Supplementary Material for From Isolated Islands to *Pangea*: Unifying Semantic Space for Human Action Understanding

Yong-Lu Li*, Xiaoqian Wu*, Xinpeng Liu, Zehao Wang, Yiming Dou, Yikun Ji, Junyi Zhang, Yixing Li, Xudong Lu, Jingru Tan, Cewu Lu†
Shanghai Jiao Tong University
{yonglu_li, enlighten, davidwang200099, douyiming, junyizhang, lyxing0, luxudong2001, lucewu}@sjtu.edu.cn, {xinpengliu0907, jiyikun2002, tanjingru120}@gmail.com

We report more details and discussions here:

1. Supplementary Related Works

1.1. Hyperbolic Representation

Hyperbolic representation has emerged in deep learning to encode hierarchical tree-like structure and taxonomy [14, 47, 71]. It has been applied in computer vision for hierarchical action search [39], video action prediction [69], and hierarchical image classification [12, 27, 38]. Long *et al*. [39] project video and action embeddings in the hyperbolic space and train a cross-modal model to perform hierarchical action search. In this work, we use hyperbolic embeddings to encode the hierarchical geometry of our structured semantic space.

1.2. Visual-Language Learning

Visual-language learning recently shows potential in learning generic representations [22, 58, 65, 74, 77]. Specifically, CLIP [58] and ALIGN [22] benefit from web-scale curated image-text pairs for training and allow zero-shot transfer to many downstream tasks. Following works [35, 73] adapt CLIP to video recognition via prompting, temporal modeling, *etc*. However, it may be hard for their implicit language embedding to capture the subtle taxonomy and structure knowledge of action semantics. Thus, we propose to solve the problem via a structured semantic space.

1.3. 3D Human Representation

3D Human Representation has been attracting much attention for a long time. A most intuitive representation is the 3D human pose, and lots of effort has been put into single-view 3D pose reconstruction [31, 45, 49, 67]. Some methods [49, 67] directly regress 3D pose from the given image. With great progress given in 2D pose estimation, many works [31, 45] adopt pre-detected 2D poses as auxiliary inputs. DensePose [19] proposes to adopt a UV map to represent the dense correspondence between the image and a human mesh, which could function as a 2.5D human representation. Lately, different parametric human body models (like SMPL [41] and SMPL-X [52]) are proposed as promising human representations. Impressive performance has been achieved with weak supervision, like 2D pose [8, 25, 41, 52, 68], semantic segmentation, motion dynamics, and so on. Also, different paradigms are proposed. Some works [40, 50] directly fit the parametric model to the weak supervision signals, which is accurate but sensitive to the initial state, and the speed is restricted. While there are also regression methods [8, 25, 68] learning a neural network to map images to human model parameters, greatly accelerating the reconstruction but losing accuracy. Combining the advantages of both kinds of methods, SPIN [29] and EFT [24] proposed to adopt regression methods for initialization and then use fitting methods for refinement. Inspired by the recent progress in NeRF [46], HumanNeRF [75] proposed a neural radiance field representation for free-view dynamic human modeling.

1.4. 3D Action Generation

3D Action Generation is an active field. With large skeleton datasets such as NTU [37] and Human3.6M [21], considerable efforts have been put on it [20, 53, 54, 70, 78]. Mean-

*The first two authors contribute equally.
†Corresponding author.

while, MoCap datasets [20, 44] push it further towards parametric human model-based generation [53]. Most efforts are either unconditional or conditioned on restricted action classes. Beyond class conditioned generation, some works conduct generation with natural language [54, 70] based on datasets composed of motion-text pairs [55, 56].

2. Details of *Pangea* Database

2.1. Data Curation

With the structured semantic space, We can collect data with diverse modalities, formats, and granularities, and adapt them into a unified form. Our database *Pangea* contains a large range of data including images, videos, and skeletons/MoCaps. We give more details of the processing and formulation as follows:

1) Semantic consistency. The class definitions of datasets are various, but they can be mapped to our semantic space with the fewest semantic damages. As mentioned in the main text, the mapping is completed via manual annotation with the help of word embedding [58] distances and OpenAI GPT-3.5. Manual annotation is the most accurate and most expensive, while word embedding comparison is the least. Thus, we adopt a hybrid method: potential class-node mapping is first filtered out roughly by comparing word embedding, then selected via GPT-3.5 prompting, and finally checked by human annotators. As more and more classes are aligned and covered, the process would be faster and faster with synonyms checking.

Suppl. Fig. 1 shows a flow chart of the action semantic mapping by human annotators. We invite 60 annotators of different backgrounds. Each candidate class is annotated three times, generating the final labels via the majority rule. Finally, for the 898 verb nodes (including 575 leaf nodes), there are a total of 515 verb nodes that have corresponding retargeted classes (including 290 leaf nodes). The missing verb nodes are mostly related to visually unrecognizable semantics, *e.g.*, `invest`.

2) Temporal consistency. Some videos [4] only have *sparse* labels for a whole clip instead of each frame. To solve this conflict, we sample the clip with *3 FPS* and give them the label of their belonged clip describing the action in the clip. More dense or spare sampling is either computationally costly or with serious information loss. On the contrary, with *dense* frame labels [18], we can easily get the clip label via fusing frame labels. Thus, we provide both frame- and clip-level labels for videos.

3) Spatial consistency. There are both instance (boxes) [6] and image [5] level labels. It is too expensive to annotate all missing human boxes and actions to make the whole *Pangea* instance-level. More realistically, we merge the instance labels of each image/frame into image/frame labels. In the future, we can also add more box labels to existing images based on the existing instance labels to support larger-scale instance-level training.

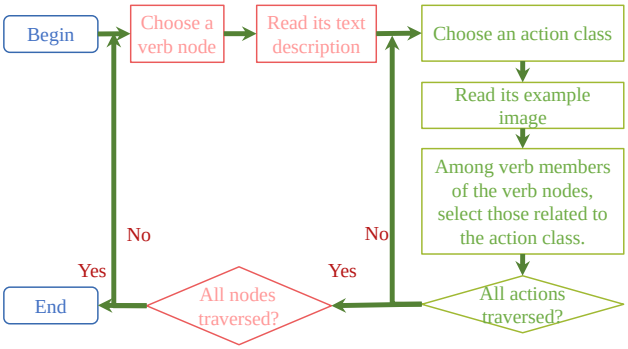


Figure 1. The flow chart of the action semantic mapping by human annotators.

4) 3D format consistency. 3D action datasets typically have different formats, *e.g.*, SMPL [40] contains 24 key-points while CMU MoCap [20] has 31 key-points. To keep format consistency, we transform all of them into SMPL via a fitting procedure.

5) 2D-3D consistency. Image/video datasets mostly contains only 2D labels without 3D human labels. We generate 3D humans via single-view reconstruction [68]. Please refer to Suppl. Sec. 2.2 for more details.

2.2. 3D Human Body Annotation Details

We adopt 3D humans for multiple reasons. First, 3D human provides a robust representation without *viewpoint* problems. Second, 3D humans can be seen as the safest choice as the physical carrier of actions with no need to consider the *domain gap* across image conditions.

In *Pangea*, we also prepare pseudo 3D human labels for images/videos. Different strategies are adopted depending on the label circumstances of the data. For different scenarios with ground truth (GT) 2D or 3D human poses, human boxes only, and no human instance labels at all, we adopt different strategies as follows:

1. If an image has a 3D human pose annotation, we fit the SMPL model to the 3D pose and associate the fitted 3D human body with the annotation. The 2D body is acquired by cropping the image with the bounding box.
2. If an image has a 2D human pose annotation, we calculate the MSE error of the annotated pose and the reprojected pose from 3D recovering and associate the annotated human instance with the reconstructed 3D body whose MSE error is the lowest among all and lower than a threshold. The 2D body representation is acquired by cropping the image with the box.
3. If an image has a human bounding box annotation, we calculate the IoU between the annotated box and the reprojected human mesh bounding box. Then, the anno-

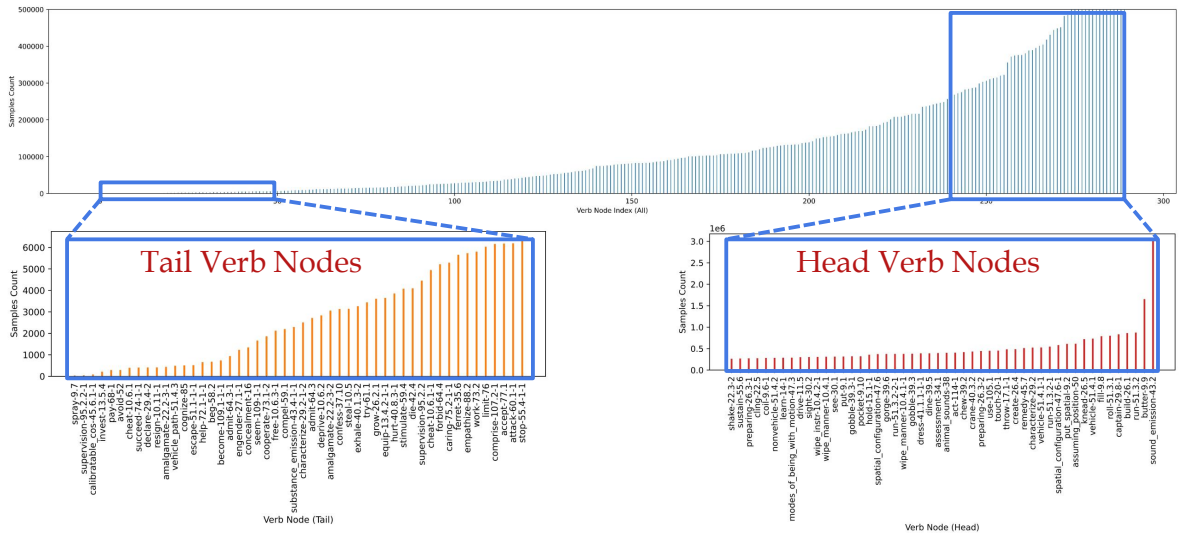


Figure 2. Semantic distribution of samples on 290 leave nodes, including detailed statistics on tail/head verb nodes.

tated human box is associated with the 3D human body whose IoU is the highest and higher than a threshold. The 2D body representation is acquired by cropping the image with the bounding box.

