.2	16.4	22.1	30.2	4.0	23.7	6.8	13.5	2.6	3.4	2.9
JPoSE [31]	ICCV 2019	72.0	61.1	59.5	60.3	28.9	29.0	14.7	19.5	52.8	23.6	4.4	7.4	38.5	79.3	2.6	4.9
CADA-VAE [24]	CVPR 2019	75.1	65.7	56.1	60.5	32.9	49.7	25.9	34.0	52.5	46.0	44.5	45.2	38.7	47.6	26.8	34.3
SynSE [8]	ICIP 2021	68.0	65.5	45.6	53.8	29.9	61.3	24.6	35.1	59.3	58.9	49.2	53.6	41.4	46.8	31.8	37.9
SMIE [36]	ACMMM 2023	79.0	-	-	-	41.0	-	-	-	57.0	-	-	-	42.3	-	-	-
STAR [1]	ACMMM 2024	81.6	71.9	70.3	71.1	42.5	66.2	37.5	47.9	65.3	59.3	59.5	59.4	44.1	53.7	34.1	41.7
Neuron [2]	CVPR 2025	**87.8**	70.6	75.9	73.2	63.3	65.3	58.1	61.5	71.1	67.5	58.9	62.9	54.0	67.0	44.9	53.8
PoseBridge	**This work**	**87.8**	83.9	84.1	**84.0**	**65.9**	73.7	57.1	**64.4**	**77.1**	70.3	64.5	**67.3**	**65.0**	67.9	55.1	**60.8**

act of taking in water by mouth to hydrate or quench thirst as a human action." These action-level descriptions are encoded by the frozen CLIP text encoder [23] and used as semantic prototypes for both seen and unseen classes.

Recent ZSSAR methods further exploit LLM-generated semantics by constructing additional fine-grained descriptions, such as body-part-specific, temporal-phase-specific, or local/global action descriptions [37, 1, 32]. In contrast, PoseBridge does not introduce extra fine-grained LLM descriptions on the text side. We use the same action-level descriptions following SMIE and instead recover additional action-relevant semantics from the HPE process through pose-anchored semantic cues. This design ensures that the performance gain does not come from stronger text-side description engineering, but from the proposed pose-anchored semantic bridge.

C Additional Performance Comparisons

C.1 Results under More Challenging Seen/Unseen Splits

We further evaluate PoseBridge under more challenging seen/unseen splits, where the number of unseen classes is substantially increased. As summarized in Table 7, PoseBridge consistently achieves the best performance across all settings. The improvement is especially pronounced on the most challenging NTU-RGB+D 60 30/30 split, where PoseBridge improves the previous best result from 25.9% to 37.2%. These results indicate that recovering pose-anchored semantics is particularly useful when the model must distinguish among a larger and more diverse set of unseen action categories.

C.2 Results on Xview and Xset Protocols

We also test whether PoseBridge generalizes beyond the Xsub protocols used in the main paper. Table 8 reports additional results on NTU-RGB+D 60 Xview and NTU-RGB+D 120 Xset. PoseBridge

Table 9: Controlled comparison under the same estimated 2D skeleton input setting. All methods use the same RTMPose-extracted COCO-17 skeletons as input. We report ZSL accuracy and GZSL harmonic mean H on NTU-RGB+D 60/120 standard splits.

Method	Venue	NTU-RGB+D 60 (Xsub)				NTU-RGB+D 120 (Xsub)			
		55/5 Split		48/12 Split		110/10 Split		96/24 Split	
		ZSL	GZSL (*H*)	ZSL	GZSL (*H*)	ZSL	GZSL (*H*)	ZSL	GZSL (*H*)
Neuron [2]	CVPR 2025	88.3	61.1	58.8	49.0	66.5	48.2	60.2	45.6
TDSM [5]	ICCV 2025	86.9	-	64.7	-	76.3	-	59.7	-
FS-VAE [32]	ICCV 2025	87.8	32.7	48.8	41.8	62.6	46.9	56.2	48.5
PoseBridge	**This work**	**88.8**	**83.0**	**73.2**	**68.0**	**80.3**	**68.0**	**70.9**	**62.6**

Table 10: HPE-stage pose estimation quality and efficiency. We compare the standard RTMPose-s with the PoseBridge-enhanced RTMPose-s used to extract both skeletons and pose-anchored semantics. AP is evaluated on the MS COCO validation set, while parameters, GFLOPs, and FPS are measured for the HPE model.

Model	Training objective	AP	#Params (M)	GFLOPs (G)	FPS
RTMPose-s	$\mathcal{L}_{\text{pose}}$	70.86	5.47	0.68	4740.71
RTMPose-s + PoseBridge	$\mathcal{L}_{\text{pose}} + \mathcal{L}_{\text{sem}}^{\text{HPE}}$	70.24	8.85	0.96	2534.7

achieves strong performance across both protocols and remains competitive or superior in both ZSL and GZSL settings. These results confirm that the proposed bridge is not limited to subject-based generalization, but also remains effective under viewpoint and setup shifts.

