Less is More: Compact-Token Masked Feature Prediction for Skeleton Representation Learning

Jeonghyeok Do
Information & Electronics Research Institute
KAIST
ehwjdgur0913@kaist.ac.kr

Chen Yun
School of Electrical Engineering
KAIST
cyruby@kaist.ac.kr

Geunhyuk Youk
School of Electrical Engineering
KAIST
rmsgurkjg@kaist.ac.kr

Munchurl Kim*
School of Electrical Engineering
KAIST
mkimee@kaist.ac.kr

Project page: https://kaist-viclab.github.io/SLiM_site/

Abstract

Current skeleton representation learning paradigms face distinct limitations: Contrastive Learning (CL) often overlooks fine-grained motion details, while Masked Auto-Encoders (MAE) rely on coordinate-level reconstruction. This reconstruction inherently demands dense token sequences and heavy decoders, wasting pre-training computation on discarded components and forcing downstream inference to process dense token grids. To resolve these bottlenecks, we propose **SLiM** (**S**keleton **L**ess **i**s **M**ore), a compact-token framework that unifies masked feature prediction and contrastive learning via a shared encoder. By shifting the objective from raw coordinate reconstruction to decoder-free, teacher-guided feature prediction, SLiM breaks the reliance on dense tokenization and enables effective learning with a highly compact token grid. Crucially, to prevent trivial shortcut learning arising from strong inter-joint dependencies of human, we introduce Semantic Tube Masking together with Skeleton-Aware Augmentations to enforce deep skeletal-temporal reasoning and anatomical consistency. Extensive experiments across multiple downstream protocols demonstrate that SLiM achieves state-of-the-art performance while structurally reducing inference computation by 7.89× compared to dense-token MAE baselines.

1 Introduction

Skeleton data provides a compact and informative representation for human action understanding, offering inherent robustness to background and viewpoint changes compared to RGB videos [14, 32, 40]. While supervised skeleton-based action recognition has achieved strong performance [4, 5, 8], it depends on costly annotations and often generalizes poorly to unseen domains. Self-supervised learning (SSL) has therefore become an essential direction for learning transferable motion representations from unlabeled sequences [2, 3, 11, 24, 42].

Existing skeleton SSL methods mainly follow two paradigms: Contrastive Learning (CL) and Masked Auto-Encoding (MAE). CL-based methods [10, 17, 23, 39] learn sequence-level invariance but often overlook fine-grained local patterns, and constructing semantically consistent views is non-trivial for complex actions. MAE-based methods [1, 22, 31, 38] address this by predicting missing information,

*Corresponding author.

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

but typically rely on fine-grained coordinate reconstruction (Fig. 2(a)). This inherently demands dense skeletal-temporal tokenization and heavy full-sequence decoders. Furthermore, extreme masking ratios (e.g., 90%) make the pretext task exceedingly difficult, forcing models to rely on even denser input frames to gather sufficient skeletal-temporal clues. Consequently, this strict dependence on dense tokenization triggers a chain of inefficiencies: wasted pre-training computation on discarded decoders, and a severe inference asymmetry where the encoder must process the entire dense, unmasked token grid.

As empirically shown in Fig. 1, simply reducing the token budget in standard MAEs severely degrades accuracy [22, 38], confirming their reliance on dense coordinate contexts. To break this bottleneck, we propose SLiM (**S**keleton **L**ess **i**s **M**ore), a unified SSL framework that combines masked feature prediction and contrastive learning. By reformulating the objective as a decoder-free, teacher-guided feature prediction task (Fig. 2(b)), SLiM shifts the target from low-level coordinates to high-level semantics. This paradigm shift completely eliminates the need for dense tokenization and heavy decoders. Furthermore, instead of a fixed extreme masking ratio, SLiM applies stochastic mask-token corruption (ranging from 50% to 90%) over a compact token grid. This dynamic masking strategy significantly increases the diversity of masked inputs, promoting robust representation learning and explicitly boosting downstream performance. Ultimately, this token-efficient paradigm structurally resolves the computational asymmetry, drastically reducing both pre-training overhead and downstream costs.

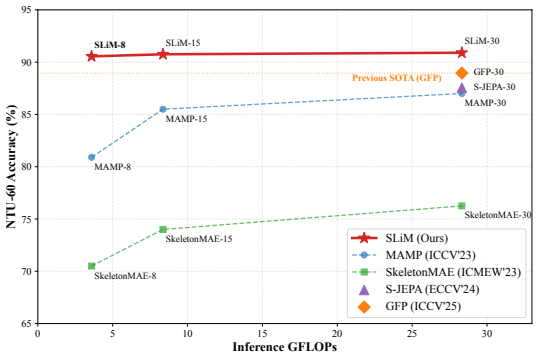


Figure 1: Accuracy–efficiency trade-off on NTU-60. Accuracy is averaged over X-Sub and X-View, and inference cost is measured by downstream GFLOPs. The suffix denotes the temporal token budget (N_T). SLiM maintains SOTA accuracy with much lower inference cost than dense-token MAE-based methods.

However, compact feature prediction alone is vulnerable to shortcut learning; strong skeletal-temporal correlations allow independent joint masking [1, 22, 31] to be trivially solved via local interpolation. To enforce true action-level reasoning, we introduce *Semantic Tube Masking*, which occludes connected anatomical regions over consecutive frames. For the contrastive branch, we propose *Skeleton-Aware Augmentations* such as skeleton-aware rotation, skeleton-aware mirroring, and bone-aware scaling to generate diverse views while strictly preserving anatomical validity. These components provide challenging masked inputs and semantically reliable contrastive pairs. Our contributions are summarized as follows:

- We propose **SLiM**, a compact-token skeleton SSL framework that uses a shared encoder to combine decoder-free masked feature prediction with global–local contrastive learning, completely bypassing the inefficiencies of coordinate reconstruction.
- We design **Semantic Tube Masking** and **Skeleton-Aware Augmentations** to support compact-token learning by avoiding interpolation-based shortcuts and ensuring anatomical consistency.
- Extensive experiments demonstrate that SLiM achieves SOTA performance across multiple downstream protocols while reducing inference cost by 7.89× compared to dense-token MAE baselines.

2 Related Work

Contrastive skeleton SSL. Contrastive learning (CL) [3] learns skeleton representations by aligning augmented views of the same sequence. CrosSCLR [15] learns view-invariant features through cross-view consistency, while AimCLR [10] improves robustness with strong skeleton augmentations. HaLP [25] mines latent positives, and ActCLR [17] decomposes actions into static and dynamic components. Recent methods such as USDRL [36] incorporate feature decorrelation objectives.

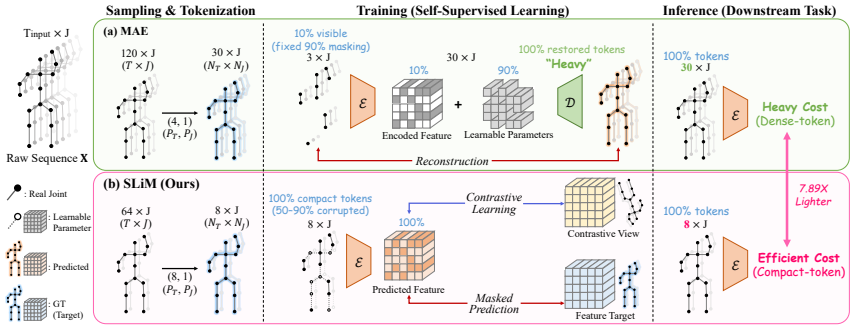


Figure 2: Conceptual comparison between coordinate-reconstruction MAE and SLiM. (a) Standard skeleton MAE relies on dense skeletal-temporal tokens ($30 \times J$) and a full-sequence decoder to reconstruct raw coordinates. Although the decoder is discarded after pre-training, downstream inference still requires dense encoder tokens. (b) SLiM replaces coordinate reconstruction with decoder-free masked feature prediction under a compact token grid ($8 \times J$), using stochastic mask-token corruption and contrastive learning during pre-training.

Despite their effectiveness, CL-based methods mainly optimize sequence-level alignment, which may miss fine-grained motion patterns and relies on semantically valid views.

Masked and predictive skeleton modeling. Masked modeling methods learn local skeletal-temporal dependencies by predicting missing skeleton information. SkeletonMAE [38] reconstructs masked joint coordinates, and MAMP [22] extends this objective to motion-aware trajectory prediction. S-JEPA [1] moves the prediction target from raw coordinates to latent embeddings, while GFP [31] introduces generative feature prediction to connect low-level reconstruction with semantic representation learning. These methods demonstrate the benefit of predictive objectives for skeleton SSL, but they commonly rely on dense skeletal-temporal tokenization, reconstruction or generation modules, and joint/patch-level masking strategies that may exploit strong local correlations among skeleton joints. While decoder-free predictive objectives have been explored in general SSL [2, 24, 42], SLiM focuses on their skeleton-specific adaptation under compact token budgets by combining masked feature prediction with structured masking and anatomically consistent augmentations.

Generative and multi-modal pretext tasks. Beyond standard CL and masked prediction objectives, recent methods have explored generative and multi-modal pretext tasks. MacDiff [37] formulates skeleton representation learning with masked conditional diffusion, while IGM [16] combines generative learning with spectral contrastive learning. HSP [35] uses a hierarchical semantic preservation framework to integrate 2D and 3D skeleton information. Although these approaches provide useful alternatives, they often require iterative generation, additional modalities, or multi-stream inputs such as bone and motion modalities. In contrast, SLiM focuses on a single 3D joint stream and studies compact-token skeleton representation learning through decoder-free masked feature prediction.

3 Method

3.1 Overview and Compact Tokenization

Fig. 3 illustrates **SLiM** (**S**keleton **L**ess **i**s **M**ore), a compact-token skeleton SSL framework that combines masked feature prediction and contrastive learning in a shared encoder. SLiM follows a teacher–student distillation scheme [33]. The student network $f_{\boldsymbol{\theta}}$ is optimized by back-propagation, while the teacher network $f_{\boldsymbol{\phi}}$ is updated by an EMA of the student weights as $\boldsymbol{\phi} \leftarrow m\boldsymbol{\phi} + (1-m)\boldsymbol{\theta}$, where m is a momentum coefficient. Both networks share the same architecture: an encoder g and two lightweight projection heads h_{MFP} and h_{CL}, for masked feature prediction and contrastive learning, respectively.