4. If an image contains no human annotation, OpenPose [3] is adopted to generate a pseudo annotation for the 2D human pose. Then we follow the same association strategy as images with 2D pose annotation. We assume the human instance with the lowest MSE error is the target human performing the annotated action.
5. For mesh sequences, we directly adopt them as 3D humans. Besides, for skeleton sequences without a 2D image available, we align the annotations with joints defined by SMPL and extract the 3D human body by fitting the SMPL model to the aligned pose.

Note that the 3D human pose and the corresponding/reprojected 2D pose could be easily extracted simultaneously. Images/frames with no human bodies or failure reconstructions were dropped. In practice, ROMP [68] and EFT [23] are adopted to directly recover humans from images.

2.3. More Statistics of *Pangea*

We list the collected datasets of *Pangea* in Suppl. Tab. 1.

2.4. Justifications of Database Design Choices

The class-node mapping is selected via **GPT-3.5 prompting**. We choose GPT-3.5 because of its excellent instruction-following abilities and easy-to-use API. In future work, we will extend our work with more powerful LLMs (*e.g.*, GPT-4) or locally deployed LLMs (*e.g.*, Llama 2).

SMPL. We choose SMPL mainly for its versatility and expressiveness since most data either provide SMPL parameters or could be conveniently converted into SMPL format.

ROMP & EFT. We find ROMP with EFT optimization managing to process our massive data efficiently with promising reconstruction quality. We would keep refining the data quality with the progress in 3D human reconstruction on our open-sourced website.

2.5. Semantic Distribution of *Pangea*

Suppl. Fig. 2 shows the sample count for 290 leaf verb nodes of our *Pangea* database. Detailed statistics on tail/head verb nodes are also listed.

2.6. Data License/Address

All the data of *Pangea* are from open-sourced datasets and for research purposes only. We give the data licenses and links of the gathered datasets here.

- Willow Action: https://www.di.ens.fr/willow/research/stillactions/
- Phrasal Recognition: https://vision.cs.uiuc.edu/phrasal/
- Stanford 40 Action: http://vision.stanford.edu/Datasets/40actions.html
- MPII: http://human-pose.mpi-inf.mpg.de/
- HICO: http://www-personal.umich.edu/~ywchao/hico/
- V-COCO: https://www.v7labs.com/open-datasets/v-coco
- HAKE: http://hake-mvig.cn/download/
- HMDB51: https://creativecommons.org/licenses/by/4.0
- HAA500: https://www.cse.ust.hk/haa/LICENSE

		Action Classes	Images/Frames	Videos
Image	Willow Action [10]	7	1 K	-
	Phrasal Recognition [61]	10	4 K	-
	Stanford 40 Actions [76]	40	4 K	-
	MPII [2]	410	4 K	-
	HICO [5]	600	38 K	-
	HAKE [32]	156	42 K	-
Video	HMDB51 [30]	51	69 K	7 K
	HAA500 [9]	500	64 K	8.5 K
	AVA [18]	80	162 K	0.5 K
	YouTube Action [36]	11	4 K	1 K
	ASLAN [28]	432	18 K	1 K
	UCF101 [66]	101	61 K	13 K
	Olympic Sports [48]	16	6 K	1 K
	Penn Action [79]	15	85 K	2 K
	Charades [63]	157	44 K	8 K
	Charades-Ego [64]	157	235 K	8 K
	ActivityNet [13]	200	2,444 K	20 K
	HACS [80]	200	1,379 K	504 K
	Home Action Genome [59]	453	702 K	6 K
	Kinetics [4]	700	14,132 K	536 K
Image+Video	Pangea	**4,296**	**19,495 K**	**1,116 K**
Skeleton/MoCap	HumanAct12 [20]	12	90 K	1 K
	CMU MoCap [20]	8	978 K	1 K
	UTD-MHAD [7]	27	90 K	1 K
	NTU RGB+D [62]	120	830 K	114 K
	Human3.6M [21]	17	3,600 K	<1 K
	BABEL [56]	260	4,050K	10K
	HAA4D [72]	300	212K	3K
Total	Pangea	**5,040**	**29,345 K**	**1,247 K**

Table 1. Statistics of collected and curated multi-modal datasets. Note that different datasets may share part of action classes (*e.g.*, ActivityNet [13] and HACS [80]).

- AVA: https://creativecommons.org/licenses/by/4.0
- Youtube Action: http://www.cs.ucf.edu/~liujg/YouTube_Action_dataset.html
- ASLAN: https://talhassner.github.io/home/projects/ASLAN/ASLAN-main.html
- UCF101: https://www.crcv.ucf.edu/data/UCF101.php
- Olympic Sports: http://vision.stanford.edu/Datasets/OlympicSports/
- Penn Action: http://dreamdragon.github.io/PennAction/
- Charades: http://vuchallenge.org/license-charades.txt
- Charades-Ego: https://prior.allenai.org/projects/data/charades-ego/license.txt
- ActivityNet: http://activity-net.org/download.html
- HACS: http://hacs.csail.mit.edu/
- Home Action Genome: https://homeactiongenome.org/index.html#what-we-do
- Kinetics: https://creativecommons.org/licenses/by/4.0
- HumanAct12: https://github.com/EricGuo5513/action-to-motion
- CMU MoCap: http://mocap.cs.cmu.edu/
- UTD-MHAD: https://personal.utdallas.edu/~kehtar/UTD-MHAD.html
- NTU RGB+D: https://rose1.ntu.edu.sg/dataset/actionRecognition/
- Human3.6M: http://vision.imar.ro/human3.6m/eula.php
- BABEL: https://babel.is.tue.mpg.de/license.html
- HAA4D: https://cse.hkust.edu.hk/haa4d/

3. Details of P2S

3.1. Label Augmentation Details

We detail the label augmentation here. Each image has a partial annotation $Y = \{y_i | y_i = 1, 0, \emptyset\}_{i=1}^{N}$, where $1, 0$ are certain positive/negative labels, and $\emptyset$ are uncertain ones.

A direct way to solve the uncertain labels is *assuming negative*: unobserved labels are considered as negatives. That is, for $\forall i$, if $y_i = \emptyset$, assign $y_i = 0$. However, some positive labels are falsely treated negatively, which hinders semantic learning, especially for few-shot nodes. There-

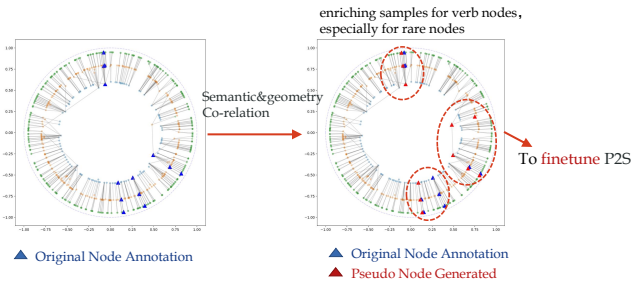


Figure 3. Illustration of label augmentation. Pseudo labels are generated based on VerbNet semantic/geometry co-relation. With generated pseudo labels, we can finetune P2S with more samples, which especially benefits verb nodes with rare samples.

fore, we propose to generate pseudo labels for uncertain labels, instead of simply treating them as negatives. That is, if $y_i = \emptyset$, assign $y_i = y_i^{'} \in [0, 1]$. The pseudo label $y_i^{'}$ is generated based on the structure and language prior to our semantic space. The pre-defined geometry and semantic information in VerbNet indicate the co-relation between verb nodes. Based on the co-relation, high-quality nodes with more samples can transfer knowledge (positive/negative labels) to low-quality nodes with fewer samples, thus generating pseudo labels to apply label augmentation and facilitate P2S learning. The process is illustrated in Suppl. Fig. 3.