C.3 Fair Comparison with Estimated 2D Skeletons

Most prior ZSSAR methods report results using Kinect-based 3D skeletons with 25 body joints, whereas PoseBridge uses estimated 2D skeletons in the COCO-17 format extracted by RTMPose [11]. To ensure a fair comparison under the same skeleton input setting, we re-evaluate representative recent methods, including Neuron [2], TDSM [5], and FS-VAE [32], using the same RTMPose-extracted 2D skeletons as PoseBridge.

As shown in Table 9, PoseBridge remains consistently superior under this controlled setting. This confirms that the improvement does not come from an input-format advantage, but from the proposed pose-anchored semantic preservation and transfer mechanism. Even with the same estimated 2D skeleton inputs, PoseBridge effectively recovers action-relevant cues lost during skeleton extraction and transfers them to the semantic prototype space. This further demonstrates that our gains cannot be explained by differences in skeleton input formats alone, but benefit from the proposed pose-anchored semantic preservation and transfer.

C.4 Effect of Semantic Alignment on Pose Estimation

Since PoseBridge modifies the HPE model by adding semantic alignment during HPE training, we examine its effect on both pose estimation quality and HPE-stage efficiency. We compare the standard RTMPose-s setting with our HPE model trained using the action-semantic alignment objective in Sec. 3.2. We evaluate pose estimation quality using mean Average Precision (AP) on the MS COCO [17] validation set, and report the parameters, GFLOPs, and FPS of the pose estimator.

Different from Table 4, which measures only the ZSSAR recognition stage after skeleton extraction, Table 10 isolates the HPE cost. As shown in Table 10, our HPE model maintains comparable pose estimation quality to the original RTMPose-s [11], with AP changing only from 70.86 to 70.24. The added semantic alignment increases the HPE-stage cost moderately, from 0.68 to 0.96 GFLOPs, while preserving real-time throughput. This indicates that PoseBridge can learn pose-anchored semantic cues without substantially degrading pose estimation quality or HPE efficiency. Thus, the downstream ZSSAR gains are not caused by an improved skeleton format, but by more semantically useful HPE representations.

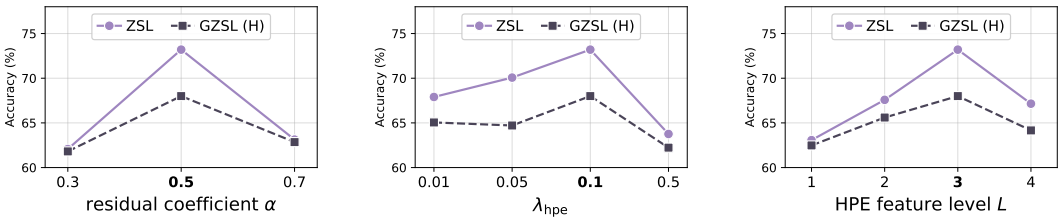

Figure 5: HPE-side hyperparameter analysis on NTU-RGB+D 60 under the standard 48/12 split. We vary the hierarchical refinement residual coefficient α, the semantic alignment weight λ_{hpe}, and the number of HPE feature levels L, while keeping the other settings fixed.

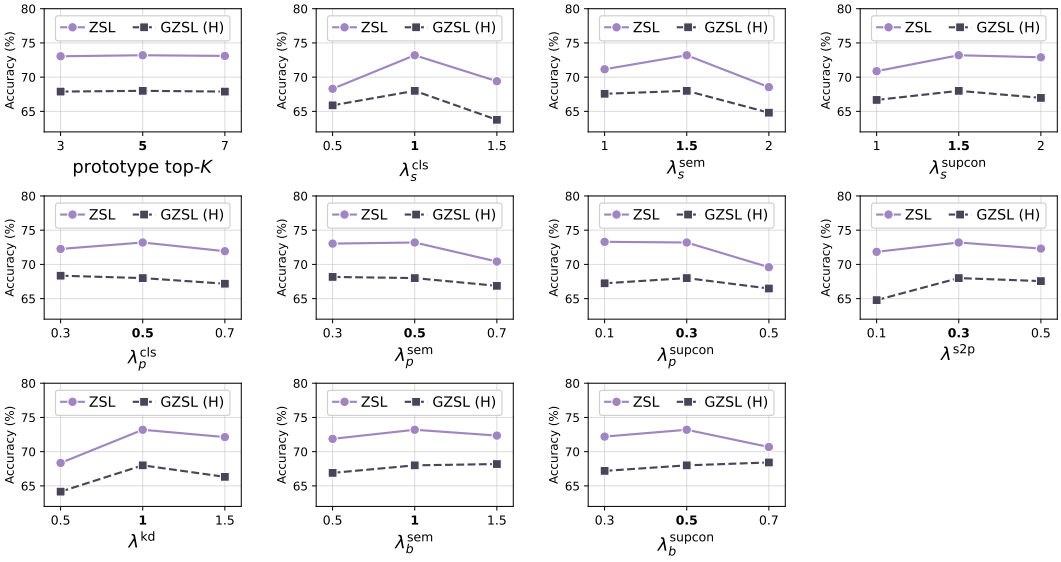

Figure 6: ZSSAR-side hyperparameter analysis on NTU-RGB+D 60 under the standard 48/12 split. We vary prototype top-K and key loss weights λ while keeping the other settings fixed.