Encoder and tokenization. We employ a Vision Transformer (ViT) [7] as the backbone encoder g. Let the input skeleton clip be denoted as $\mathbf{X} \in \mathbb{R}^{T \times J \times C_{\mathrm{in}}}$, where T is the sampled clip length, J is the number of joints, and $C_{\mathrm{in}} = 3$ is the coordinate dimension. We reshape $\mathbf{X}$ into skeletal-temporal

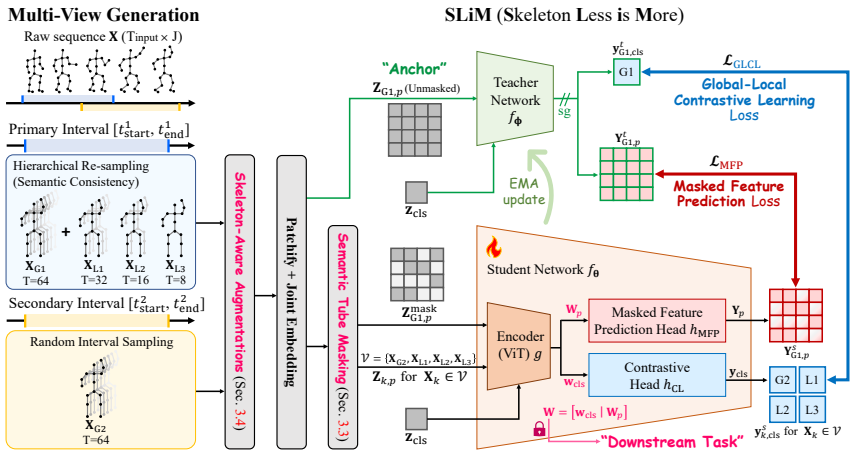


Figure 3: Overview of SLiM. SLiM combines decoder-free masked feature prediction (MFP) and global-local contrastive learning (GLCL) in a teacher–student framework. The student predicts teacher feature distributions from masked skeleton views and aligns global-local views for contrastive learning, using a shared compact-token encoder.

patches $\mathbf{X}_p \in \mathbb{R}^{N \times D_p}$, where

$$N = N_T N_J, \qquad N_T = \frac{T}{P_T}, \qquad N_J = \frac{J}{P_J}, \qquad D_p = P_T P_J C_{\text{in}}. \tag{1}$$

Here, (P_T, P_J) denotes the temporal and joint patch size, and $N_T \times N_J$ is the encoder token grid after patchification, not the raw input length. Each patch token is linearly projected by $\mathbf{E} \in \mathbb{R}^{D_p \times D}$ and combined with learnable skeletal positional embeddings $\mathbf{E}_{\text{skel}} \in \mathbb{R}^{N_J \times D}$:

$$\mathbf{Z}_p = \left[\mathbf{x}_p^1\mathbf{E} \mid \mathbf{x}_p^2\mathbf{E} \mid \cdots \mid \mathbf{x}_p^N\mathbf{E}\right] + \mathbf{E}_{\text{skel}}^b, \tag{2}$$

where $\mathbf{Z}_p \in \mathbb{R}^{N \times D}$ and $\mathbf{E}_{\text{skel}}^b \in \mathbb{R}^{N \times D}$ is the broadcast version of $\mathbf{E}_{\text{skel}}$ along the temporal dimension. We append a learnable class token $\mathbf{z}_{\texttt{cls}} \in \mathbb{R}^{1 \times D}$ to form $\mathbf{Z}_0 = [\mathbf{z}_{\texttt{cls}} \mid \mathbf{Z}_p]$. To model temporal dependencies, we adopt 1D temporal rotary positional embeddings (RoPE) [27]. Specifically, RoPE is applied to the query and key vectors according to the temporal token index before computing self-attention, while joint identity is encoded by the skeletal positional embedding. Given the initialized token sequence $\mathbf{Z}_0$, the encoder g produces the final token representations:

$$\mathbf{W} = g(\mathbf{Z}_0) = [\mathbf{w}_{\texttt{cls}} \mid \mathbf{W}_p],$$

where $\mathbf{W}$ is the final representation used for downstream evaluation.

Projection heads. Given the encoder output $\mathbf{W} = [\mathbf{w}_{\texttt{cls}} \mid \mathbf{W}_p]$, we use two projection heads for the global and patch-wise objectives. The contrastive head h_{CL} maps the class token to K-dimensional logits for global-local contrastive learning, while the masked feature prediction head h_{MFP} maps patch-wise features to K-dimensional logits for masked feature prediction:

$$\mathbf{y}_{\texttt{cls}} = h_{\text{CL}}(\mathbf{w}_{\texttt{cls}}) \in \mathbb{R}^{1 \times K}, \qquad \mathbf{Y}_p = h_{\text{MFP}}(\mathbf{W}_p) \in \mathbb{R}^{N \times K}, \tag{3}$$

where $\mathbf{y}_{\texttt{cls}}$ is used for the sequence-level contrastive objective, and $\mathbf{Y}_p$ is used for masked feature prediction at the patch level.

3.2 Decoder-Free Masked Feature Prediction with Contrastive Learning

Multi-view generation. To learn representations across different temporal granularities, we generate two global views $\mathbf{X}_{\text{G1}}$ and $\mathbf{X}_{\text{G2}}$ by sampling temporal intervals from the raw skeleton sequence. Given the primary global view $\mathbf{X}_{\text{G1}}$, we further generate local views $\mathbf{X}_{\text{L1}}$, $\mathbf{X}_{\text{L2}}$, and $\mathbf{X}_{\text{L3}}$ by re-sampling from the same temporal interval at reduced resolutions of 32, 16, and 8 frames. For each resolution, we sample two independent local crops, resulting in six local views in total: $\mathcal{V}_{\text{local}} =$

$\{\mathbf{X}^1_{\mathrm{L1}}, \mathbf{X}^2_{\mathrm{L1}}, \mathbf{X}^1_{\mathrm{L2}}, \mathbf{X}^2_{\mathrm{L2}}, \mathbf{X}^1_{\mathrm{L3}}, \mathbf{X}^2_{\mathrm{L3}}\}$. Anchoring local views to the same global interval reduces semantic misalignment between views and encourages temporal scale invariance. All views are processed by Skeleton-Aware Augmentations in Sec. 3.4. We define the following losses using $\mathbf{X}_{\mathrm{G1}}$ as the anchor and apply the same procedure symmetrically with $\mathbf{X}_{\mathrm{G2}}$ as the anchor.

Masked feature prediction (MFP). We formulate masked modeling as feature prediction rather than coordinate reconstruction. The teacher network receives the unmasked global view $\mathbf{X}_{\mathrm{G1}}$ and produces patch-level target features:

$$[\mathbf{y}^t_{\mathrm{G1},\texttt{cls}} \mid \mathbf{Y}^t_{\mathrm{G1},p}] = f_{\boldsymbol{\phi}}([\mathbf{z}_{\texttt{cls}} \mid \mathbf{Z}_{\mathrm{G1},p}])\,. \tag{4}$$

For the student network, a binary mask $\mathcal{M} \in \{0,1\}^N$ is applied to the same patch tokens:

$$\mathbf{Z}^{\texttt{mask}}_{\mathrm{G1},p} = (1-\mathcal{M}) \odot \mathbf{Z}_{\mathrm{G1},p} + \mathcal{M} \odot \mathbf{e}^b_{\texttt{[mask]}}, \tag{5}$$

where $\mathcal{M}_i = 0$ denotes a visible token, $\mathcal{M}_i = 1$ denotes a masked token, $\mathbf{e}_{\texttt{[mask]}} \in \mathbb{R}^D$ is a learnable mask token, and $\mathbf{e}^b_{\texttt{[mask]}} \in \mathbb{R}^{N\times D}$ is its broadcast version. The student network prediction is:

$$[\mathbf{y}^s_{\mathrm{G1},\texttt{cls}} \mid \mathbf{Y}^s_{\mathrm{G1},p}] = f_{\boldsymbol{\theta}}\big([\mathbf{z}_{\texttt{cls}} \mid \mathbf{Z}^{\texttt{mask}}_{\mathrm{G1},p}]\big)\,. \tag{6}$$

Let $\Omega = \{i \mid \mathcal{M}_i = 1\}$ be the set of masked patch tokens. For each masked token i, we obtain K-dimensional probability distributions by applying softmax to the teacher and student patch outputs:

$$\mathbf{p}^t_i = \mathrm{softmax}\left(\mathbf{Y}^t_i/\tau_t\right), \qquad \mathbf{p}^s_i = \mathrm{softmax}\left(\mathbf{Y}^s_i/\tau_s\right), \tag{7}$$

where τ_t and τ_s are temperature parameters. We adopt the teacher-student cross-entropy loss [42] and apply it to masked skeleton tokens:

$$\mathcal{L}_{\mathrm{MFP}} = -\frac{1}{|\Omega|}\sum_{i\in\Omega}\sum_{c=1}^{K} \mathrm{sg}\big(\mathbf{p}^t_i(c)\big)\log \mathbf{p}^s_i(c), \tag{8}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. Since MFP predicts teacher features rather than raw coordinates, SLiM removes the reconstruction decoder during pre-training. The downstream inference cost is then governed by the compact encoder token grid used for evaluation.

Global-local contrastive learning (GLCL). We use the teacher network's global class token from $\mathbf{X}_{\mathrm{G1}}$ as the anchor and align it with student network predictions from the other global and local views. Let $\mathcal{V} = \{X_{\mathrm{G2}}\} \cup \mathcal{V}_{\mathrm{local}}$. For each $\mathbf{X}_k \in \mathcal{V}$, the student produces:

$$[\mathbf{y}^s_{k,\texttt{cls}} \mid \mathbf{Y}^s_{k,p}] = f_{\boldsymbol{\theta}}([\mathbf{z}_{\texttt{cls}} \mid \mathbf{Z}_{k,p}])\,. \tag{9}$$

Note that local views are sampled from the same temporal interval as $\mathbf{X}_{\mathrm{G1}}$. The GLCL loss is:

$$\mathcal{L}_{\mathrm{GLCL}} = -\frac{1}{|\mathcal{V}|}\sum_{\mathbf{X}_k\in\mathcal{V}}\sum_{c=1}^{K} \mathrm{sg}\big(\mathbf{p}^t_{\mathrm{G1},\texttt{cls}}(c)\big)\log \mathbf{p}^s_{k,\texttt{cls}}(c), \tag{10}$$

where $\mathbf{p}^t_{\mathrm{G1},\texttt{cls}}$ and $\mathbf{p}^s_{k,\texttt{cls}}$ are the K-dimensional probability distributions obtained in the same manner as Eq. 7. This encourages temporal scale invariance by aligning local clips with the global semantic context provided by the teacher network.