In the implementation, we first obtain a co-relation matrix $\mathbf{C} = \{c_{ij}\}_{N\times N}$ of N verb nodes via language priors and VerbNet structure. Then pseudo labels are generated based on $\mathbf{C}$ and certain labels. That is, for each i where $y_i = \emptyset$, we assign $y_i^{'} = \sum_{j:y_j=1, j\neq i} c_{ij} y_j$.

The co-relation matrix $\mathbf{C}$ is calculated from two components: 1) C_L based on language priors; 2) C_E based on VerbNet structure. For C_L, we encode the semantic information of each verb node into l_i via a pretrained text encoder [16] and then construct $C_L = cos(l_i, l_j)$, where $cos(\cdot, \cdot)$ measures the cosine similarity of two vectors. For C_E, based on the trained hyperbolic embeddings $E = \{e_i\}_{i=1}^N$, we obtain $C_E = -d_{\mathcal{L}}(e_i, e_j)$, where $d_{\mathcal{L}}(\cdot, \cdot)$ is the Lorentzian Distance (detailed in Suppl. Sec. 3.2). Finally, we normalize both C_L and C_E into $[0, 1]$ and obtain C via $C = (C_L + C_E)/2$.

With label augmentation, the long-tail distribution is effectively alleviated with credible pseudo labels. The sample distribution before/after generating pseudo labels is shown in Suppl. Fig. 4. We can find that many tail nodes have more samples after the augmentation which alleviates the long-tailed distribution a lot. To benefit from label augmentation, we train P2S mapping in two phases. In phase 1, the whole model is trained via *assuming negative*. In phase 2, we finetune the model with certain labels and pseudo labels. Phase 2 benefits from the eased long-tail distribution, thus facilitating P2S learning.

Another consideration is to bind prediction with **soft** or **hard** pseudo labels. For soft labels, we directly use the pseudo label $y_i^{'} \in [0, 1]$ as ground truth. For hard labels, we consider only pseudo labels above the given threshold and use $y_i^{'} \in \{0, 1\}$ as ground truth. We find hard labels drag the performance a little, possibly because of the amplified noise of generated pseudo labels. Thus, we adopt soft labels in practice.

3.2. Lorentz Model for Verb Hierarchy

Preliminaries [11]. Lorentz model represents a hyperbolic space of n dimensions on the upper half of a two-sheeted hyperboloid in $\mathbf{R}^{n+1}$. We refer to the hyperboloid's axis of symmetry as time dimension and all other axes as space dimensions [11]. Every vector $\mathbf{x} \in \mathbf{R}^{n+1}$ can be written as $[\mathbf{x}_{\text{space}}, \mathbf{x}_{\text{time}}]$, where $\mathbf{x}_{\text{space}} \in \mathbf{R}^n$ and $\mathbf{x}_{\text{time}} \in \mathbf{R}$.

Let $\langle \cdot, \cdot \rangle$ be Euclidean inner product and $(\cdot, \cdot)_{\mathcal{L}}$ denote the *Lorentzian inner product* that is induced by the Riemannian metric of the Lorentz model. For two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n+1}$, it is computed as follows:

$$(\mathbf{x}, \mathbf{y})_{\mathcal{L}} = \langle \mathbf{x}_{\text{space}}, \mathbf{y}_{\text{space}} \rangle - \mathbf{x}_{\text{time}} \mathbf{y}_{\text{time}}. \tag{1}$$

The induced *Lorentzian norm* is $\|x\|_{\mathcal{L}} = \sqrt{(\mathbf{x}, \mathbf{x})_{\mathcal{L}}}$. The Lorentz model possessing a constant curvature $-c$ is defined as the following set of vectors:

$$\mathcal{L}^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : (\mathbf{x}, \mathbf{x})_{\mathcal{L}} = -1/c,\ c > 0\}. \tag{2}$$

All vectors in this set satisfy the following constraint:

$$\mathbf{x}_{\text{time}} = \sqrt{1/c + \|\mathbf{x}_{\text{space}}\|^2}. \tag{3}$$

Lifting Embeddings onto the Hyperboloid [11]. We map the physical representation $V = \{v_i\}_{i=1}^N$ and node representation $E = \{e_i\}_{i=1}^N$ into the Lorentz model as $v_i^{\mathcal{L}}$ and $e_i^{\mathcal{L}}$ via the exponential map. Let the embedding vector (v_i, g_i) be $\mathbf{v}_{\text{enc}} \in \mathbb{R}^n$. We need to apply a transformation such that the resulting vector lies on the Lorentz hyperboloid $\mathcal{L}^n$ in $\mathbb{R}^{n+1}$. Let the vector $\mathbf{v} = [\mathbf{v}_{\text{enc}}, 0] \in \mathbb{R}^{n+1}$. We parameterize *only the space components* of the Lorentz model ($\mathbf{v}_{\text{enc}} = \mathbf{v}_{\text{space}}$) [11]. Due to such parameterization, we can simplify the exponential map as:

$$\mathbf{x}_{\text{space}} = \frac{\sinh(\sqrt{c\|\mathbf{v}_{\text{space}}\|})}{\sqrt{c\|\mathbf{v}_{\text{space}}\|}} \mathbf{v}_{\text{space}}. \tag{4}$$

The corresponding *time component* x_{time} can be computed from $\mathbf{x}_{\text{space}}$ using Eq. 3. The resulting $\mathbf{x}$ always lies on the hyperboloid. To prevent numerical overflow, we scale all vectors $\mathbf{v}_{\text{space}}$ in a batch before applying the mapping using two learnable scalars ω_{img} and ω_{txt}. These are initialized to $\sqrt{1/n}$ so that the Euclidean embeddings have an expected unit norm at initialization.

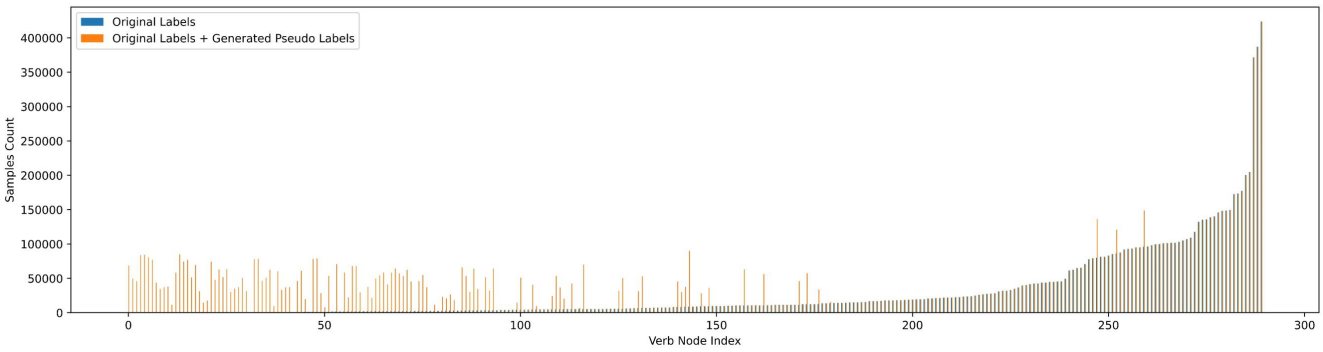


Figure 4. Sample distribution before/after generating pseudo labels.

Lorentzian Distance [11]. The similarity $\mathcal{S}(v_i, e_i)$ is measured via the negative of Lorentzian distance $d_{\mathcal{L}}(\cdot,\cdot)$ between $v_i^{\mathcal{L}}$ and $e_i^{\mathcal{L}}$. A *geodesic* is the shortest path between two points on the manifold. Geodesics in the Lorentz model are curves traced by the intersection of the hyperboloid with hyperplanes passing through the origin of $\mathbb{R}^{n+1}$. The *Lorentzian distance* between two points $\mathbf{x}, \mathbf{y} \in \mathcal{L}^n$ is:

$$d_{\mathcal{L}}(\mathbf{x}, \mathbf{y}) = \sqrt{\frac{1}{c} \cdot \cosh^{-1}(-c(\mathbf{x}, \mathbf{y})_{\mathcal{L}})}. \tag{5}$$

Entailment Cone [11]. If $y_i = 1$, the physical representation $v_i^{\mathcal{L}}$ should lie inside the entailment cone [14] of the node representation $e_i^{\mathcal{L}}$.