D Additional Ablation and Design Studies

D.1 HPE-side Hyperparameters

We analyze the HPE-side hyperparameters used in Sec. 3.2. Specifically, we vary the residual coefficient α in hierarchical refinement, the semantic alignment weight λ_{hpe}, and the number of HPE feature levels L. As shown in Fig. 5, the default setting $\alpha = 0.5$, $\lambda_{\text{hpe}} = 0.1$, and $L = 3$ provides the best overall performance. Using a small λ_{hpe} weakens semantic alignment, while an overly large value can disturb pose-oriented representation learning. Similarly, using too few HPE feature levels limits access to fine-grained visual cues, whereas adding more levels may introduce less action-relevant low-level responses. These results support our default HPE design for extracting pose-anchored semantic cues.

D.2 ZSSAR-side Hyperparameters

We further analyze the ZSSAR-side hyperparameters used in Sec. 3.3 and Sec. 3.4. Figure 6 shows the sensitivity to prototype adaptation, skeleton-side losses, pose-semantic losses, cross-branch regularization, and bridge-side losses. Overall, PoseBridge is stable across a reasonable range of hyperparameters, and the default values used in our experiments generally provide the best or near-best performance. The model is relatively insensitive to the prototype top-K, while loss weights related to semantic alignment and cross-branch transfer show clearer effects. This is consistent with

Table 11: Component-wise computational profiles of PoseBridge. The upper block reports individual module costs, while the lower block shows cumulative configurations from the skeleton-only baseline to the full model. Detector and HPE costs are aggregated over 16-frame clips, and throughput is measured on a single NVIDIA RTX A6000.

Component	#Params (M)	GFLOPs (G)	Throughput (clips/s)
Individual module profiles			
RTMDet-nano	0.99	5.12	550.5
RTMPose-s	5.47	10.91	723.0
Shift-GCN	0.8	0.778	3536.8
CLIP Text Encoder	123.7	6.548	–
Cumulative pipeline configurations			
Baseline	131.95	23.47	238.6
+HR	135.33	27.92	234.5
+HR+BP	135.33	27.92	233.7
+HR+BP+SB	138.05	27.94	233.0
+HR+BP+SB+PA (Full)	138.05	27.94	232.8

Table 12: Analysis with different HPE settings and 2D pose estimators. RTMPose (original) uses estimated skeletons without HPE intermediate features or pose-anchored semantic cues. For PoseBridge-HRNet, PoseBridge-ViTPose, and PoseBridge-RTMPose, we apply the same PoseBridge framework and report HPE performance, ZSSAR performance, parameters, GFLOPs, and FPS.

HPE Setting	MS COCO (AP)	NTU-RGB+D 60 (Xsub) 48/12 Split		NTU-RGB+D 120 (Xsub) 96/24 Split		#Params (M)	GFLOPs (G)	FPS
		ZSL	GZSL *(H)*	ZSL	GZSL *(H)*			
RTMPose (original)	70.9	54.7	52.0	55.1	52.4	5.5	0.7	4740.7
PoseBridge-HRNet [28]	70.8	65.3	63.8	**71.0**	61.7	28.8	7.8	766.9
PoseBridge-ViTPose [34]	**73.7**	69.2	61.8	69.8	61.7	27.4	6.2	1020.6
PoseBridge-RTMPose [11]	70.2	**73.2**	**68.0**	70.9	**62.6**	**8.9**	**1.0**	**2534.7**

our motivation: the final performance depends not only on skeleton-text matching, but also on how effectively pose-anchored semantics are transferred to the skeleton representation.

D.3 Component-wise Computational Cost

We analyze the computational cost of PoseBridge in Table 11. The upper block profiles individual modules, including RTMDet-nano, RTMPose-s, Shift-GCN, and the CLIP text encoder. The lower block reports cumulative pipeline configurations, progressively adding HR, BP, SB, and PA to the skeleton-only baseline. Under the 16-frame profiling setting, the full model increases the reported complexity from 23.47 to 27.94 GFLOPs while maintaining comparable throughput. Most additional computation arises from HPE-side semantic preservation, whereas BP and PA introduce negligible overhead. These results highlight the efficiency of exploiting the existing HPE stream without a separate RGB action-recognition backbone.