Total objective. SLiM jointly optimizes masked feature prediction and global-local contrastive learning as $\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{MFP}} + \lambda\mathcal{L}_{\mathrm{GLCL}}$, where we empirically set $\lambda = 1.0$ for all experiments.

3.3 Semantic Tube Masking (STM)

Independent joint- or patch-level masking can encourage shortcut solutions, since skeleton joints are strongly correlated in space and time. As illustrated in Fig. 4(a), independently masking isolated joints can leave nearby spatial and temporal cues that allow local interpolation. We propose Semantic Tube Masking (STM), shown in Fig. 4(e), which masks connected anatomical regions, such as an arm, leg, or torso, over consecutive frames. Compared to isolated joint masking, STM occludes entire functional body parts across time, forcing the model to infer missing features from broader body context and motion dynamics.

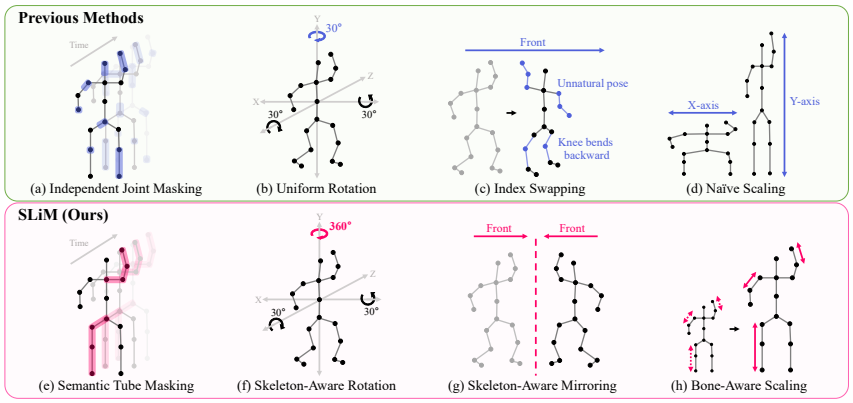


Figure 4: Comparison of masking and augmentation strategies. Top: independent joint masking and conventional geometric augmentations may allow local interpolation shortcuts or distort skeleton structure. Bottom: Semantic Tube Masking masks connected anatomical regions over time, while Skeleton-Aware Augmentations preserve basic anatomical consistency during view generation.

Skeletal-temporal tube design. We define semantic joint groups according to the human body structure, such as torso, arms, and legs. During mask generation, STM first selects an anatomical group and then samples a connected subset of joints within that group. The selected joints are masked over a consecutive temporal span, forming a skeletal-temporal tube. To balance masking difficulty across body parts, we adopt a constant-volume strategy: the temporal duration is adjusted inversely to the spatial size of the selected joint subset. Thus, smaller regions are masked for longer temporal spans, while larger regions are masked for shorter spans. This design reduces reliance on local interpolation and encourages prediction from broader skeletal-temporal context.

Dual role in training. STM is used in two ways. For MFP, it defines the masked patch tokens replaced by $\mathbf{e}_{\texttt{[mask]}}$ in Eq. 5. For GLCL, it is applied stochastically to local views as a structural augmentation, encouraging the representation to remain stable under partial anatomical occlusion.

3.4 Skeleton-Aware Augmentations (SAA)

Standard geometric augmentations [10, 15, 17] may disregard the articulated structure of the human body, leading to distorted skeletons or unreliable contrastive pairs, which can harm representation learning, as illustrated in Fig. 4(b–d). We introduce Skeleton-Aware Augmentations (SAA), shown in Fig. 4(f–h), a set of transformations designed to improve view diversity while preserving basic anatomical consistency.

Skeleton-aware rotation. This is illustrated in Fig. 4(f) and compared with the uniform rotation in Fig. 4(b). Standard rotation [4, 10, 17] is typically confined to a limited range of angles. To maintain a realistic upright posture, we apply a full 360° random rotation exclusively around the vertical Y-axis. For the remaining non-gravity axes (X, Z), we constrain perturbations to small angles (e.g., 30°) to avoid physically implausible tilting. Unlike previous works that restrict all axes to maintain stability, our approach decouples the rotation logic to maximize view diversity while preserving physical validity.

Skeleton-aware mirroring. As shown in Fig. 4(g), and compared with the index swapping in Fig. 4(c), heuristic index swapping [5, 10, 17] is a common approach but fails to achieve a mathematically rigorous reflection, thereby resulting in unnatural poses. Our strategy performs a geometric mirroring that simultaneously negates coordinate values along the lateral axis (X-axis) and reassigns joint indices according to bilateral symmetry. This dual-operation ensures that the resulting motion is anatomically consistent–preserving the correct "Front" orientation and left-right semantics–rather than producing a distorted artifact.

Bone-aware scaling. This is depicted in Fig. 4(h) and compared with the naive scaling in Fig. 4(d). The naive scaling of joint coordinates [4, 10] can disrupt the structure of the skeleton. To preserve the essence of the action, we decompose each bone into a directional unit vector $\hat{\mathbf{v}}$ and a scalar length l. We apply random scaling factors only to l while strictly maintaining $\hat{\mathbf{v}}$. This effectively simulates

Table 1: Linear evaluation on NTU-60, NTU-120, and PKU-MMD II. We report Top-1 accuracy (%) using the joint modality. **Bold** and underline indicate the best and second-best results, respectively.

Method	Publication	NTU-60		NTU-120		PKU-MMD II
		X-Sub	X-View	X-Sub	X-Set	X-Sub
Other skeleton SSL baselines:						
GL-Transformer [13]	ECCV'22	76.3	83.8	66.0	68.7	–
MacDiff [37]	ECCV'24	86.4	91.0	79.4	80.2	–
IGM [16]	ECCV'24	86.2	91.2	80.0	81.4	–
USDRL [36]	AAAI'25	85.2	91.7	76.6	78.1	54.4
HSP [35]	CVPR'25	80.7	88.0	71.0	73.2	48.9
Contrastive learning (CL):						
CPM [39]	ECCV'22	78.7	84.9	68.7	69.6	48.3
CMD [23]	ECCV'22	79.8	86.9	70.3	71.5	43.0
AimCLR [10]	AAAI'22	74.3	79.7	63.4	63.4	–
RVTCLR [44]	ICCV'23	74.7	79.1	–	–	–
PTSL [43]	AAAI'23	77.3	81.8	66.2	67.7	49.3
HYSP [9]	ICLR'23	78.2	82.6	61.8	64.6	–
HaLP [25]	CVPR'23	79.7	86.8	71.1	72.2	43.5
ActCLR [17]	CVPR'23	80.9	86.7	69.0	70.5	–
Masked / predictive modeling:						
SkeletonMAE [38]	ICMEW'23	74.8	77.7	72.5	73.5	36.1
MAMP [22]	ICCV'23	84.9	89.1	78.6	79.1	53.8
S-JEPA [1]	ECCV'24	85.3	89.8	79.6	79.9	53.5
GFP [31]	ICCV'25	85.9	92.0	79.1	80.3	56.2
AMR [29]	CVPR'26	<u>87.4</u>	<u>92.3</u>	<u>81.1</u>	<u>81.9</u>	**60.3**
Unified MFP + CL:						
SLiM (Ours)	–	**87.9**	**93.2**	**81.2**	**83.6**	<u>59.7</u>

the same action performed by subjects of varying body sizes and proportions, thereby enriching the intra-class variance without semantic distortion.

4 Experimental Results

4.1 Datasets and Protocols

We evaluate SLiM on three standard skeleton-based action recognition benchmarks: NTU RGB+D 60 (NTU-60) [26], NTU RGB+D 120 (NTU-120) [19], and PKU-MMD II [18]. NTU-60 contains 56,880 skeleton sequences from 60 action classes and provides Cross-Subject (X-Sub) and Cross-View (X-View) protocols. NTU-120 extends NTU-60 to 120 classes and 114,480 samples, with Cross-Subject (X-Sub) and Cross-Setup (X-Set) protocols. PKU-MMD II contains 51 action classes, and we follow the standard Cross-Subject (X-Sub) protocol. Unless otherwise specified, all experiments use only the 3D joint modality, without multi-stream fusion from bone or motion modalities.

4.2 Implementation Details

Architecture. Following previous MAE-based skeleton SSL methods [1, 22, 31, 38], we use a ViT [7] encoder with 8 Transformer layers, hidden dimension $D = 256$, and 8 attention heads. This matches the encoder depth and width of existing baselines, while SLiM uses a compact token grid. Specifically, global views are sampled to $T = 64$ frames with $J = 25$ joints, and patchified with $P_T = 8$ and $P_J = 1$, resulting in an encoder token grid of $N_T \times N_J = 8 \times 25$.

Pre-training setup. SLiM is pre-trained for 150 epochs with the teacher–student framework. The teacher momentum m is increased from 0.994 to 1.0 during training. We use AdamW [21] with a base learning rate of 2.0×10^{-4}, a 20-epoch linear warmup, and a cosine decay [20] to 1.0×10^{-6}. The model is trained with a total batch size of 768 on 4 NVIDIA RTX A6000 GPUs.

Masking and augmentations. For Semantic Tube Masking (STM), the target mask ratio is sampled uniformly from $[0.5, 0.9]$. For masked feature prediction, STM is applied to the primary global view by replacing selected patch tokens with learnable mask tokens. For global-local contrastive learning, STM is applied stochastically to local views as a structural perturbation. For Skeleton-Aware Augmentations (SAA), each transformation is applied independently with probability 0.5. Rotation

Table 2: Additional downstream evaluations on NTU-60. (a) Semi-supervised action recognition using 1% and 10% labeled data. (b) Action retrieval using k-NN with $k = 1$.