For each $\mathbf{x}$, which narrows as we go farther from the origin, the entailment cone is defined by the half-aperture:

$$\alpha(\mathbf{x}) = \sin^{-1}\left(\frac{2K}{\sqrt{c}\|\mathbf{x}_{\text{space}}\|}\right), \tag{6}$$

where a constant $K = 0.1$ is used for setting boundary conditions near the origin. We aim to identify and penalize occasions where the paired image embedding $\mathbf{y}$ lies outside the entailment cone. For this, we measure the exterior angle $\theta(\mathbf{x}, \mathbf{y}) = \pi - \angle O\mathbf{x}\mathbf{y}$:

$$\theta(\mathbf{x}, \mathbf{y}) = \cos^{-1}\left(\frac{y_{\text{time}} + x_{\text{time}} c(\mathbf{x}, \mathbf{y})_{\mathcal{L}}}{\|\mathbf{x}_{\text{space}}\|\sqrt{(c(\mathbf{x}, \mathbf{y})_{\mathcal{L}})^2 - 1}}\right). \tag{7}$$

If the exterior angle is smaller than the aperture, then the partial order relation between $\mathbf{x}$ and $\mathbf{y}$ is already satisfied and we need not penalize anything, while if the angle is greater, we need to reduce it. This is captured by the following loss function (written below for a single $\mathbf{x}, \mathbf{y}$ pair):

$$\mathcal{L}_{\text{entail}}(\mathbf{x}, \mathbf{y}) = \max(0, \theta(\mathbf{x}, \mathbf{y}) - \alpha(\mathbf{x})). \tag{8}$$

4. Details of S2P

Though we focus on P2S mapping, with the learned abundant semantic representation of nodes and the collected 3D data, we wonder if we can do the inverse mapping, *i.e.*, Semantic-to-Physical space (S2P). S2P should be scalable to different semantic granularities and flexible with either **single**- or **multi**-node and generate reasonable 3D motions. We propose a simple model to verify our assumption. We train conditional Variational Auto-Encoders (cVAE) conditioned on the node semantic and geometric encoding E to map S to P. The encoder takes the E and V as input, outputting the mean μ and log-variance σ for a Gaussian distribution, from which we sample a latent encoding z. z is concatenated with E and then fed to the decoder, getting the reconstructed V'. We adopted SMPL [40] parameters as V. For a sample belonging to multiple nodes, we take the mean of their corresponding E as the condition. We train S2P on the 3D data of *Pangea*, using KL divergence driving the predicted distribution to normal distribution and an L2 reconstruction loss of the SMPL parameters.

The encoder and decoder in the cVAE are implemented as a 2-layer MLP. The semantic and geometric encoders in P2S are *frozen* during S2P. The model is trained on *Pangea* using an Adam optimizer for 100 epochs, with a batch size of 256. The learning rate is warmed up from 5e-8 to 2e-6 for the initial 2 epochs and then decayed with a cosine scheduler.

5. Datasets Details in Experiments

HICO [5] is an *image-level* benchmark for Human-Object Interaction (HOI) recognition. It has 38,116 and 9,658 images in the train and test sets and defines 600 HOIs composed of 117 verbs and 80 COCO objects [34]. Each image has an image-level label which is the aggregation over all HOIs in an image without human boxes. We use mAP for multi-label classification.

HAA [9] is a video *clip-level* human-centric atomic action dataset. It defined 500 actions and contains 10,000 video clips which are split into 8,000 training, 500 validating, and 1,500 testing clips. Each video clip has one single action label. The top-1 accuracy metric is utilized for multi-

class classification.

HMDB51 [30] is a video *clip-level* dataset consisting of 6,766 internet videos over 51 classes, and each video has from 20 to 1,000 frames. Each video clip has one single action label. We report the average top-1 accuracy on the standard three splits.

Kinetics-400 [26] is a video *clip-level* human-focused dataset that includes 240 K training clips and 20 K validation clips over 400 action classes. Each video lasts for about 10 seconds and contains one single label. We report the top-1 accuracy and top-5 accuracy on the official validation set as the convention.

BABEL [56] is a large-scale 3D action dataset covering a wide range of human motions, including over 250 unique action classes. It is built upon AMASS [44] by annotating the sequences with *sequence-level* and *frame-level* action classes, represented with the SMPL/SMPL-X body model [40, 50]. Over 43.5 hours of MoCap data is provided with 28,033 sequence labels and 63,353 frame labels and is categorized into one of 260 action classes. We follow the evaluation protocol of BABEL-120 under the dense label-only setting, containing a span of MoCap sequences belonging to 120 classes, where 13,320 sequences are divided into train (60%), val (20%), and test (20%) sets. A motion-capture span of 5 seconds or less is given, and our model is required to predict the actions in it. Top-1 accuracy is reported. To show our ability in the long-tail classes, the Top-1-norm (the mean Top-1 across classes) is also reported. We adopt PointNet++ [57] trained on *Pangea* as initialization and finetune it on BABEL. Note that the BABEL-120 benchmark is based on motion sequences. To adapt our model to the setting, we down-sample the original sequence from 60 FPS to 3 FPS, perform inference on all the down-sampled frames, and use *mean pooling* to acquire the final score.

HAA4D [72] is an extension of HAA [9]. 3,300 videos of 300 human atomic action classes from HAA are selected to construct a class-balanced and diverse dataset. Each video is annotated with globally aligned 4D human skeletons. We follow the conventional action classification setting and data split [72]. For classes containing 20 samples, the first 10 samples are adopted for training, and the rest are used for inference. For classes containing 2 samples, the one with a bigger index is adopted for training, while the other one is adopted for inference. We adopt PointNet++ [57] pretrained on *Pangea* as initialization and finetune it on HAA4D. Since HAA4D only provides 4D skeletons, we fit the provided skeletons with SMPL [40] and use the SMPL parameters for training and inference. We perform inference on all the down-sampled frames and use *mean pooling* to acquire the final score.

6. Details of Image/Video Transfer Learning

6.1. Transfer Learning Stages

P2S pretrained on *Pangea* with node classification is a knowledgeable **backbone** and can be used in transfer learning. There are three stages in transfer learning: a) Training P2S on *Pangea*, but with the val & test sets of the downstream target dataset **excluded** following a strict transfer learning setting. b) Finetuning P2S on the target dataset train set. c) Training a small MLP to transform $\mathcal{S}_{node}$ to $\mathcal{S}_{act}$, with the node prediction fixed.

6.2. Training P2S

For the convenience of expression, we divide our *Pangea* database in Suppl. Tab. 1 into 4 splits: 1) Willow Action [10] $\sim$ HAKE [32]: image datasets; 2) HMDB51 [30] $\sim$ Charades [63]: video datasets with relatively small scale; 3) Charades-Ego [64] $\sim$ Kinetics [4]: video datasets with relatively large scale; 4) HumanAct12 [20] $\sim$ HAA4D [72]: skeleton/MoCap datasets.

We select images from split 1&2 to construct *Pangea* test set to represent verb node semantics. The remaining images are used for training. We first train a CLIP model with image-text pairs to get good physical representations, and then freeze the physical representations and train P2S.

To train physical representations, we use a CLIP pretrained ViT-B/32 image encoder to extract visual features with a resolution of 224. An AdamW [42] optimizer with a weight decay of 0.05 is used in training. We first use split 1&2 data to train the model for 15 epochs with a batch size of 256 (split 3 is currently excluded to avoid the image domain gap). The learning rate is warmed up from 5e-7 to 1e-5 for the initial 2 epochs, then decayed with a cosine scheduler. Then we use split 1&2&3 data to finetune the model for 50 epochs with a batch size of 256. The learning rate is warmed up from 5e-8 to 2e-6 for the initial 2 epochs, then decayed with a cosine scheduler. When training with split 1&2&3 data, a fixed number of samples from split 3 data are randomly sampled in each epoch for efficiency.

To train P2S, we freeze the physical representations and train the text encoders and the hyperbolic representations. The model is trained for 5 epochs with a batch size of 64. The learning rate is warmed up from 5e-8 to 2e-6 for the initial 2 epochs, then decayed with a cosine scheduler.

Additionally, HMDB51 data is excluded from the training data to prevent data pollution because it has three train/test splits that intersect with each other. Also, for transfer learning on Kinetics-400, we use a new P2S model, where we exclude the data in Kinetics-700 but not in Kinetics-400 to prevent data pollution.

6.3. Video Temporal Encoding

For video benchmarks, we adopt lite implementations for temporal encoding and do not use video augmentation methods. For the simplest temporal coding, we cut out fixed 8 frames for each video clip and *average* logits of 8 frames as the clip logit. We compare several simple temporal coding methods on HAA [9] transfer learning.

- Average prediction of frames. In training, supervision is applied to each frame. In testing, predictions of 8 frames are *averaged* as the clip prediction. Our P2S achieves 71.40% acc with this temporal encoding.
- Mean pooling. Frame-level visual features are first extracted, and the clip-level visual feature is obtained via simple mean pooling of frame-level ones. In training, supervision is applied to each clip. In inference, the clip prediction is directly outputted. With feature mean pooling, our P2S achieves 71.02% acc.
- Temporal transformer. It is operated similarly to mean pooling, other than a temporal transformer inserted before the mean pooling of frame-level features. With the temporal transformer, our P2S achieves 71.47% acc.

From the above results, we can find that with a more sophisticated model, the performance is higher too. In future work, we believe a larger model with more computation power support will achieve more significant performance improvements with our *Pangea*. In this work, we report P2S results with *average prediction of frames* temporal encoding for simplicity. Even with a very simple temporal encoding, P2S performs comparably with some spatio-temporal (ST) methods. P2S can also be used as a *plug-and-play* method, we report the results of fusing P2S with SOTA video models.

6.4. HICO

With the pretrained P2S (Suppl. Sec 6.2), we first finetune P2S on the HICO train set for 10 epochs, with a batch size of 64. The learning rate is warmed up from 5e-7 to 1e-5 for the initial 2 epochs, then decayed with a cosine scheduler. We then train the transformation from $\mathcal{S}_{node}$ to $\mathcal{S}_{act}$ with the node prediction fixed. The model is trained for 50 epochs, with a batch size of 64 and a learning rate of 1e-4.