D.4 Different Pose Estimators

We evaluate whether PoseBridge is robust to the choice of the 2D pose estimator. We first include RTMPose-s (original) as a skeleton-only baseline, which uses estimated skeleton sequences without HPE intermediate features or pose-anchored semantic cues. In addition to our default RTMPose-s [11], we experiment with HRNet-w32 [28] and ViTPose-s [34]. Each pose estimator is trained on MS COCO using its standard pose-estimation recipe and hyperparameters, while applying the same HPE-side action-semantic alignment objective described in Sec. 3.2. As shown in Table 12, PoseBridge consistently improves over the skeleton-only baseline and remains effective across different pose estimators. We also report MS COCO AP, parameter count, GFLOPs, and FPS for

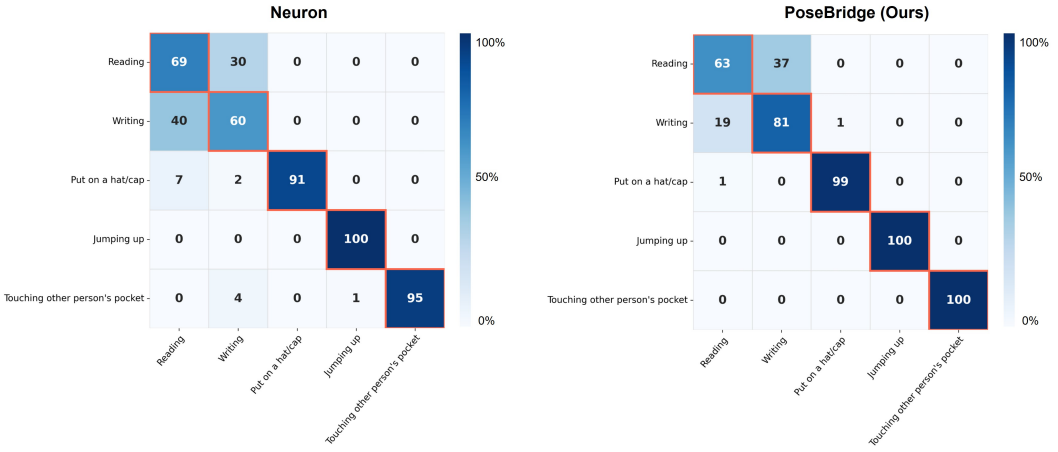

Figure 7: Row-normalized confusion matrices of Neuron and PoseBridge on NTU-RGB+D 60 under the standard 55/5 split.

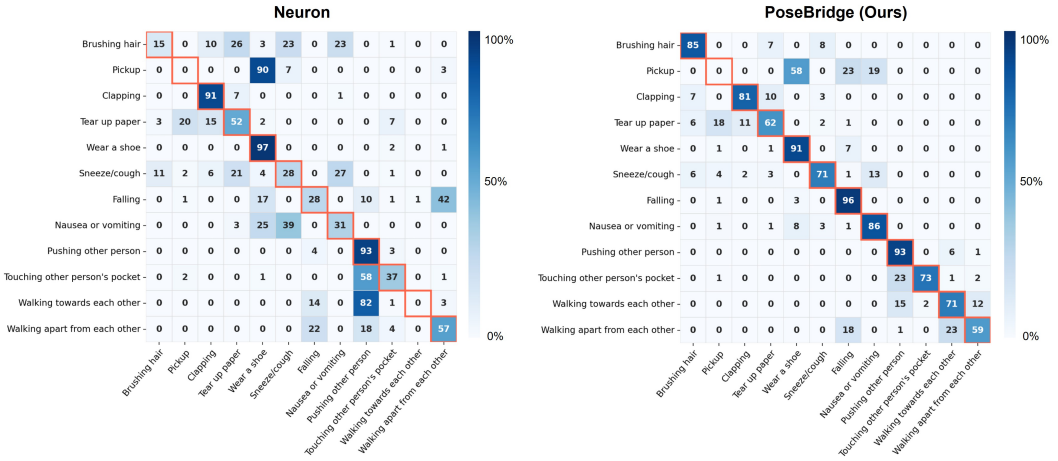

Figure 8: Row-normalized confusion matrices of Neuron and PoseBridge on NTU-RGB+D 60 under the standard 48/12 split.

reference. These results indicate that the proposed pose-anchored semantic bridge is not tied to a specific HPE backbone.

E Additional Analyses

E.1 Confusion Matrix Analysis

We provide additional confusion matrix visualizations to analyze class-level prediction behavior under the same skeleton input setting. Specifically, we compare Neuron [2] and PoseBridge using RTMPose-extracted skeletons on the standard NTU-RGB+D 60/120 splits. As shown in Fig. 7–10, PoseBridge generally produces stronger diagonal responses and fewer off-diagonal confusions than Neuron. This trend is meaningful because both methods use the same estimated 2D skeleton inputs, indicating that the improvement comes from the proposed pose-anchored semantic bridge rather than differences in skeleton format. The results further show that pose-anchored semantics help distinguish semantically similar or motion-ambiguous actions in zero-shot recognition.