(a) Semi-supervised learning

Method	X-Sub		X-View	
	1%	10%	1%	10%
CPM [39]	56.7	73.0	57.5	77.1
CMD [23]	50.6	75.4	53.0	80.2
HYSP [9]	–	76.2	–	80.4
HaLP [25]	46.6	72.6	48.7	77.1
HiCo [6]	54.4	73.0	54.8	78.3
USDRL [36]	57.3	80.2	60.7	84.0
SkeletonMAE [38]	54.4	80.6	54.6	83.5
MAMP [22]	66.0	88.0	68.7	91.5
S-JEPA [1]	67.5	88.4	69.1	91.4
GFP [31]	71.8	88.7	72.9	**92.1**
SLiM (Ours)	**72.1**	**88.8**	**75.8**	91.9

(b) Action retrieval

Method	NTU-60	
	X-Sub	X-View
LongT-GAN [41]	39.1	48.1
P&C [28]	50.7	76.3
ISC [34]	62.5	82.6
HaLP [25]	65.8	83.6
HiCo [6]	68.3	84.8
SkeAttnCLR [12]	69.4	76.8
UmURL [30]	71.3	88.3
MAMP [22]	62.0	70.0
GFP [31]	70.9	87.1
SLiM (Ours)	**72.5**	**89.9**

Table 3: Token budget and computational cost comparison on NTU-60. $N_T \times N_J$ denotes the encoder token grid after patchification. SLiM reduces encoder inference cost by 7.89× compared to dense-token MAE-based methods.

Methods	Tokens	Inference (GFLOPs)	Training (GFLOPs)	
	$N_T \times N_J$	Encoder	Encoder	Decoder
SkeletonMAE [38]	30×25	28.32	1.97	17.70
MAMP [22]	30×25	28.32	1.97	17.70
S-JEPA [1]	30×25	28.32	1.97	17.70
GFP [31]	30×25	28.32	1.97	1.57
SLiM (Ours)	8×25	**3.59**	3.59	–

Table 4: Ablation studies for the objective functions on NTU-60. $\mathcal{L}_{\text{CL}}$ represents a standard CL loss without temporal diversity.

Objective			NTU-60	
$\mathcal{L}_{\text{MFP}}$	$\mathcal{L}_{\text{CL}}$	$\mathcal{L}_{\text{GLCL}}$	X-Sub	X-View
✗	✓	✗	73.6	78.9
✗	✗	✓	75.3	80.1
✓	✗	✗	85.3	90.6
✓	✓	✗	86.2	91.3
✓	✗	✓	87.7	92.7

uses a full 360° range around the vertical Y-axis and limits the non-gravity axes (X, Z) to ±30°. Bone-aware scaling samples length scales from $[0.85, 1.15]$ while preserving bone direction vectors.

4.3 Performance Comparison

We compare SLiM with SOTA skeleton SSL methods under three protocols: linear evaluation, semi-supervised learning, and action retrieval. To isolate representation quality, we report results using the single joint modality and exclude gains from multi-stream fusion such as bone or motion streams.

Linear evaluation. Table 1 reports Top-1 accuracy under the standard linear evaluation protocol, where the pre-trained encoder is frozen and only a linear classifier is trained. Including concurrent AMR [29], SLiM obtains the best result on four of five protocols, exceeding AMR by 0.5, 0.9, 0.1, and 1.7%-points on NTU-60 X-Sub, NTU-60 X-View, NTU-120 X-Sub, and NTU-120 X-Set, respectively. On PKU-MMD II, SLiM ranks second, 0.6%-points below AMR. Importantly, these results are obtained with a compact encoder token grid. As summarized in Fig. 1, coordinate-reconstruction MAE baselines degrade noticeably when the temporal token budget is reduced, whereas SLiM maintains strong accuracy under compact token budgets. Table 3 further provides the computational breakdown: SLiM uses an 8×25 encoder token grid and requires only 3.59 GFLOPs for downstream inference, yielding a 7.89× reduction compared to dense-token MAE-based methods with 30×25 tokens. During pre-training, SLiM also removes the decoder, reducing the computation spent on components that are discarded after pre-training.

Semi-supervised learning and retrieval. Table 2(a) evaluates the label efficiency of the learned representations. SLiM performs particularly well in the 1% label setting, improving the previous best by 0.3 and 2.9%-points on X-Sub and X-View, respectively, while remaining competitive in the 10% setting. Table 2(b) further evaluates frozen representations with non-parametric retrieval. SLiM achieves the best retrieval accuracy on both protocols, indicating that the learned features are discriminative even without training a task-specific classifier.

Table 5: Ablation studies on NTU-60. (a) Dual-role Semantic Tube Masking (STM) for the global masked-view branch and local contrastive-view branch. (b) Skeleton-Aware Augmentations (SAA), including rotation, mirroring, and scaling. Limited denotes the corresponding conventional masking or augmentation strategy.

(a) Semantic Tube Masking

Masking		NTU-60	
Global View	Local View	X-Sub	X-View
✗	✓	75.3	80.1
✓ (Limited)	✓	84.2	90.4
✓	✗	83.3	89.4
✓	✓ (Limited)	85.7	89.9
✓ (90%)	✓ (90%)	86.8	92.1
✓ (0–50%)	✓ (0–50%)	84.5	89.6
✓ (50–90%)	✓ (50–90%)	87.7	92.7

(b) Skeleton-Aware Augmentations

Rotation	Mirroring	Scaling	X-Sub	X-View
✗	✓	✓	81.6	86.9
✓ (Limited)	✓	✓	84.8	89.9
✓	✗	✓	85.3	90.6
✓	✓ (Limited)	✓	85.9	91.0
✓	✓	✗	84.4	90.0
✓	✓	✓ (Limited)	86.4	91.7
✓	✓	✓	87.7	92.7

4.4 Ablation Studies

We conduct ablation studies on NTU-60. For efficiency, all ablation models are pre-trained for 100 epochs, while the final model is pre-trained for 150 epochs. Thus, the ablation results may be slightly lower than the final model reported in Table 1. In the following analyses, "Limited" denotes replacing the proposed component with its conventional counterpart illustrated in Fig. 4(a–d).

Objective functions. Table 4 evaluates the contribution of masked feature prediction and contrastive learning. $\mathcal{L}_{\text{CL}}$-only and $\mathcal{L}_{\text{GLCL}}$-only variants show limited performance, indicating that discriminative losses alone are insufficient for learning detailed skeleton representations. $\mathcal{L}_{\text{MFP}}$ alone provides a strong baseline by learning local predictive features, achieving 85.3% on X-Sub. Adding a standard CL loss improves the result to 86.2%, while replacing it with $\mathcal{L}_{\text{GLCL}}$ further improves performance to 87.7%. This suggests that $\mathcal{L}_{\text{MFP}}$ and temporally diverse global-local contrastive learning are complementary: $\mathcal{L}_{\text{MFP}}$ captures local skeletal-temporal patterns, while $\mathcal{L}_{\text{GLCL}}$ improves sequence-level semantic discrimination across temporal granularities.

Semantic Tube Masking (STM). Table 5(a) evaluates STM's dual role in generating masked global views (MFP) and perturbing local views (GLCL). Replacing STM with conventional joint masking [1, 22, 31] in either role degrades X-Sub accuracy by up to 3.5%-point (87.7% vs. 84.2%), confirming that structured anatomical masking is crucial to prevent trivial interpolation shortcuts. Additionally, our stochastic masking strategy (50–90%) outperforms fixed extreme masking (90%) by 0.9%-point and low-ratio masking (0–50%) by 3.2%-point on X-Sub.

Skeleton-Aware Augmentations (SAA). Table 5(b) isolates the contributions of geometric augmentations. By substituting standard augmentations from prior works [10, 17] with our SAA, we strictly preserve the 3D geometric validity of contrastive pairs. This anatomical consistency directly translates to enhanced downstream accuracy. Consequently, compared to their conventional counterparts, our rotation, mirroring, and scaling yield substantial improvements.

5 Conclusion

We presented **SLiM**, a compact-token skeleton SSL framework that replaces coordinate reconstruction with decoder-free masked feature prediction and combines it with global-local contrastive learning. Semantic Tube Masking and Skeleton-Aware Augmentations help avoid interpolation shortcuts and provide reliable views for skeleton representation learning. Across linear evaluation, semi-supervised learning, and retrieval, SLiM achieves state-of-the-art performance while reducing encoder inference computation by 7.89× compared to dense-token MAE-based methods. Building upon these promising results, future work will explore extending our compact-token paradigm to more challenging scenarios, such as noisy 2D poses, missing joints, and heterogeneous skeleton topologies.

Acknowledgments

This work was supported by IITP grant funded by the Korea government (MSIT) (RS2022-00144444, Deep Learning Based Visual Representational Learning and Rendering of Static and Dynamic Scenes).

References

[1] Mohamed Abdelfattah and Alexandre Alahi. S-jepa: A joint embedding predictive architecture for skeletal action recognition. In *European Conference on Computer Vision*, pages 367–384. Springer, 2024.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

[4] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13359–13368, 2021.

[5] Jeonghyeok Do and Munchurl Kim. Skateformer: skeletal-temporal transformer for human action recognition. In *European Conference on Computer Vision*, pages 401–420. Springer, 2024.

[6] Jianfeng Dong, Shengkai Sun, Zhonglin Liu, Shujie Chen, Baolong Liu, and Xun Wang. Hierarchical contrast for unsupervised skeleton-based action representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 525–533, 2023.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] 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.

[9] Luca Franco, Paolo Mandica, Bharti Munjal, and Fabio Galasso. Hyperbolic self-paced learning for self-supervised skeleton-based action representations. *arXiv preprint arXiv:2303.06242*, 2023.

[10] Tianyu Guo, Hong Liu, Zhan Chen, Mengyuan Liu, Tao Wang, and Runwei Ding. Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 762–770, 2022.

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[12] Yilei Hua, Wenhan Wu, Ce Zheng, Aidong Lu, Mengyuan Liu, Chen Chen, and Shiqian Wu. Part aware contrastive learning for self-supervised action recognition. *arXiv preprint arXiv:2305.00666*, 2023.

[13] Boeun Kim, Hyung Jin Chang, Jungho Kim, and Jin Young Choi. Global-local motion transformer for unsupervised skeleton-based action learning. In *European conference on computer vision*, pages 209–225. Springer, 2022.