We find that HICO [5] designed for human-object interaction (HOI) recognition (verb-object, *e.g.*, `sit_on-chair`) is more difficult than common action recognition (verb, *e.g.*, `sitting`). Moreover, most of *Pangea* data are videos and thus have a larger domain gap with HICO. Thus, compared with other video-based benchmarks, HICO [5] benefits less from P2S pretraining.

6.5. HAA

With the pretrained P2S (Suppl. Sec 6.2), we conduct the transfer learning. We finetune P2S on the HAA train set for 10 epochs, with a batch size of 64. The learning rate is warmed up from 5e-7 to 1e-5 for the initial 2 epochs, then decayed with a cosine scheduler. Then we train the transformation from $\mathcal{S}_{node}$ to $\mathcal{S}_{act}$ with the node prediction fixed. The model is trained for 40 epochs, with a batch size of 64 and a learning rate of 2e-4.

For the experiments of integrating P2S with MLLM, we tried a SOTA MLLM: LLaMA Adapter V2 [15].

When trained *without* P2S, the backbone is finetuned on train set to output captions indicating the activity. The prompt is formulated as

"Generate caption of this image".

The model is required to answer:

"The image shows a person's activity: XXX."

(*e.g.* "The image shows a person's activity: shuffle_dance.") Then the top-1 accuracy is calculated by comparing the semantic distance between the output caption and ground-truth actions based on a CLIP [58] ViT-B/32 pretrained text encoder.

When trained *with* P2S, we formulate P2S prediction as a prompt and require the LLM to output the activity shown in the image. In detail, the prompt is formulated as

"Some information related to the person's activity is: XXX. Describe the person's activity."

The model is required to answer:

"The image shows a person's activity: XXX."

To fuse the model w/ and w/o P2S, during inference, we ensemble the semantic distances between captions w/wo P2S and ground-truth action caption described above. For HICO[5], as the model is required to give a list of HOIs containing one verb and one object each, we extract the HOIs from the generated caption, compare the semantic distances between predicted HOIs and GT HOIs, and calculate mAP following standard evaluation protocol.

We prepare the data following the setting of stage 2/stage 1 for models w/wo P2S. The model is trained following the setting of stage 1 on both occasions.

6.6. HMDB51

With the pretrained P2S (Suppl. Sec 6.2), we first finetune P2S on HMDB51 train set for 10 epochs, with a batch size of 64. The learning rate is warmed up from 5e-7 to 1e-5 for the initial 2 epochs, then decayed with a cosine scheduler. Then we train the transformation from $\mathcal{S}_{node}$ to $\mathcal{S}_{act}$ with the node prediction fixed. The model is trained for 10 epochs, with a batch size of 512. The learning rate is warmed up from 5e-7 to 1e-5 for the initial 2 epochs, then decayed with a cosine scheduler.

6.7. Kinetics-400

With the pretrained P2S (Suppl. Sec 6.2), we conduct: a) Finetuning P2S on Kinetics-400 train set for 15 epochs, with a batch size of 192. The learning rate is warmed up from 1e-7 to 2e-6 for the initial 5 epochs, then decayed with a cosine scheduler. b) Training the transformation from $\mathcal{S}_{node}$ to $\mathcal{S}_{act}$ with the node prediction fixed. The model is trained for 20 epochs, with a batch size of 512. The learning rate is warmed up from 1e-7 to 2e-6 for the initial 5 epochs, then decayed with a cosine scheduler.

We find a decreased performance when pre-trained with CLIP-Pangea on Kinetics-400. This is possibly caused by the large data scale and complex action classes (400 total) of Kinetics-400 compared with other downstream datasets.

7. Details of 3D Transfer Learning

For 3D human point clouds, we use PointNet++ [57] as the encoder. An AdamW [42] optimizer with a weight decay of 0.05 is used. The model is trained for 100 epochs with a batch size of 128. The learning rate is warmed up from 5e-8 to 2e-6 for the initial 2 epochs, then decayed with a cosine scheduler. For P2S learning, we use 601 K 3D training human instances and test the model on *Pangea* test set with 172 K 3D human instances. About 75% of the human instances are obtained from single-view reconstruction [23, 68]. We adopt GT 3D human for BABEL [56] and use reconstructed 3D human for other datasets.

7.1. BABEL

To show the strength of P2S, we further conduct transfer learning on a large-scale 3D action dataset BABEL [56]. We compare our method with the BABEL official baseline [56]. We adopt 2s-AGCN as the baseline following BABEL [56], which utilizes temporal information. Besides, we use PointNet++ and CLIP as extra baselines. Surprisingly, we find that the simple pipeline PointNet++ considerably outperforms its counterpart 2s-AGCN. On one hand, we find that the baseline CLIP performs not well. The reason may be that, without enough 3D pretraining data, the image-based CLIP cannot adapt to the domain of BABEL well. It can be verified that CLIP-*Pangea* performs much better and even outperforms PointNet++-*Pangea* with the help of 3D pretraining samples from *Pangea*. On the other hand, PointNet++ performs much more robustly than CLIP as it is designed to encode the 3D point cloud information which suits this task better. However, they all perform worse than our P2S. As shown, P2S without heavy temporal encoding outperforms all baselines. PointNet++-*Pangea* and CLIP-*Pangea* also show superiorities upon their original setting PointNet++ and CLIP thanks to the extensive knowledge from *Pangea*.

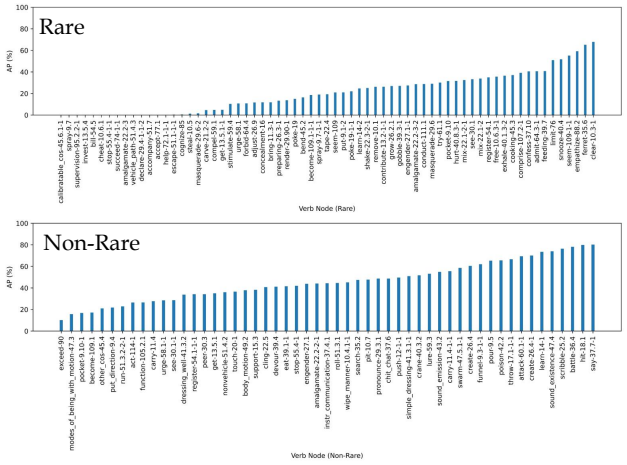


Figure 5. P2S performance on selected rare/non-rare verb nodes on Pangea benchmark. There are a total of 133 rare nodes and 157 non-rare nodes.

7.2. HAA4D

Transfer learning is also conducted on the recently proposed 3D action dataset HAA4D [72]. We compare our method with the HAA4D official baseline [72]. From the comparison of results, we draw a similar conclusion to the one on BABEL. As shown, competitive performance is achieved with the help of *Pangea* pretraining for PointNet++ and CLIP. Meanwhile, the proposed methods such as disentangling, semantic, and geometric encoding help P2S further outperform all baselines and SOTA. We also notice that the improvement on HAA4D of P2S upon the SOTA method SGN is relatively smaller. We recognize the reason as two-fold. First, HAA4D provides 3D keypoints as GT annotation, thus we have to fit the SMPL model to the keypoints for the SMPL parameters. This results in noisy inputs for P2S. Second, HAA4D tends to focus more on human atomic body motions. The frames are therefore less discriminative, weakening the performance of our frame-level P2S on HAA4D.

8. Additional Results of P2S and S2P

8.1. Action Recognition with P2S

We list performance on selected rare/non-rare verb nodes on *Pangea* benchmark in Suppl. Fig. 5. Our P2S achieves decent performance on both rare and non-rare verb nodes.

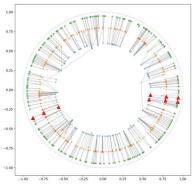
Suppl. Fig. 6 further illustrates two examples of images and predicted verb node logits from the *Pangea* test set. For each leaf node with high prediction, its verb members and parent node are shown.

We make further analysis and discussion of Pangea pretraining benefits as follows. **a) 3D vs. 2D Benchmark:** P2S presents more evident performance improvement in 3D

Action Classes

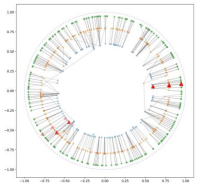
HAA500：basketball_shoot

Predicted Verb Nodes



Verb Node	Members	Parent Node	Logit
carry-11.4-1-1	heave, push, thrust…	11-Verbs of Sending and Carrying	1.00
push-12	heave…	12-Verbs of Exerting Force: Push/Pull Verbs	1.00
throw-17.1-1-1	throw, toss, launch…	17-Verbs of Throwing	1.00
pelt-17.2	pelt, stone…	17-Verbs of Throwing	0.63

HAA500：chopping_meat



Verb Node	Members	Parent Node	Logit
carve-21.2-1	chip, crunch, slice…	21-Verbs of Cutting	0.94
cut-21.1-1	chop, cut, slice…	21-Verbs of Cutting	0.93
disassemble-23.3	detach, disassemble, slice…	23-Verbs of Separating and Disassembling	0.77
touch-20-1	grasp, grip, touch…	20-Verbs of Contact: Touch Verbs	0.26

Figure 6. Example images and predicted verb node logits from the *Pangea* test set.