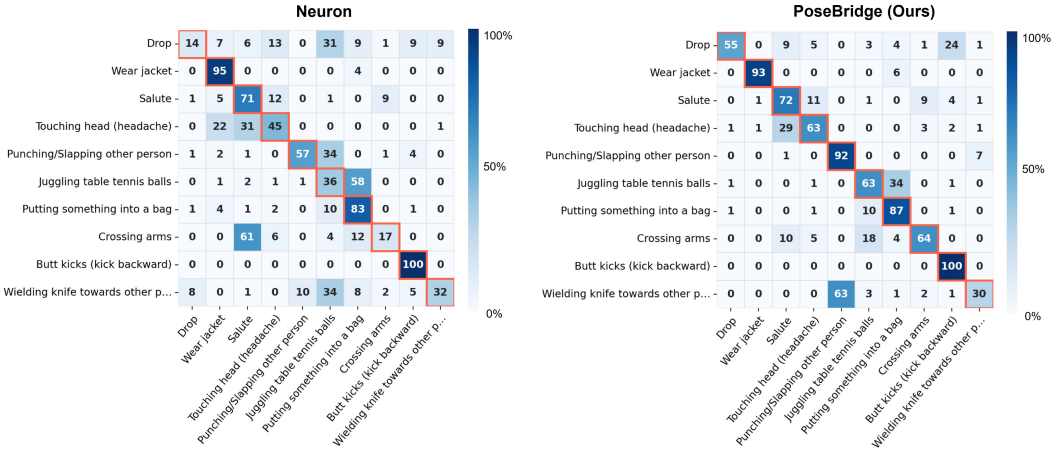

Figure 9: Row-normalized confusion matrices of Neuron and PoseBridge on NTU-RGB+D 120 under the standard 110/10 split.

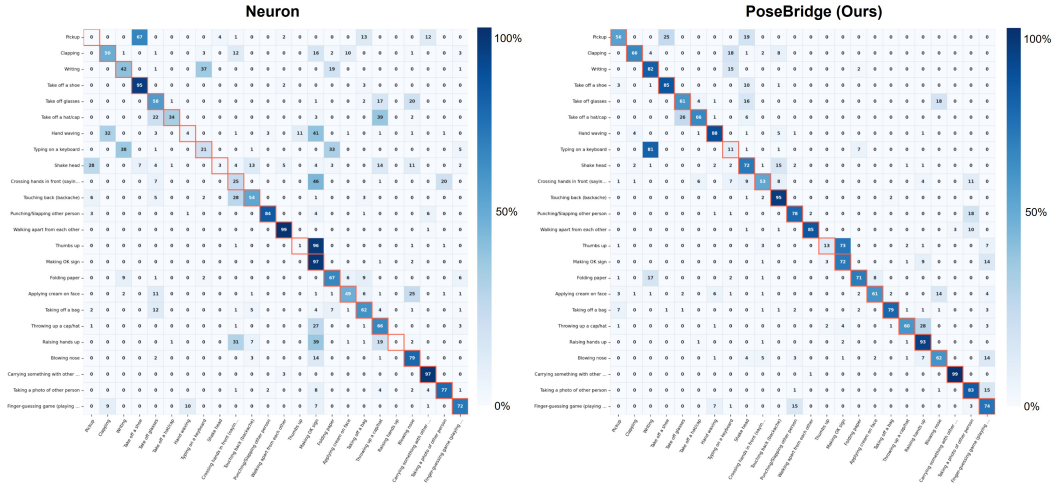

Figure 10: Row-normalized confusion matrices of Neuron and PoseBridge on NTU-RGB+D 120 under the standard 96/24 split.

E.2 Object-Recognition Feature Baseline

To examine whether object recognition alone explains PoseBridge's improvements, we compare against a baseline using pretrained YOLOv11 [13] object features extracted from person-centered crops and associated with pose anchors. As shown in Table 13, object-recognition features improve ZSL accuracy from 54.7% to 56.6% and GZSL-H from 52.0% to 56.1%, whereas PoseBridge achieves 73.2% and 68.0%, respectively. These results demonstrate that PoseBridge captures richer pose-anchored visual evidence beyond object identity alone, including body configuration and pose-relative interaction cues.

E.3 Analysis by Action Semantics

To investigate the benefits of pose-anchored semantics across different action types, we divide the 12 unseen classes of NTU-RGB+D 60 (48/12) into motion-centric and object/interaction-sensitive groups based on their action descriptions, before examining predictions. As shown in Table 14, PoseBridge achieves larger improvements on object/interaction-sensitive actions (+20.0 points) than on motion-centric actions (+15.4 points). This supports our skeletonization-gap hypothesis that

Table 13: Comparison with object-recognition features on NTU-RGB+D 60 (48/12). Object features are obtained from a pretrained YOLOv11 detector on person-centered crops. We report ZSL accuracy and GZSL harmonic mean (H) in percentages.

Method	**ZSL**	**GZSL (H)**
Skeleton-only	54.7	52.0
Skeleton + object-recognition features	56.6	56.1
PoseBridge	**73.2**	**68.0**

Table 14: ZSL accuracy (%) by unseen action group on NTU-RGB+D 60 (48/12). Object/interaction-sensitive actions require additional object or interaction cues for discrimination, whereas motion-centric actions primarily depend on body motion.

Unseen action group	**#Cls.**	**Skeleton-only**	**PoseBridge**	**Gain**
Motion-centric	4	68.2	83.6	+15.4
Object/interaction-sensitive	8	47.9	67.9	**+20.0**
All unseen classes	12	54.7	73.2	+18.5

Table 15: Spatial concentration of HPE Grad-CAM activations on NTU-RGB+D 60 (48/12). Top-25% activation mass measures the fraction of total normalized activation within the most highly activated 25% of spatial locations.