[14] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *International Journal of Computer Vision*, 130(5):1366–1401, 2022.

[15] Linguo Li, Minsi Wang, Bingbing Ni, Hang Wang, Jiancheng Yang, and Wenjun Zhang. 3d human action representation learning via cross-view consistency pursuit. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4741–4750, 2021.

[16] Lilang Lin, Lehong Wu, Jiahang Zhang, and Jiaying Liu. Idempotent unsupervised representation learning for skeleton-based action recognition. In *European Conference on Computer Vision*, pages 75–92. Springer, 2024.

[17] Lilang Lin, Jiahang Zhang, and Jiaying Liu. Actionlet-dependent contrastive learning for unsupervised skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2363–2372, 2023.

[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] 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.

[20] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[22] Yunyao Mao, Jiajun Deng, Wengang Zhou, Yao Fang, Wanli Ouyang, and Houqiang Li. Masked motion predictors are strong 3d action representation learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10181–10191, 2023.

[23] Yunyao Mao, Wengang Zhou, Zhenbo Lu, Jiajun Deng, and Houqiang Li. Cmd: Self-supervised 3d action representation learning with cross-modal mutual distillation. In *European Conference on Computer Vision*, pages 734–752. Springer, 2022.

[24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[25] Anshul Shah, Aniket Roy, Ketul Shah, Shlok Mishra, David Jacobs, Anoop Cherian, and Rama Chellappa. Halp: Hallucinating latent positives for skeleton-based self-supervised learning of actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18846–18856, 2023.

[26] 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.

[27] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[28] Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9631–9640, 2020.

[29] Shengkai Sun, Zhiyong Cheng, Zefan Zhang, Jianfeng Dong, Zhihui Li, and Meng Wang. Exploring adaptive masked reconstruction for self-supervised skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[30] Shengkai Sun, Daizong Liu, Jianfeng Dong, Xiaoye Qu, Junyu Gao, Xun Yang, Xun Wang, and Meng Wang. Unified multi-modal unsupervised representation learning for skeleton-based action understanding. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 2973–2984, 2023.

[31] Shengkai Sun, Zefan Zhang, Jianfeng Dong, Zhiyong Cheng, Xiaojun Chang, and Meng Wang. Towards efficient general feature prediction in masked skeleton modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12212–12221, 2025.

[32] Zehua Sun, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu. Human action recognition from various data modalities: A review. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3200–3225, 2022.

[33] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[34] Fida Mohammad Thoker, Hazel Doughty, and Cees GM Snoek. Skeleton-contrastive 3d action representation learning. In *Proceedings of the 29th ACM international conference on multimedia*, pages 1655–1663, 2021.

[35] Hongsong Wang, Xiaoyan Ma, Jidong Kuang, and Jie Gui. Heterogeneous skeleton-based action representation learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19154–19164, 2025.

[36] Wanjiang Weng, Hongsong Wang, Junbo Wang, Lei He, and Guo-Sen Xie. Usdrl: Unified skeleton-based dense representation learning with multi-grained feature decorrelation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 8332–8340, 2025.

[37] Lehong Wu, Lilang Lin, Jiahang Zhang, Yiyang Ma, and Jiaying Liu. Macdiff: Unified skeleton modeling with masked conditional diffusion. In *European Conference on Computer Vision*, pages 110–128. Springer, 2024.

[38] Wenhan Wu, Yilei Hua, Ce Zheng, Shiqian Wu, Chen Chen, and Aidong Lu. Skeletonmae: Spatial-temporal masked autoencoders for self-supervised skeleton action recognition. In *2023 IEEE international conference on multimedia and expo workshops (ICMEW)*, pages 224–229. IEEE, 2023.

[39] Haoyuan Zhang, Yonghong Hou, Wenjing Zhang, and Wanqing Li. Contrastive positive mining for unsupervised 3d action representation learning. In *European Conference on Computer Vision*, pages 36–51. Springer, 2022.

[40] Jiahang Zhang, Lilang Lin, Shuai Yang, and Jiaying Liu. Self-supervised skeleton-based action representation learning: A benchmark and beyond. *International Journal of Computer Vision*, 134(1):38, 2026.

[41] Nenggan Zheng, Jun Wen, Risheng Liu, Liangqu Long, Jianhua Dai, and Zhefeng Gong. Unsupervised representation learning with long-term dynamics for skeleton based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[42] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

[43] Yujie Zhou, Haodong Duan, Anyi Rao, Bing Su, and Jiaqi Wang. Self-supervised action representation learning from partial spatio-temporal skeleton sequences. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 3825–3833, 2023.

[44] Yisheng Zhu, Hu Han, Zhengtao Yu, and Guangcan Liu. Modeling the relative visual tempo for self-supervised skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13913–13922, 2023.

Appendix Overview

Table 6: Overview of appendix contents.

Section	Descriptions for Analysis and Discussion
Sec. A	**Implementation Details**: Architecture, experiment settings, and hyper-parameters
Sec. B	**Hierarchical Temporal Sampling**: Global/local view generation
Sec. C	**Semantic Tube Masking (STM)** (Algorithm 1)
Sec. D	**Skeleton-Aware Augmentations (SAA)**: Rotation (Algorithm 2), mirroring (Algorithm 3), and scaling (Algorithm 4)
Sec. E	**Additional Results and Discussion**: Token-Budget Study (Table 7), detailed cost comparison (Table 8), transfer learning (Table 9), and controlled prediction-target analysis (Table 10), missing/noisy-joint robustness and degraded pre-training (Tables 11–12)
Sec. F	**Limitations**
Sec. G	**Broader impact**
Sec. H	**Compute resources and existing assets**

This appendix provides additional implementation details and analysis for SLiM. Sec. A describes the architecture, projection heads, losses, and hyperparameters. Sec. B details the hierarchical temporal sampling strategy used for global and local view generation. Sec. C and Sec. D provide the algorithmic details of Semantic Tube Masking and Skeleton-Aware Augmentations, respectively. Sec. E reports additional token-budget, cost, transfer, controlled-target, and robustness analyses. Finally, Secs. F–H discuss limitations, broader impact, compute resources, and existing assets.

A Implementation Details

Detailed architecture. Our encoder g follows a ViT [7] backbone adapted to skeleton sequences, with 8 Transformer layers, hidden dimension $D = 256$, and 8 attention heads. The encoder is defined as:

$$\begin{aligned} \mathbf{Z}_0 &= [\mathbf{z}_{\texttt{cls}} \mid \mathbf{Z}_p], \\ \mathbf{Z}'_l &= \mathrm{MSA}_{\mathrm{RoPE}}(\mathrm{LN}\left(\mathbf{Z}_{l-1}\right)) + \mathbf{Z}_{l-1}, \\ \mathbf{Z}_l &= \mathrm{MLP}(\mathrm{LN}\left(\mathbf{Z}'_l\right)) + \mathbf{Z}'_l, \qquad l = 1, \ldots, L, \\ \mathbf{W} &= [\mathbf{w}_{\texttt{cls}} \mid \mathbf{W}_p] = \mathrm{LN}\left(\mathbf{Z}_L\right), \end{aligned} \tag{11}$$

where $\mathrm{MSA}_{\mathrm{RoPE}}$ denotes multi-head self-attention with temporal RoPE, LN denotes layer normalization, and $\mathbf{W}$ is the final representation used for downstream evaluation. Both the student network $f_{\boldsymbol{\theta}}$ and the teacher network $f_{\boldsymbol{\phi}}$ use the same encoder architecture and two 3-layer MLP projection heads, h_{CL} and h_{MFP}. Each head has a hidden dimension of 2,048 and a bottleneck dimension of 256, and outputs $K = 65{,}536$-dimensional prototype-assignment logits. The resulting probability distributions used in the MFP and GLCL losses are therefore defined over K prototypes. The total objective balances the masked feature prediction loss $\mathcal{L}_{\mathrm{MFP}}$ and the global-local contrastive loss $\mathcal{L}_{\mathrm{GLCL}}$ with $\lambda = 1.0$.

Settings for the token-budget study. For the token-budget study in Fig. 1, we vary the temporal token number N_T by adjusting the sampled clip length and/or temporal patch size while keeping the encoder depth and width fixed. Unless otherwise stated, all variants are trained with the same optimization settings as their corresponding baseline.

Downstream evaluation details. For linear evaluation, we freeze the pre-trained encoder and train a single linear classifier for 10 epochs using AdamW with learning rate 1×10^{-4}, and batch size

256. For semi-supervised evaluation, we fine-tune the full encoder using the standard 1% and 10% labeled splits following prior works [1, 22, 31]. For retrieval, we use ℓ_2-normalized features and cosine similarity with $k = 1$.

Scope of reported variability. Unless explicitly stated otherwise, the main results use one pre-training run. The uncertainty in Table 10 is the sample standard deviation from three independently trained linear probes on one fixed pre-trained checkpoint. The uncertainty in Tables 11–12 is the sample standard deviation over three independently sampled test-time corruptions. Neither quantity estimates pre-training-run variability.

B Hierarchical Temporal Sampling

SLiM uses hierarchical temporal sampling to construct semantically consistent global and local views at different temporal granularities. For each view, we first sample a temporal interval from the raw skeleton sequence and then uniformly sample frames within that interval. If the sampled physical duration is shorter than the target number of frames, we use linear temporal interpolation to resize the clip to the desired resolution.

Global view sampling. We generate two global views, $\mathbf{X}_{\mathrm{G1}}$ and $\mathbf{X}_{\mathrm{G2}}$, by sampling two temporal intervals from the raw sequence and resizing each to $T = 64$ frames. The interval duration is randomly sampled to cover 50% to 100% of the input sequence length: $0.5 < (t_{\mathrm{end}} - t_{\mathrm{start}})/T_{\mathrm{input}} < 1.0$. This allows each global view to capture the main action context while providing view diversity for contrastive learning.