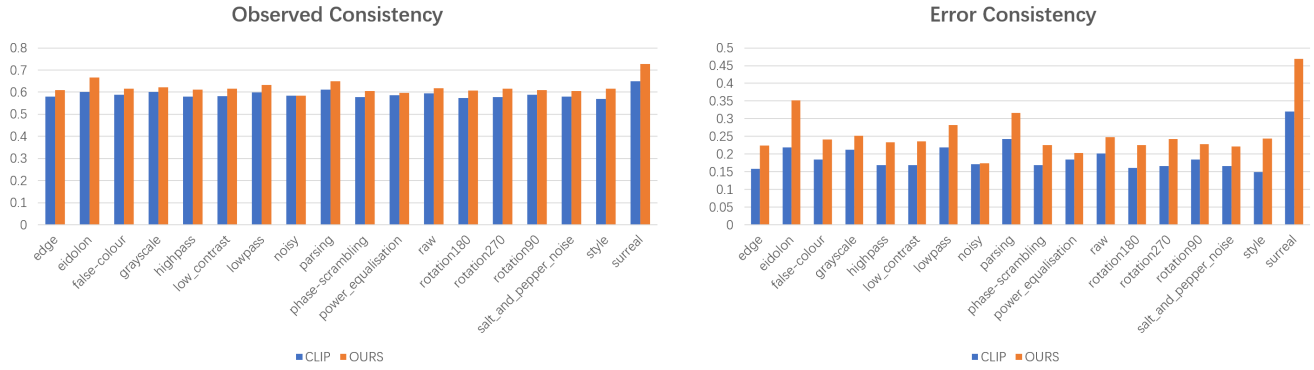


Figure 7. Consistency analysis upon CLIP and our method.

benchmarks mainly because: *1) Nature of tasks.* Typically, the 3D benchmarks have smaller-scale train/test sets and simpler baselines than 2D image/video benchmarks. *2) Smaller domain gap.* Various datasets from the Pangea database share SMPL parameters as 3D representations, whose domain gap is smaller than 2D image/frame pixels. **b) Image vs. Video Benchmark:** P2S performs better in image benchmark (HICO) as the baseline is a concise, image-based one, rather than a sophisticated video-based model. **c) Variations within Video Benchmarks:** P2S shows different benefits across video benchmarks since: *1) Size of pre-training data.* For example, Kinetics-400 is a large-scale dataset. Pangea per-training data which is not from Kinetics-400 accounts for a relatively smaller proportion. *2) Node samples distribution.* Benchmarks are defined on their labels, thus mapped to different nodes. If its nodes have fewer samples in Pangea, the benchmark tends to *benefit less* from P2S and perform worse. To roughly estimate the benefits from node samples distribution, for a benchmark b, we adopt an indicator $I_b = \sum_{i=1}^{N} cnt_i^P \cdot min(1, cnt_i^b)$. Here, N: number of nodes, cnt_i^P/cnt_i^b: sample counts of node i in Pangea/b, $min(1, cnt_i^b)$: whether b has samples for node i. For HAA/HMDB/Kinetics-400, I_b is 93M/66M/92M. The gap between HAA (93M) and HMDB (66M) verifies this.

8.2. P2S Consistency Analysis

To measure the robustness of models, we carry out a consistency test. We follow the setting of [17] and choose 100 head nodes from *Pangea*. For each node, we chose 20 positive image samples and 20 negative samples. The negative samples are chosen from images of other nodes randomly. Each image has undergone **17 transformations**. The first 13 transformations are color-related transformations: grayscale, low contrast, noisy, salt and pepper noise, eidolon, false colourm, highpass, lowpass, phase scrambling, power equalization, rotating 90 degrees, rotating 180 degrees, and rotating 270 degrees. The last 4 transforma-

tions are style changing, edge extracting, human parsing, and surreal. For the so-called surreal transformation, we grab a constructed 3D human mesh from one image and paste it into another background.

Given the results of our method and the baseline CLIP, we make an evaluation based on the metrics proposed in [17] and calculate the **observed consistency** and **error consistency**. Observed consistency and error consistency are calculated concerning every node. For every node, the observed consistency is near or over 60%, the error consistency is between 20% and 30%. There are three transformations with *striking high* consistency, namely human parsing, eidolon, and surreal. We believe that it is because these three transformations are too difficult. Thus, we take the results of all nodes with the 3 weird high-consistency transformations deleted. The final results are shown in Suppl. Fig. 7.

We can find that on both observed and error consistencies, our method P2S performs better than CLIP. Thus, our method not only achieves better accuracy on recognition but also performs more robustly.

8.3. 3D Motion Generation with S2P

We further visualize more results of S2P in Suppl. Fig. 8.

In detail, we align the samples by the pelvis joint, eliminate the root rotation along the z-axis to make the face orientation consistent, and draw skeletons for 100 samples of the *same node* in the same figure to show the sample distribution. As illustrated, S2P is capable of generating reasonable poses for various nodes. And different nodes hold different geometric characteristics. For example, `ride` poses have elbows away from the spine; `sit` poses tend to have elbows near the spine; while there appears to exist more limb contraction for `kneel` and `sleep`. Also, sample generation of node combination is also accessible. By adding the condition `cellphone` upon `sit`, the wrist of the generated samples is restricted to distribute around the pelvis more. Another interesting example is that adding `walk` upon `hug` amplifies the motion range. We show rare combinations like `kneel` plus `hug`. We also show some failure cases of our S2P in Suppl. Fig 9. As shown, when the node combination becomes more complicated, *e.g.*, combining nodes with a larger semantic gap, our S2P could fail to generate accurate 3D actions. Here, we only use a simple cVAE to implement S2P. We believe more advanced models such as Transformer [43] or Diffusion [78] could generate more diverse and realistic 3D actions based on *Pangea*. We leave this to future work.

Representation	Method	Full	Non-Rare	Rare
SMPL	MLP	8.32	12.47	3.42
VPoser	MLP	7.81	11.31	2.55
KeyPoints	MLP	5.45	8.44	1.92
Point Clouds	PointNet++	9.16	12.76	3.76
Point Clouds	CLIP	**11.57**	**16.12**	**6.21**

Table 2. Comparison of different 3D representations on Pangea benchmark.

9. Additional Ablation Studies

9.1. 3D Representation in P2S

To find the best feature extractor for 3D action data, we have tried different ways. Specifically, we compared the performance of different representations of the 3D data: (i) SMPL[40] parameters, (ii) VPoser[50], (iii) body keypoints, and (iv) body point cloud. Note that our dataset only contains SMPL parameters and the other 3 representations are all generated from the SMPL parameters.

For the first 3 representations, we utilize two separate MLPs to encode and classify the 3D data. For the point cloud, we use the PointNet++[57] as the 3D encoder, with an MLP as the classifier, which is referred to as Pointnet++ in the main text and Suppl. Tab. 2. Moreover, we also evaluate the CLIP-like classifier, where the cosine similarity between the encoded point cloud feature and the node semantic feature encoded by a text encoder is adopted as the final classification score. This is referred to as CLIP in the main text and Suppl. Tab. 2. Suppl. Tab. 2 shows the results of different 3D representations on the Pangea Benchmark. Specifically, in some instances of the Pangea dataset, ROMP [68] fails to reconstruct 3D human bodies from the images. For these images, we eliminate these 3D data from the dataset during training and evaluation.

Among these four representations, the point cloud achieves the best results. As for the method, we find that the performance of the model is further improved with a CLIP-like classifier.

We also evaluate the contribution of certain P2S components under the 3D only setting. For example, without disentanglement, the performance degrades to **10.34** mAP, with a considerable performance decline of **2.51** mAP on the Rare set, proving the efficacy of our disentanglement strategy again.