Metric	**Vanilla HPE**	**PoseBridge-HPE**
Top-25% activation mass	0.387	**0.411**

Table 16: Quantitative separability of zero-shot matching embeddings on NTU-RGB+D 60 (48/12). Metrics are computed on the original L2-normalized embeddings before t-SNE projection. Higher values indicate stronger class separation.

Embedding metric	**Skeleton-only**	**PoseBridge**
Silhouette score ↑	0.268	**0.336**
Inter-/intra-class distance ratio ↑	2.560	**3.295**

preserving visual evidence is particularly beneficial when body motion alone cannot fully distinguish semantically different actions.

E.4 Quantitative Representation Analysis

We complement the Grad-CAM and t-SNE visualizations in Fig. 4 with quantitative analyses of spatial activation and embedding separability. As shown in Table 15, the top-25% activation mass increases from 0.387 for vanilla HPE to 0.411 for PoseBridge-HPE, indicating more concentrated spatial responses. Moreover, Table 16 shows improvements in silhouette score (0.268 to 0.336) and inter-/intra-class distance ratio (2.560 to 3.295), measured on the original L2-normalized embeddings. These results provide quantitative support for the improved representation quality achieved by PoseBridge.

E.5 Robustness Under Input Degradation

We evaluate PoseBridge under two controlled degradation settings on NTU-RGB+D 60 (48/12): Gaussian perturbations applied to both skeletons and HPE cues, and random block occlusions applied to RGB frames before rerunning the full pipeline. As shown in Table 17, PoseBridge consistently outperforms skeleton-only recognition under perturbation (67.9% vs. 48.1%) and occlusion (65.5%

Table 17: Robustness evaluation on NTU-RGB+D 60 (48/12) under skeleton/HPE-feature perturbation and random RGB block occlusion. Results are ZSL accuracy (%), and Δ denotes the change from the original input condition.

Input condition	**Skeleton**	Δ	**PoseBridge**	Δ
Original input	54.7	–	73.2	–
Skeleton/HPE perturbation	48.1	-6.6	67.9	-5.3
Random RGB block occlusion	46.8	-7.9	65.5	-7.7

Table 18: Effect of HPE-side semantic supervision on NTU-RGB+D 60 (48/12). All variants use the same PoseBridge-ZSSAR stage. The visual distillation variant does not use paired captions during HPE adaptation, although its CLIP teacher was pretrained with image–text supervision. Results are ZSL accuracy and GZSL harmonic mean (H) in percentages.

HPE-side supervision	**ZSL**	**GZSL (H)**
None (vanilla HPE features)	61.4	61.7
Frozen CLIP visual teacher	67.0	64.4
COCO caption alignment (Ours)	**73.2**	**68.0**

Table 19: Three-seed evaluation on the standard NTU-RGB+D 60/120 Xsub splits. Results are mean $\pm$ standard deviation (%) over three independent training seeds with fixed seen/unseen class partitions.

Dataset (Split)	**ZSL**	**GZSL (S)**	**GZSL (U)**	**GZSL (H)**
NTU60 (55/5)	88.6 ± 1.0	81.9 ± 1.0	83.6 ± 1.3	82.7 ± 1.0
NTU60 (48/12)	73.1 ± 1.5	75.2 ± 1.3	61.8 ± 1.7	67.8 ± 0.9
NTU120 (110/10)	80.8 ± 0.9	71.0 ± 0.8	66.3 ± 1.0	68.6 ± 0.6
NTU120 (96/24)	70.5 ± 1.2	67.9 ± 1.7	58.9 ± 2.2	62.7 ± 0.9

vs. 46.8%). These results demonstrate partial robustness to degraded observations, although severe pose-estimation failures and persistent occlusion remain challenging.

E.6 Alternative HPE-Side Semantic Supervision

We investigate the effect of alternative semantic supervision by replacing COCO caption alignment with visual distillation from a frozen pretrained CLIP visual encoder, without using paired captions during HPE training. As shown in Table 18, visual distillation improves ZSL accuracy from 61.4% to 67.0%, while our COCO caption alignment achieves 73.2%. Although the CLIP teacher itself is pretrained with image–text supervision, these results demonstrate the flexibility of our framework and highlight the effectiveness of explicit visual–language alignment for zero-shot transfer.

E.7 Statistical Reliability

To assess statistical reliability, we repeat the four standard NTU-RGB+D 60/120 Xsub settings with three independent random seeds while keeping the seen/unseen class splits fixed. As shown in Table 19, PoseBridge achieves consistent performance across runs, including 73.1$\pm$1.5% ZSL accuracy and 67.8$\pm$0.9% GZSL-H on NTU60 (48/12). These results confirm that the reported improvements remain stable under training randomness.