Local view sampling. As illustrated in Fig. 3, local views $\mathbf{X}_{\mathrm{L1}}$, $\mathbf{X}_{\mathrm{L2}}$, and $\mathbf{X}_{\mathrm{L3}}$ are sampled from the same temporal interval $[t^1_{\mathrm{start}}, t^1_{\mathrm{end}}]$ as the primary global anchor $\mathbf{X}_{\mathrm{G1}}$. This anchoring reduces semantic misalignment between global and local views. For each resolution, we sample two independent crops:

- 32-frame views $\mathbf{X}_{\mathrm{L1}}$: crop duration ratio $p \in [0.35, 0.7]$.
- 16-frame views $\mathbf{X}_{\mathrm{L2}}$: crop duration ratio $p \in [0.15, 0.4]$.
- 8-frame views $\mathbf{X}_{\mathrm{L3}}$: crop duration ratio $p \in [0.05, 0.2]$.

C Semantic Tube Masking (STM)

Concept and motivation. Independent joint or patch masking can be solved by local interpolation because neighboring skeleton joints are highly correlated in space and time. Semantic Tube Masking (STM) reduces this shortcut by masking connected anatomical regions, such as arms, legs, or torso, over consecutive frames. This forms skeletal-temporal tubes and encourages the model to predict missing features from broader body context and temporal motion.

Anatomical joint grouping. For the standard 25-joint topology used by NTU-60 [26], NTU-120 [19], and PKU-MMD II [18], we divide the body into five semantic regions: trunk/head, left arm, right arm, left leg, and right leg. Within a sampled region, STM selects a graph-connected joint span rather than independent joints.

Target ratio, tube proposals, and realized coverage. For each corrupted view, we sample a target ratio $r \sim \mathcal{U}(0.5, 0.9)$ and set the budget to $B = \lfloor rN \rfloor$, where $N = N_T N_J$ and the default compact grid has $N = 8 \times 25 = 200$ tokens. We use $A_{\min} = 8$ and $A_{\max} = 0.5N$ for the proposal area. Each proposal samples an anatomical group, a connected joint span S of width $w = |S|$, an area A no larger than the remaining budget, and a temporal extent $h = \mathrm{clip}(\mathrm{round}(A/w), 1, N_T)$. Its temporal start is sampled uniformly. A proposal is accepted only when it adds at least one new cell without exceeding B; generation stops after ten consecutive proposals add no new cells. This acceptance rule means that the realized coverage approaches the sampled target from below. Across 11,520 generated masks, the realized ratio has mean 0.695, standard deviation 0.114, and range $[0.485, 0.895]$.

Mask-token and positional-encoding order. Let $\widetilde{\mathbf{Z}}_p$ denote the linearly projected patch content before the skeletal positional embedding is added. The implemented corrupted tokens expand Eq. 5 as

$$\mathbf{Z}_p^{\texttt{mask}} = (1 - \mathcal{M}) \odot \widetilde{\mathbf{Z}}_p + \mathcal{M} \odot \mathbf{e}^b_{\texttt{[mask]}} + \mathbf{E}^b_{\mathrm{skel}}. \tag{12}$$

Algorithm 1 Semantic Tube Masking (STM)

Input: target ratio r, grid size $N_T \times N_J$, semantic groups $\mathcal{G}$, proposal bounds $A_{\min}, A_{\max}$
$B \leftarrow \lfloor r N_T N_J \rfloor$; $\mathcal{M} \leftarrow \mathbf{0}_{N_T \times N_J}$; $C \leftarrow 0$; $F \leftarrow 0$
while $C < B$ **and** $F < 10$ **do**
  $P_{\max} \leftarrow \min(B - C, A_{\max})$
  if $P_{\max} < A_{\min}$ **then**
    break
  end if
  Sample $G \in \mathcal{G}$ and a graph-connected span $S \subseteq G$; $w \leftarrow |S|$
  Sample $A \sim \mathcal{U}(A_{\min}, P_{\max})$
  $h \leftarrow \mathrm{clip}(\mathrm{round}(A/w), 1, N_T)$
  Sample $t \sim \mathcal{U}\{0, \ldots, N_T - h\}$
  $\widetilde{\mathcal{M}} \leftarrow \mathcal{M}$; $\widetilde{\mathcal{M}}[t : t + h, S] \leftarrow 1$; $\widetilde{C} \leftarrow \sum_{i,j} \widetilde{\mathcal{M}}[i, j]$
  if $C < \widetilde{C} \leq B$ **then**
    $\mathcal{M} \leftarrow \widetilde{\mathcal{M}}$; $C \leftarrow \widetilde{C}$; $F \leftarrow 0$
  else
    $F \leftarrow F + 1$
  end if
end while
Output: Semantic Tube Mask $\mathcal{M}$

Thus, content replacement occurs before the skeletal positional embedding is added, and masked positions retain their joint identities. Temporal locations are subsequently supplied by RoPE inside self-attention.

Interpretation of the main-paper STM ablation. In the first four rows of Table 5(a), $\times$ denotes no masking, "Limited" denotes conventional independent joint masking, and ✓ denotes STM. In the last three rows, both the global and local views use STM; only the target-ratio policy changes among fixed $r = 0.9$, $r \sim \mathcal{U}(0, 0.5)$, and the proposed $r \sim \mathcal{U}(0.5, 0.9)$. These intervals specify target ratios rather than guaranteed realized coverage.

Dual-role application. STM is used for both MFP and GLCL. For the primary global view, STM is always applied to construct the corrupted input for masked feature prediction. For local views in the GLCL branch, STM is applied stochastically with probability $p = 0.5$ as a structural perturbation.

D Skeleton-Aware Augmentations (SAA)

As illustrated in Fig. 4, standard geometric augmentations can distort the articulated structure of the human body and produce unreliable contrastive pairs. Skeleton-Aware Augmentations (SAA) improve view diversity while preserving basic anatomical consistency. Each augmentation is applied independently with probability $p = 0.5$.

D.1 Skeleton-Aware Rotation

Uniform rotation strategies often restrict all axes to a narrow range to avoid unrealistic poses. We decouple vertical rotation from out-of-plane tilting. The vertical angle β is sampled from $\mathcal{U}(-\theta_{\mathrm{vert}}, \theta_{\mathrm{vert}})$ with $\theta_{\mathrm{vert}} = 180^\circ$, allowing full 360° rotation around the Y-axis. The tilt angles α and γ along the non-gravity axes (X, Z) are sampled from $\mathcal{U}(-\theta_{\mathrm{tilt}}, \theta_{\mathrm{tilt}})$ with $\theta_{\mathrm{tilt}} = 30^\circ$ to avoid excessive out-of-plane tilting.

D.2 Skeleton-Aware Mirroring

Index swapping alone changes left-right semantics but does not perform geometric reflection in coordinate space. We therefore combine coordinate reflection along the lateral X-axis with left-right joint reassignment using a predefined bilateral mapping $\mathcal{I}_{\mathrm{swap}}$. This keeps joint identity consistent with the mirrored geometry.

Algorithm 2 Skeleton-Aware Rotation

Input: Skeleton sequence $\mathbf{X} \in \mathbb{R}^{C\times T\times J}$, tilt angle limit θ_{tilt}, vertical rotation limit θ_{vert}.

Step 1. *Sample Rotation Angles:*

Sample X-axis tilt angle $\alpha \sim \mathcal{U}(-\theta_{\text{tilt}}, \theta_{\text{tilt}})$.

Sample Y-axis vertical angle $\beta \sim \mathcal{U}(-\theta_{\text{vert}}, \theta_{\text{vert}})$. ▷ Typically large, e.g., 180°

Sample Z-axis tilt angle $\gamma \sim \mathcal{U}(-\theta_{\text{tilt}}, \theta_{\text{tilt}})$.

Step 2. *Construct Rotation Matrix:*

Compute 3D rotation matrices $\mathbf{R}_x(\alpha)$, $\mathbf{R}_y(\beta)$, and $\mathbf{R}_z(\gamma)$.

Combine into a single rotation matrix: $\mathbf{R} = \mathbf{R}_z\mathbf{R}_y\mathbf{R}_x \in \mathbb{R}^{3\times 3}$.

Step 3. *Apply Rotation:*

Transform the coordinates for all joints:
$\mathbf{X}_{\text{rot}}[:, t, j] = \mathbf{R}\mathbf{X}[:, t, j]$ for all $t \in \{0, \cdots, T-1\}$ and $j \in \{0, \cdots, J-1\}$.

Output: Rotated sequence $\mathbf{X}_{\text{rot}}$

Algorithm 3 Skeleton-Aware Mirroring

Input: Skeleton sequence $\mathbf{X} \in \mathbb{R}^{C\times T\times J}$, predefined left-right joint swapping mapping $\mathcal{I}_{\text{swap}}$.

Step 1. *Joint Index Swapping:*

Swap the left and right anatomical joints based on the mapping:
$\mathbf{X}_{\text{swap}}[:, :, j] = \mathbf{X}[:, :, \mathcal{I}_{\text{swap}}(j)]$ for all $j \in \{0, \cdots, J-1\}$.

Step 2. *Spatial Reflection:*

Invert the coordinates along the X-axis (Horizontal mirroring):
$\mathbf{X}_{\text{mirror}}[0, :, :] = -\mathbf{X}_{\text{swap}}[0, :, :]$. ▷ Assuming channel 0 corresponds to the X-axis

Output: Mirrored sequence $\mathbf{X}_{\text{mirror}}$

D.3 Bone-Aware Scaling

Naive scaling of raw joint coordinates can distort local skeleton structure. We instead operate in bone-vector space. Each bone is decomposed into a direction and a length, and random scaling factors $s_G \sim \mathcal{U}(0.85, 1.15)$ are applied to bone lengths while preserving bone directions. Joint coordinates are then reconstructed in topological order from the root joint trajectory.

E Additional Results and Discussion

E.1 Token-Budget Study

Table 7 provides the detailed configurations used for the token-budget analysis in Fig. 1. For coordinate-reconstruction MAE baselines, reducing the temporal token budget from 30 to 8 leads to clear accuracy degradation. SkeletonMAE [38] drops from 76.25% to 70.50%, and MAMP [22] drops from 87.00% to 80.90%. This suggests that coordinate- or trajectory-level reconstruction objectives rely strongly on dense skeletal-temporal context. In contrast, SLiM remains stable across token budgets, achieving 90.55%, 90.75%, and 90.90% with 8, 15, and 30 temporal tokens, respectively. This shows that SLiM's complete design is compatible with compact skeletal-temporal tokenization. The controlled analysis in Sec. E.4 further shows that feature prediction alone does not explain this difference; the form and stability of the target matter. The additional SLiM-8 result with batch size 384 isolates the effect of training batch size under the same 8×25 token grid. Increasing the batch size to 768 improves the average accuracy from 89.60% to 90.55%, suggesting that SLiM benefits from the larger SSL batch in this configuration. This comparison isolates batch-size sensitivity, not aggregate pre-training cost.