9.2. 2D-3D Fusion in P2S

We compare different 2D-3D fusion strategies in the P2S model. Note that since *Pangea* contains data from different sources, some of which do not provide GT 3D human annotation, we adopt ROMP [68] to generate pseudo 3D human annotations. Suppl. Tab. 3 shows the performance comparison of fusing the multi-modal data at different model stages. Early and middle fusion means that we fuse the ex-

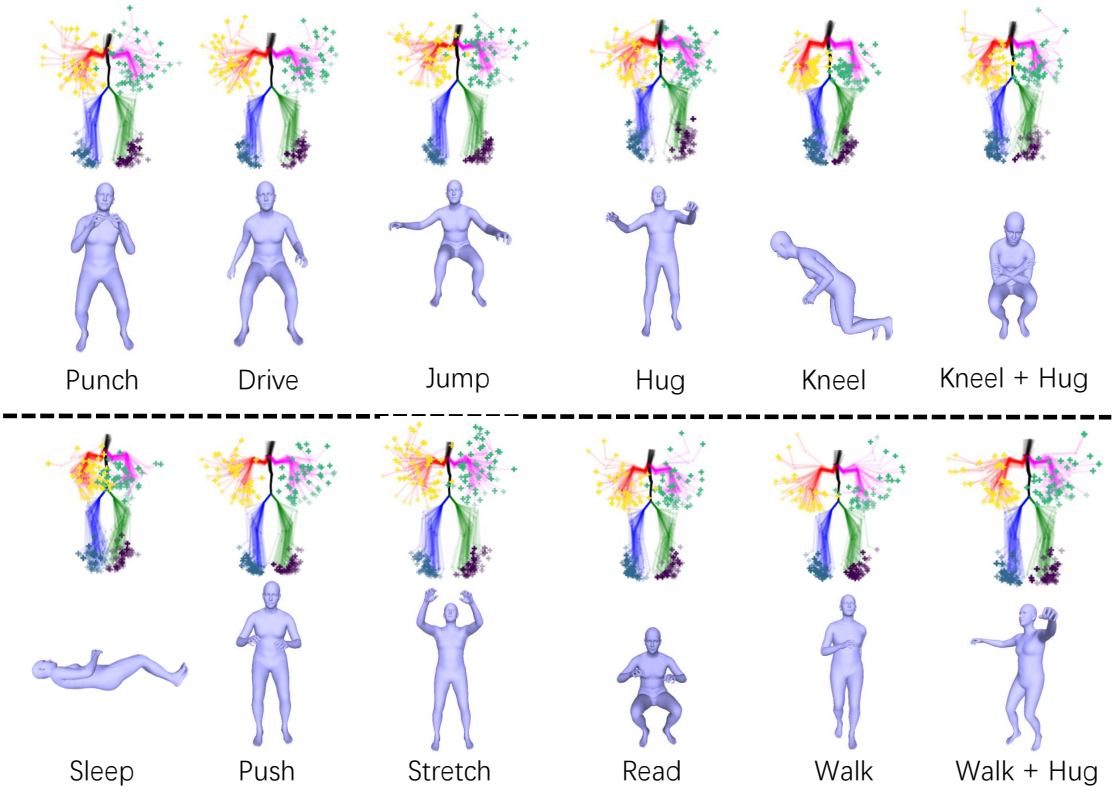


Figure 8. More S2P results.

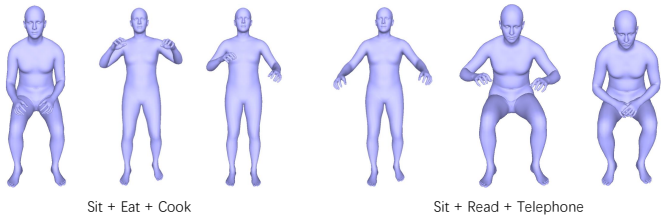


Figure 9. Failure cases of S2P.

Method	Full	Non-Rare	Rare
Early Fusion	37.08	48.05	24.12
Middle Fusion	36.30	47.37	23.23
Late Fusion	**37.55**	**48.84**	**24.22**

Table 3. Comparison of different multi-modal fusion strategies on *Pangea* benchmark.

tracted features of 2D and 3D at the early and middle layers of models respectively. For late fusion, we directly fuse the logits. We can find that the late fusion which directly fuses the outputs of 2D and 3D models performs best.

We also conduct a comparison between 2D only, 3D only, and 2D-3D fusion on *Pangea* in Suppl. Tab. 4. As shown, though P2S with 3D only is not very competitive by itself, they could still compensate for 2D only and bring considerable improvement.

Method	Full	Non-Rare	Rare
CLIP [58]	28.25	37.87	16.90
P2S (2D)	34.46	45.15	21.84
P2S (3D)	11.57	16.12	6.21
P2S (2D+3D)	**37.55**	**48.84**	**24.22**

Table 4. Results of different modality utilization on *Pangea*.

9.3. Verb Node Encoding and Alignment in P2S

Entailment loss. Pangea faces a partial-label learning problem, where a few uncertain labels should have been annotated as True. The entailment loss, which enforces partial order relationships, adds more constraints than the classification loss. Thus, the entailment loss is only applied to positive samples to *avoid over-constraints* on uncertain labels. We apply *an additional ablation study*, where the entailment loss has an additional item for negative samples, following Eq. 32 from [11]. We find a slight performance *degradation* from *34.25* mAP to *33.93* mAP (with $\gamma = 0.1$) on the Pangea test set.

Semantic encoding flexibility. For semantic encoding, the input texts of the nodes are fixed, while the node em-

bedding E is not fixed with *trainable* semantic and geometry encoding. We use TextRank to sample key texts clarifying the node semantics better and taking the summarized text as the text encoder input, to include abundant information for semantic encoding. To further explore semantic encoding flexibility, we conduct *two additional experiments* compared with P2S (34.01 mAP) on the Pangea test set: *1) Augmentation via sampling node descriptions.* In training, a few randomly sampled sentences (up to 77 tokenized symbols) from the node descriptions are input. In inference, the fixed summarized text is input. The performance degrades to 32.43 mAP, possibly since the text bias from unrelated words in node descriptions: *e.g.*, "I put the book on the table", "book" and "table" bring bias. *2) Use pretrained language vectors.* Each sentence is encoded via another pretrained CLIP text encoder, then the encoded features are fed into our text encoder as tokens. Thus, our text encoder receives various sentences as input. The strategy performs comparably (33.96 mAP) with the original one. The possible reason is that there may be a trade-off between the encoded text length and learning difficulty. In the future, given more advanced LLMs to fully utilize the diverse semantic information of verb nodes in our database, we believe things may be different and will further stimulate the potential of our work.

10. More Discussions

In this section, we give some discussions about our system, some possible applications, and future studies based on our *Pangea* and structured semantic space.

(1) Firstly, we discuss more possible **future applications** of our system as follows:

New Emergent and Very Rare Actions. Interestingly, we are creating new actions every day, *e.g.*, new actions such as `play VR games`, `telesurgery` given the new inventions like VR player, telesurgery machine. These new emergent actions may have very limited visual and text data. Given our structured semantic space, we can directly align new actions to their related verb nodes efficiently. Then, we can easily find out the related/similar actions from the previous action database robustly instead of teaching machines a new action from scratch. The need for data collection would be largely reduced. Moreover, it could alleviate the difficulty of incremental learning. Furthermore, sometimes it is very hard to collect data for very rare actions (*e.g.*, `put out fire`), but we can get data easily from its parent, grandparent, or sibling nodes to help us gather its semantics. In inference, different levels of predictions also help because we can enforce their geometry relation consistency to get more robust results.

Customized Finetuning for Downstream Benchmarks. We can also customize the pretrain set for each downstream dataset. For example, for AVA, its classes are related to n nodes in the tree. We can only collect the samples related to these n nodes in our *Pangea* and their closely related neighbors to build a customized and more powerful pretrain or train set for AVA.

Data Usage and Sharing. Given our *Pangea*, it is easy to add new action data in pretraining or finetuning via the one-time verb node-class alignment. This provides a new solution for future applications to connect the data owners of different domains and fields. In the future, it is also promising to marry *Pangea* and Federated learning to study data sharing and security. Thus, we can build an action data platform to share and fully use data and evaluate the contributions of different data providers and annotators.

Training Considering Different Verb Tree Levels. Another possible application is that we can pretrain a model with high-level verb node labels only and then finetune it with finer-grained verb node labels. This follows the learning paradigm from abstract concepts to specific concepts. We leave this to future work.

Joint Learning of P2S and S2P. A promising application of our method is to jointly train P2S and S2P. For example, firstly train P2S and get the representative verb node features and then use it in S2P training. Secondly, we can generate new 3D human samples with S2P via distribution sampling. Next, these new 3D human samples can be input into P2S as pseudo samples. During the process, we can gradually add new data with labels to tune two models. This design may construct a loop to connect the bottom-up and top-down models and may show an interesting property. It lays a foundation for a better understanding of the relationship between human geometry and behavioral semantics.

Hyperbolic Embedding. Besides the geometry information encoding, the hyperbolic latent space also acts as an **interpretable indicator** to represent the action semantics and their change in images and videos, which is more than the performance gains. We think this would be vital for future general and interpretable action recognition studies.

Compositional Complexity. Human actions have compositional complexity at the human part level. On one hand, we can composite two actions such as `eat` and `walk` easily via human body parts control in 3D action generation. On the other hand, this compositionality also brings challenges. Sometimes the label of a sample only reflects the action semantics carried by human parts, *e.g.*, `hold` by hands, `kick` by feet. This phenomenon was studied by HAKE [32, 33] before. Given our structured action semantic space, we may be able to connect human body part states with our verb tree nodes to find out which nodes represent the part-level action semantics and which nodes carry the whole body semantics.

(2) Next, we discuss the **design choices** of our system.