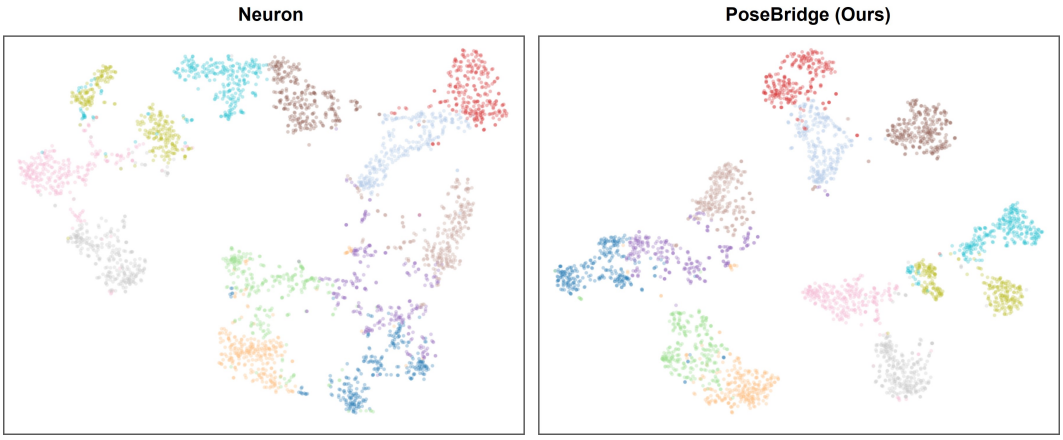

Figure 11: t-SNE visualization of zero-shot matching features on NTU-RGB+D 60 under the standard 48/12 split. We compare Neuron and PoseBridge using the same RTMPose-extracted 2D skeleton inputs.

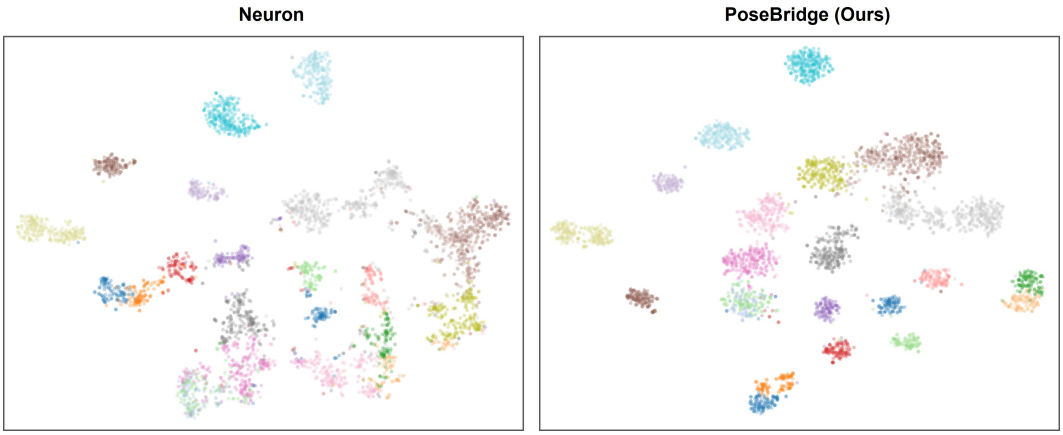

Figure 12: t-SNE visualization of zero-shot matching features on NTU-RGB+D 120 under the standard 96/24 split. We compare Neuron and PoseBridge using the same RTMPose-extracted 2D skeleton inputs.

F Additional Qualitative Results

F.1 t-SNE Visualization Against Neuron

To further analyze the representation quality of PoseBridge, we visualize zero-shot matching features using t-SNE and compare them with Neuron [2]. Both methods are evaluated under the same RTMPose-extracted 2D skeleton input setting, so the comparison focuses on the learned representation space rather than differences in skeleton format. As shown in Fig. 11 and Fig. 12, PoseBridge produces more compact intra-class clusters and clearer inter-class separation, whereas Neuron shows more scattered embeddings with noticeable overlap across classes. This indicates that the proposed pose-anchored semantics help organize skeleton features into a more discriminative semantic space. The trend is consistent on both NTU-RGB+D 60 under the 48/12 split and NTU-RGB+D 120 under the 96/24 split, suggesting that PoseBridge remains effective even when the unseen action space becomes larger and more ambiguous. These visualizations further support that bridging HPE-derived semantics to skeleton-text matching improves zero-shot class separability.