Algorithm 4 Bone-Aware Scaling

Input: Skeleton sequence $\mathbf{X} \in \mathbb{R}^{C \times T \times J}$, parent joint mapping $\mathcal{P}$, semantic joint groups $\mathcal{G}$, scale range $[s_{\min}, s_{\max}]$.

Step 1. *Joint to Bone Conversion:*

The root bone is initialized to zero: $\mathbf{B}[:, :, 0] = \mathbf{0}$.

Extract bone vectors $\mathbf{B}$ from joints $\mathbf{X}$. For each joint $j \in \{1, \cdots, J-1\}$: $\mathbf{B}[:, :, j] = \mathbf{X}[:, :, j] - \mathbf{X}[:, :, \mathcal{P}(j)]$.

Step 2. *Generate Independent Scale Factors:*

Initialize scale factor matrix $\mathbf{S} \in \mathbb{R}^{J}$.

for each group $G \in \mathcal{G}$ **do**

  Sample a scale factor $s_G \sim \mathcal{U}(s_{\min}, s_{\max})$.

  Assign s_G to all joints in group G: $\mathbf{S}[j] = s_G$ for $j \in G$.

end for

Step 3. *Apply Bone Scaling:*

Scale each bone vector: $\mathbf{B}_{\text{scale}}[:, :, j] = \mathbf{B}[:, :, j] \odot \mathbf{S}[j]$. ▷ $\odot$ denotes element-wise multiplication

Step 4. *Reconstruct Joints:*

Initialize $\mathbf{X}_{\text{scale}}$ and preserve the original root joint trajectory: $\mathbf{X}_{\text{scale}}[:, :, 0] = \mathbf{X}[:, :, 0]$.

for each joint $j \in \{1, \cdots, J-1\}$ in topological order **do**

  Recover joint positions: $\mathbf{X}_{\text{scale}}[:, :, j] = \mathbf{X}_{\text{scale}}[:, :, \mathcal{P}(j)] + \mathbf{B}_{\text{scale}}[:, :, j]$

end for

Output: Scaled sequence $\mathbf{X}_{\text{scale}}$

E.2 Detailed Cost Comparison

Table 8 provides a module-level cost breakdown. Dense-token MAE-based methods feed only visible tokens to the pre-training encoder, but use a full-sequence decoder and, in some cases, an additional target module. SLiM removes the coordinate-reconstruction decoder; its target/auxiliary entry is the EMA-teacher forward on compact tokens. Because the methods use different view counts and training pipelines, these single-view numbers should not be read as aggregate pre-training-cost ratios. The controlled 7.89× claim concerns the downstream encoder, where all methods process their full unmasked token grid. In our implementation, SLiM is trained with a total batch size of 768 on 4 NVIDIA RTX A6000 GPUs.

E.3 Transfer Learning Evaluation

We further evaluate cross-dataset transferability on PKU-MMD II [18]. The encoder is first pre-trained on the source dataset using self-supervised learning, then frozen and evaluated on the target dataset with a linear classifier. This setting measures whether the learned representation transfers across datasets without end-to-end fine-tuning.

Table 9 reports the results when pre-training on the X-Sub splits of NTU-60 [26] and NTU-120 [19], followed by linear evaluation on PKU-MMD II X-Sub. SLiM achieves 72.7% and 75.3% accuracy when pre-trained on NTU-60 and NTU-120, respectively, outperforming prior skeleton SSL baselines under the same transfer setting. These results suggest that the features learned by masked feature prediction, STM, and GLCL transfer effectively across dataset distributions.

E.4 Controlled Analysis of Masked-Prediction Targets

The main paper compares the complementary MFP and GLCL losses. We additionally isolate the target used within MFP. All variants in Table 10 use the same 8×25 transferred encoder, GLCL

Table 7: Token-budget configurations and average accuracy on NTU-60. T denotes the sampled clip length, P_T denotes the temporal patch size, and $N_T = T/P_T$ is the number of temporal tokens. All settings use $J = 25$ joints and $P_J = 1$. Accuracy is averaged over the X-Sub and X-View protocols under linear evaluation. Inference GFLOPs are measured for the downstream encoder. SLiM-8 uses the default total batch size of 768 unless otherwise specified.

Method	**Clip** T	**Patch** P_T	**Tokens** $N_T \times N_J$	**Inference GFLOPs**	**Avg. Acc. (%)**
SkeletonMAE-8 [38]	64	8	8×25	3.59	70.50
SkeletonMAE-15 [38]	120	8	15×25	8.36	74.00
SkeletonMAE-30 [38]	120	4	30×25	28.32	76.25
MAMP-8 [22]	64	8	8×25	3.59	80.90
MAMP-15 [22]	120	8	15×25	8.36	85.50
MAMP-30 [22]	120	4	30×25	28.32	87.00
S-JEPA [1]	120	4	30×25	28.32	87.55
GFP [31]	120	4	30×25	28.32	88.95
SLiM-8 (Batch 384)	64	8	8×25	3.59	89.60
SLiM-8 (Batch 768)	64	8	8×25	3.59	90.55
SLiM-15	120	8	15×25	8.36	90.75
SLiM-30	120	4	30×25	28.32	90.90

Table 8: Detailed cost comparison on NTU-60. $N_T \times N_J$ denotes the encoder token grid after patchification. Inference GFLOPs are measured for the downstream encoder. Module-level GFLOPs are separated into student encoder, decoder, and target/auxiliary branches for one forward view. The target/auxiliary branch includes target generation networks for prior methods when used, and the EMA teacher forward for SLiM. These values do not aggregate SLiM's multi-view student/teacher forwards and therefore are not a total per-iteration pre-training comparison. Training resources are reported as implementation context and are not intended as a controlled hardware comparison.

Method	**Tokens**	**Inference GFLOPs**	**Single-view module GFLOPs**			**Reported Training Resource**	
	$N_T \times N_J$	Enc.	Enc.	Dec.	Target/Aux.	GPU Type	Batch Size
SkeletonMAE [38]	30×25	28.32	1.97	17.70	–	8×A6000	64
MAMP [22]	30×25	28.32	1.97	17.70	–	4×RTX3090	128
S-JEPA [1]	30×25	28.32	1.97	17.70	28.32	8×A100	256
GFP [31]	30×25	28.32	1.97	1.57	0.64	–	–
SLiM (Ours)	8×25	**3.59**	3.59	–	3.59	4×A6000	768

branch, masking, optimization schedule, and 100-epoch pre-training budget; only the MFP target, loss, and target-specific module change. Every target-specific module is discarded before linear evaluation.

Coordinate reconstruction and direct feature regression perform almost identically (86.96 vs. 86.92 average accuracy). Thus, removing the full-token coordinate decoder or replacing coordinates with features is not, by itself, the source of the gain. The direct feature-MSE branch also shows a degenerate training trajectory: its MFP loss decreases from 0.313 to 3.3×10^{-5}, while the effective rank of the masked-position projected features decreases to 1.12. For a diagnostic feature matrix with singular values $\{\sigma_j\}$, we compute the entropy-based effective rank as

$$r_{\text{eff}} = \exp\left(-\sum_j q_j \log q_j\right), \qquad q_j = \frac{\sigma_j}{\sum_k \sigma_k}. \tag{13}$$

An effective rank close to one indicates that the branch outputs occupy an almost one-dimensional subspace. This degeneration is specific to the MFP branch; GLCL continues to train the shared encoder, whose features remain non-degenerate. A predictor head stabilizes feature regression and improves it by 0.54/0.80 points on X-Sub/X-View, but remains 2.59/2.56 points below the prototype objective. We therefore attribute the result to the complete asymmetric prototype-assignment design—EMA teacher, stop-gradient, and clean/corrupted-view asymmetry—rather than to cross-entropy alone. Feature regression is not intrinsically unstable, as the predictor control also shows.

Table 9: Transfer learning performance on PKU-MMD II. We pre-train on the X-Sub splits of NTU-60 or NTU-120 and evaluate frozen representations on PKU-MMD II X-Sub with a linear classifier. **Bold** and underline indicate the best and second-best results, respectively.

Method	To PKU-II	
	NTU-60	NTU-120
LongT-GAN [41]	44.8	–
ISC [34]	51.1	52.3
CMD [23]	56.0	57.0
MacDiff [37]	<u>72.2</u>	73.4
IGM [16]	59.8	–
SkeletonMAE [38]	58.4	61.0
MAMP [22]	70.6	73.2
S-JEPA [1]	71.4	<u>74.2</u>
SLiM (Ours)	**72.7**	**75.3**

Table 10: Matched comparison of masked-prediction targets on NTU-60. The uncertainty shown for the proposed variant is the sample standard deviation over three independently trained linear probes on one fixed pre-trained checkpoint; it is not pre-training-run variability.

MFP target	Loss	Target-specific module	X-Sub	X-View	Avg.
Raw joint coordinates	MSE	5-block decoder	84.30	89.62	86.96
Teacher-projected features	MSE	None	84.53	89.32	86.92
Teacher-projected features	MSE	Predictor head	85.07	90.12	87.59
Teacher prototype distribution	Cross-entropy	None	87.66 ± 0.11	92.68 ± 0.09	**90.17**

E.5 Robustness to Missing and Noisy Joints

We distinguish *declared missingness*, where unavailable joints are routed through the learned mask token, from *unflagged corruption*, where filled coordinates are presented as valid input. We use the submitted 150-epoch encoder and its clean-trained linear classifier without corruption-specific training, tuning, or checkpoint selection. Corruption is applied only at test time. Selected joints are removed for every frame of a clip, preventing recovery by temporal interpolation. Reported uncertainty is the sample standard deviation over three independently sampled corruption seeds.

With eight declared missing joints, accuracy remains 87.23/92.14, approximately 0.6/1.1 points below the unrounded clean scores. Localization noise up to 5% of mean bone length also causes a modest decrease. In contrast, treating filled coordinates as valid leads to substantially larger degradation. X-View follows the same trend: parent-joint fill decreases from 86.02 to 71.01, and zero fill from 76.95 to 53.14, as the number of missing joints increases from three to eight. SLiM therefore provides a mechanism for *declared* missingness, but is not inherently robust to arbitrary incorrect coordinates. A deployment would need an upstream confidence or visibility signal that can be converted into the corresponding token mask; this conversion has not been evaluated end-to-end with a particular RGB pose estimator.