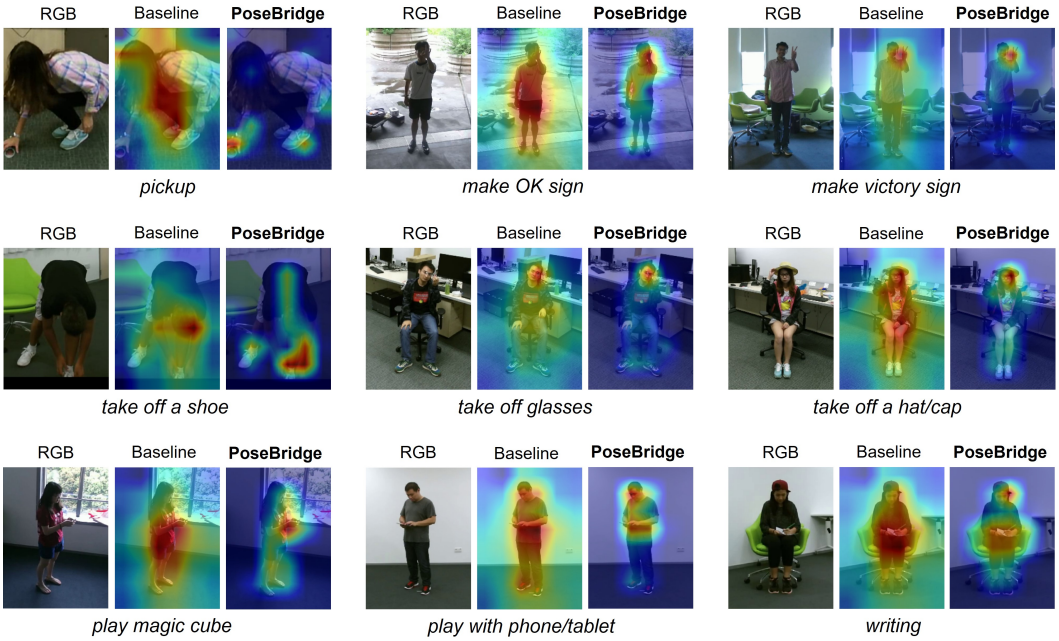

Figure 13: Additional qualitative comparison between the baseline and PoseBridge. We show RGB samples, Grad-CAM maps of baseline HPE features, and Grad-CAM maps of PoseBridge pose-anchored semantics for various actions.

F.2 Additional Grad-CAM Visualizations

We provide additional Grad-CAM examples in Fig. 13. The baseline HPE model, trained only for pose estimation, tends to activate broadly over the human body, reflecting its focus on keypoint localization rather than action semantics. In contrast, PoseBridge produces more localized and action-discriminative responses. For actions such as "pickup," "take off a shoe," "take off glasses," "take off a hat/cap," "play magic cube," "play with phone/tablet," and "writing," PoseBridge focuses more strongly on action-related visual cues around the manipulated object and the interacting body parts. For fine-grained gestures such as "make OK sign" and "make victory sign," it also highlights subtle hand and finger regions that are critical for distinguishing the action. These results further show that PoseBridge preserves action-relevant semantics during HPE, rather than relying only on coordinate-level skeleton information, which benefits downstream zero-shot skeleton-text matching.

G Limitations and Broader Impacts

Limitations. PoseBridge reduces the skeletonization gap by reusing pose-anchored semantic cues from the same HPE process that produces skeletons. Its effectiveness therefore depends on the quality and semantic coverage of the HPE stream. When the actor is heavily occluded, the manipulated object is very small or outside the body-attended region, or the pose estimator fails under severe motion blur or crowded scenes, both the skeletons and the pose-anchored cues may become unreliable. In addition, PoseBridge is designed for an HPE-aware ZSSAR setting rather than a strictly coordinate-only skeleton protocol, since it assumes access to intermediate HPE representations during skeleton extraction. Future work could further improve robustness under challenging visual conditions and investigate lighter ways to preserve pose-relative semantic cues.

Broader Impacts. PoseBridge aims to improve zero-shot skeleton action recognition, which can reduce the need for exhaustive action-specific annotation and may benefit applications such as human-computer interaction, rehabilitation analysis, sports motion understanding, and assistive robotics. However, action recognition systems can also raise privacy and fairness concerns when applied to real-world videos, especially in surveillance-like or high-stakes settings. Although PoseBridge

does not introduce an independent RGB action-recognition branch, it still relies on videos during skeleton extraction and may inherit biases from the pose estimator, caption supervision, and pretrained language models. We therefore recommend careful dataset-specific validation, privacy-preserving deployment, and human oversight before using such systems in sensitive applications.

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 present the motivation, proposed method, and main contributions of the paper.

   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: We provide the limitations of this research in Appendix G. We described it in the appendix rather than the main paper due to the page limit

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not include formal theoretical analysis or proofs. Our contributions are primarily methodological and empirical.

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: We provide detailed experimental settings for reproducibility in Sections 4.1, B, and C, including details of the datasets, evaluation protocols, and implementation settings.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We use established benchmark datasets and publicly release our implementation, training and evaluation scripts, and reproduction instructions at https://github.com/hyeon0819/PoseBridge.

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: We include the dataset details in Appendix B, along with implementation settings in Appendix A.

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 accuracy and standard deviation over three independent training seeds on the four standard splits in Appendix E.7.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We include the computer resources in Appendix A.2.

   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: We understand the code of ethics and adhere to it.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses both potential positive applications and possible negative societal impacts, including privacy concerns, bias inheritance, and risks in surveillance-related settings, in Appendix G

    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: This research poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We use publicly available datasets and open-source code and properly cite them.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We publicly release the PoseBridge implementation with documentation, training and evaluation scripts, and instructions for reproducing the experimental results at `https://github.com/hyeon0819/PoseBridge`.

    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: This paper does not involve crowdsourcing nor research with human subjects.

    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: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: We use ChatGPT to generate action-level text descriptions following established ZSSAR protocols. This usage is described in Appendix B.3; the LLM itself is not trained or modified as part of our proposed framework.

    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.