E.6 Corruption-Specific Degraded Pre-training

We further test whether joint degradation in the pre-training data can improve robustness. For each pre-training clip, five of the 24 non-reference joints are sampled once, zero-filled for all frames, and left unflagged. The reference joint used by standard NTU preprocessing is never removed. We additionally add Gaussian noise with $\sigma = 0.03$ times the sequence-level mean bone length. All other settings match the clean control. Because the degraded run reached 50 epochs during the response period, Table 12 compares matched 50-epoch checkpoints. Each encoder uses its own linear classifier trained and selected on clean data. The table reports NTU-60 X-Sub; each $\pm$ value is the sample standard deviation over three test-time corruption draws, and there is one pre-training run per condition.

Degraded pre-training substantially improves the matched unflagged zero-fill condition: with eight missing joints, accuracy increases from 54.73 to 75.92, and the drop relative to each model's own clean

Table 11: Test-time robustness on NTU-60. (a) Declared missing joints and localization noise, where σ is normalized by each sequence's mean bone length. (b) Unflagged missing joints on X-Sub. The same selected joints are absent throughout a clip. Each $\pm$ value is the sample standard deviation over three corruption draws, not pre-training variability.

(a) Declared missingness and noise

Condition	Severity	X-Sub	X-View
Clean	–	87.9	93.2
Declared missing	3/25	87.86 ± 0.06	92.94 ± 0.03
Declared missing	5/25	87.69 ± 0.12	92.68 ± 0.09
Declared missing	8/25	87.23 ± 0.01	92.14 ± 0.01
Gaussian noise	$\sigma = 0.01$	87.78 ± 0.04	93.18 ± 0.02
Gaussian noise	$\sigma = 0.03$	87.71 ± 0.05	93.01 ± 0.02
Gaussian noise	$\sigma = 0.05$	87.62 ± 0.02	92.84 ± 0.03

(b) Missing-joint representation (X-Sub)

Representation	Missing joints / 25		
	3	5	8
Declared mask token	87.86	87.69	87.23
Parent-joint fill	81.30 ± 0.29	76.41 ± 0.06	66.51 ± 0.07
Zero fill	72.20 ± 0.18	62.23 ± 0.21	49.63 ± 0.23

Table 12: Matched 50-epoch clean and degraded pre-training. Degraded pre-training uses persistent unflagged zero-fill of five non-reference joints plus Gaussian noise with $\sigma = 0.03$ times mean bone length.

Test condition	Clean pre-training	Degraded pre-training	Δ
Clean test	86.12	83.86	-2.26
Declared missing, 3/25	85.65 ± 0.07	82.48 ± 0.06	-3.17
Declared missing, 8/25	85.03 ± 0.12	80.95 ± 0.12	-4.08
Parent-joint fill, 3/25	80.40 ± 0.18	77.73 ± 0.10	-2.67
Parent-joint fill, 8/25	67.39 ± 0.37	67.01 ± 0.37	-0.38
Zero fill, 3/25	74.18 ± 0.12	79.75 ± 0.08	$+5.57$
Zero fill, 5/25	66.14 ± 0.28	78.21 ± 0.35	$+12.07$
Zero fill, 8/25	54.73 ± 0.06	75.92 ± 0.11	$+21.18$
Gaussian noise, $0.05\times$ bone length	85.83 ± 0.03	83.72 ± 0.07	-2.10

score decreases from 31.39 to 7.94 points. The improvement is corruption-specific. Clean accuracy decreases by 2.26 points on X-Sub and 2.24 points on X-View (91.22 to 88.98); declared missingness becomes worse, and the benefit does not transfer to parent-joint fill or stronger localization noise. We therefore present degraded pre-training as a robustness analysis, not as a default modification to SLiM. When missingness can be declared through confidence or visibility information, routing those positions through the learned mask token remains preferable.

F Limitations

SLiM targets efficient skeleton representation learning under a compact skeletal-temporal token budget. While this provides a favorable accuracy–efficiency trade-off on standard benchmarks, actions requiring subtle high-frequency motion or precise temporal ordering may benefit from denser temporal tokens. Our empirical scope is 3D joint sequences from three benchmarks with a fixed 25-joint topology. Adapting the manually defined STM groups to other topologies and evaluating motion capture, articulated meshes, body-worn sensors, or 2D poses from in-the-wild RGB videos remain future work; cross-modal generality is not established here.

The robustness analysis further distinguishes known missingness from arbitrary incorrect coordinates. Routing unavailable joints through the learned mask token works well, but requires a reliable upstream confidence or visibility signal. We have not evaluated confidence-to-mask conversion with a particular pose estimator, and unflagged parent- or zero-filled joints can cause severe degradation. Reproducing one unflagged corruption during pre-training improves that matched condition, but costs approximately 2.25 clean-accuracy points and does not generalize uniformly to other corruptions. Finally, the main results come from one pre-training run. The reported $7.89\times$ reduction concerns downstream encoder GFLOPs, not aggregate multi-view pre-training cost or measured latency; wall-clock behavior depends on hardware and implementation.

G Broader Impact

Efficient skeleton representation learning may benefit applications such as human-computer interaction, robotics, rehabilitation support, and assistive systems by reducing the computational cost of action understanding. Since skeletons are more compact than RGB videos, they may also reduce storage and computation in some privacy-sensitive settings. At the same time, action recognition technologies can raise privacy and surveillance concerns if deployed to monitor individuals without appropriate consent, transparency, or governance. Although this work uses public benchmark datasets and does not deploy a real-world system, practical deployment should follow privacy-preserving data collection practices, obtain appropriate consent, and evaluate potential biases caused by pose estimation quality, missing joints, camera viewpoints, or domain-specific sensor conditions.

H Compute Resources and Existing Assets

Compute resources. SLiM is pre-trained on 4 NVIDIA RTX A6000 GPUs with a total batch size of 768. The final model is pre-trained for 150 epochs, while the main ablation models are pre-trained for 100 epochs; the exploratory degraded-pre-training comparison uses matched 50-epoch checkpoints. We report single-view module GFLOPs and downstream encoder inference GFLOPs, but do not interpret the former as an aggregate multi-view pre-training comparison. Wall-clock training and inference time may vary depending on hardware, implementation, batch size, data loading, and mixed-precision settings.

Existing datasets. We use NTU RGB+D 60 [26], NTU RGB+D 120 [19], and PKU-MMD II [18] as public benchmark datasets for skeleton-based action recognition. We cite the original dataset papers and follow their official evaluation protocols. The datasets are not redistributed in this work; users should obtain them from the official sources and follow the corresponding terms of use specified by the dataset providers.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

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 state the main contributions of SLiM: compact-token masked feature prediction, integration with contrastive learning, Semantic Tube Masking, Skeleton-Aware Augmentations, and reduced encoder inference cost. The claims are supported by the experimental results in Sec. 4 and are scoped to skeleton-based representation learning with 3D joint inputs.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The supplementary material includes a Limitations section discussing the trade-off of compact temporal tokenization, the reliance on reasonably reliable 3D skeleton inputs, and the distinction between GFLOPs-based efficiency and actual wall-clock speed. It

also notes that noisy 2D pose estimates, missing joints, or very fine-grained temporal actions may require additional handling.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present new theoretical results, theorems, or formal proofs. The mathematical formulations are used to define the model, losses, and tokenization procedure.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

Justification: The paper specifies the datasets, evaluation protocols, input modality, architecture, tokenization, losses, optimizer, learning rate schedule, batch size, masking policy, and augmentation settings in Sec. 3 and Sec. 4. These details are sufficient to reproduce the main experimental setup.

Guidelines:

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

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

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

Answer: [No]

Justification: The experiments are conducted on publicly available benchmark datasets, and the paper provides implementation details for reproducibility. We plan to release the code and pre-trained weights upon publication, but an anonymized public code release is not provided at submission time.

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: Sec. 4.1 specifies the datasets and protocols, and Sec. 4.2 provides the architecture, optimizer, learning rate schedule, batch size, pre-training epochs, masking ratios, and augmentation settings. The evaluation protocols for linear evaluation, semi-supervised learning, and retrieval are also described in Sec. 4.

   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: [No]

   Justification: The main results follow the standard reporting protocol in prior skeleton SSL works and are reported as single-run Top-1 accuracy. We do not report error bars or confidence intervals because full multi-seed pre-training across all benchmarks is computationally expensive.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper reports the GPU type and training batch size used for the proposed method, and provides encoder GFLOPs for inference comparison. Detailed wall-clock runtime may vary depending on implementation and hardware, so we focus on GFLOPs for controlled efficiency analysis.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work uses publicly available benchmark datasets and does not involve collecting new personal data, human-subject experiments, or high-risk deployment. We have reviewed the NeurIPS Code of Ethics and do not identify any violation.

   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 supplementary material discusses potential positive impacts of efficient skeleton representation learning, such as applications in human-computer interaction, robotics, rehabilitation support, and assistive systems. It also discusses potential negative impacts, including privacy and surveillance concerns if action recognition systems are deployed without appropriate consent, transparency, or governance.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate

to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not introduce scraped datasets, generative models, large language models, or other assets with high misuse risk. The work focuses on skeleton representation learning using established public benchmarks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The paper cites the original dataset and baseline method papers, and the supplementary material describes the use of existing public benchmark datasets and published baselines. We follow the official evaluation protocols and terms of use for the datasets, and we do not redistribute third-party datasets, code, checkpoints, or derived assets.

    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: [No]

    Justification: The paper introduces a new method and plans to release code and pre-trained weights upon publication, but these new assets are not provided as an anonymized public release at submission time. The method and training details are documented in the paper to support reproducibility.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or new human-subject data collection. It uses existing public skeleton action recognition benchmarks.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not conduct new human-subject experiments and does not collect new participant data. All experiments are performed on previously released public benchmark datasets.

    Guidelines:

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

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development and experiments do not involve LLMs as an important, original, or non-standard component. Any use of writing or editing tools does not affect the scientific method, experimental design, or originality of the research.

    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.