3D Human. In our system, we use multi-modal inputs, *i.e.*, 2D image/video and 3D human point cloud from SMPL mesh. Because we believe though 2D data carries abundant

information about human actions, 3D human carries relatively more geometric information about human bodies. In our tests, we also find that they are complementary to each other. In the future, we believe that 3D action understanding will be a more and more important direction. Moreover, 3D action/motion generation has attracted more and more attention recently too. Currently, we do not use the face and hand detection and reconstruction of 3D humans for simplicity. We can use a more advanced but also heavier whole body detection and 3D reconstruction model such as SMPLify-X [51], to pursue better performance on face-hands related actions such as `eat, talk, grasp`, *etc*. We leave this to future work.

Difference between CLIP-like Models and Ours. Action understanding has a long story but the semantic space is usually defined without guidance, *e.g*., selecting action classes according to the research interests or application requirements. Thus, different datasets cannot be directly used by other domains due to the action class setting divergence and semantic gap. This inhibits the development of general and open-action understanding. Recently, CLIP [58] is proposed to utilize the flexible language prompt to encode the class labels, being able to bypass the class setting to achieve open-vocabulary training. But action semantics have their unique property overlooked by the intuitive visual-language alignment. In detail, verbs usually have many senses under different contexts and scenes. Moreover, verb taxonomy is hierarchical, and different datasets usually adopt verbs in different granularities making the direct visual-language alignment difficult to capture the subtle semantics of actions. Directly using the label texts without any guidance is inefficient and makes it hard to scale for future large-scale applications. Recent works also find that CLIP-style works usually perform not as open as we thought since the confusion of competing text features [60]. In our experiments, we also find that the ambiguity and complexity of action verbs and the obvious multi-label property of active persons hinder the effectiveness of CLIP a lot. In contrast, our structured semantic space design is explicit, well-designed to alleviate ambiguity, and relates the similar verbs thanks to the linguistic knowledge from VerbNet. Thus, our model performs much better than the vanilla CLIP design on large-scale action learning tasks while showing great generalization ability, openness, and extensibility [60]. Besides the unity and broad coverage, an extra benefit of our semantic space is that, though all the data would be placed in our verb tree, different users or researchers can only care about a part of the tree and do not need to process all the data of all the nodes while keeping the semantic structure knowledge.

Weakly-Supervised Learning. In our *Pangea*, due to the costly full annotation of the whole verb tree for all samples, we adopt a weakly supervised way to train the models. In the future, we can annotate more verb nodes for more action classes from existing datasets, supplement more node labels for the existing samples, or utilize the self-supervised learning method designed for the typical positive unlabeled setting (PU, only some of the positive samples have labels) [1] to further advance our weakly-supervised system.

Long-tailed Distribution. Though we collect a lot of data in *Pangea*, the distribution is still long-tailed due to the natural data distribution. However, in the future, the community can easily collect data for the rare nodes and train a more versatile model covering more nodes, and study more on how to generate better pseudo labels according to the language structure knowledge.

References

[1] Anish Acharya, Sujay Sanghavi, Li Jing, Bhargav Bhushanam, Dhruv Choudhary, Michael Rabbat, and Inderjit Dhillon. Positive unlabeled contrastive learning. *arXiv preprint arXiv:2206.01206*, 2022. 14

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 4

[3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *TPAMI*, 2019. 3

[4] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 2, 4, 7

[5] Yu Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng. Hico: A benchmark for recognizing human-object interactions in images. In *ICCV*, 2015. 2, 4, 6, 8

[6] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *WACV*, 2018. 2

[7] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *ICIP*, 2015. 4

[8] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. Monocular expressive body regression through body-driven attention. In *ECCV*, 2020. 1

[9] Jihoon Chung, Cheng hsin Wuu, Hsuan ru Yang, Yu-Wing Tai, and Chi-Keung Tang. Haa500: Human-centric atomic action dataset with curated videos. In *ICCV*, 2021. 4, 6, 7, 8

[10] V. Delaitre, I. Laptev, and J. Sivic. Recognizing human actions in still images: a study of bag-of-features and part-based representations. In *BMVC*, 2010. 4, 7

[11] Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Shanmukha Ramakrishna Vedantam. Hyperbolic image-text representations. In *ICML*, pages 7694–7731. PMLR, 2023. 5, 6, 12

[12] Ankit Dhall, Anastasia Makarova, Octavian Ganea, Dario Pavllo, Michael Greeff, and Andreas Krause. Hierarchical image classification using entailment cone embeddings. In *CVPRW*, 2020. 1

[13] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 4

[14] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *ICML*, 2018. 1, 6

[15] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023. 8

[16] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021. 5

[17] Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Partial success in closing the gap between human and machine vision. *NeurIPS*, 2021. 10, 11

[18] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 2, 4

[19] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018. 1

[20] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *ACMMM*, 2020. 1, 2, 4, 7

[21] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 2013. 1, 4

[22] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 1

[23] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. *arXiv preprint*, 2020. 3, 9

[24] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In *3DV*, 2021. 1

[25] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 1

[26] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 7

[27] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *CVPR*, 2020. 1

[28] O. Kliper-Gross, T. Hassner, and L. Wolf. The action similarity labeling challenge. *TPAMI*, 2012. 4

[29] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019. 1

[30] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011. 4, 7

[31] Jiefeng Li, Can Wang, Wentao Liu, Chen Qian, and Cewu Lu. Hmor: Hierarchical multi-person ordinal relations for monocular multi-person 3d pose estimation. In *ECCV*, 2020. 1

[32] Yong-Lu Li, Liang Xu, Xinpeng Liu, Xijie Huang, Yue Xu, Shiyi Wang, Hao-Shu Fang, Ze Ma, Mingyang Chen, and Cewu Lu. Pastanet: Toward human activity knowledge engine. In *CVPR*, 2020. 4, 7, 13

[33] Yong-Lu Li, Xinpeng Liu, Xiaoqian Wu, Yizhuo Li, Zuoyu Qiu, Liang Xu, Yue Xu, Hao-Shu Fang, and Cewu Lu. Hake: A knowledge engine foundation for human activity understanding. *arXiv preprint arXiv:2202.06851*, 2022. 13

[34] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[35] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. *arXiv preprint arXiv:2208.03550*, 2022. 1

[36] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009. 4

[37] 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. *TPAMI*, 2019. 1

[38] Shaoteng Liu, Jingjing Chen, Liangming Pan, Chong-Wah Ngo, Tat-Seng Chua, and Yu-Gang Jiang. Hyperbolic visual embedding learning for zero-shot recognition. In *CVPR*, 2020. 1

[39] Teng Long, Pascal Mettes, Heng Tao Shen, and Cees GM Snoek. Searching for actions on the hyperbole. In *CVPR*, 2020. 1

[40] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *SIGGRAPH Asia*, 2015. 1, 2, 6, 7, 11

[41] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *TOG*, 2015. 1

[42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7, 9

[43] Thomas Lucas*, Fabien Baradel*, Philippe Weinzaepfel, and Grégory Rogez. Posegpt: Quantization-based 3d human motion generation and forecasting. In *ECCV*, 2022. 11

[44] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, 2019. 2, 7

[45] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2640–2649, 2017. 1

[46] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1

[47] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *NIPS*, 2017. 1

[48] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010. 4

[49] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7025–7034, 2017. 1

[50] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019. 1, 7, 11

[51] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019. 14

[52] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019. 1

[53] Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *ICCV*, 2021. 1, 2

[54] Mathis Petrovich, Michael J. Black, and Gül Varol. TEMOS: Generating diverse human motions from textual descriptions. In *ECCV*, 2022. 1, 2

[55] Matthias Plappert, Christian Mandery, and Tamim Asfour. The KIT motion-language dataset. *Big Data*, 2016. 2

[56] Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. BABEL: Bodies, action and behavior with english labels. In *CVPR*, 2021. 2, 4, 7, 9

[57] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NIPS*, 2017. 7, 9, 11

[58] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 8, 12, 14

[59] Nishant Rai, Haofeng Chen, Jingwei Ji, Rishi Desai, Kazuki Kozuka, Shun Ishizaka, Ehsan Adeli, and Juan Carlos Niebles. Home action genome: Cooperative compositional action understanding. In *CVPR*, 2021. 4

[60] Shuhuai Ren, Lei Li, Xuancheng Ren, Guangxiang Zhao, and Xu Sun. Rethinking the openness of clip. *arXiv preprint arXiv:2206.01986*, 2022. 14

[61] Mohammad Amin Sadeghi and Ali Farhadi. Recognition using visual phrases. In *CVPR*, 2011. 4

[62] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *CVPR*, 2016. 4

[63] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 4, 7

[64] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *CVPR*, 2018. 4, 7

[65] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *CVPR*, 2022. 1

[66] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 4

[67] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2602–2611, 2017. 1

[68] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Black Michael J., and Tao Mei. Monocular, one-stage, regression of multiple 3d people. In *ICCV*, 2021. 1, 2, 3, 9, 11

[69] Dídac Surís, Ruoshi Liu, and Carl Vondrick. Learning the predictability of the future. In *CVPR*, 2021. 1

[70] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. *arXiv preprint arXiv:2203.08063*, 2022. 1, 2

[71] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincar\'e glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*, 2018. 1

[72] Mu-Ruei Tseng, Abhishek Gupta, Chi-Keung Tang, and Yu-Wing Tai. Haa4d: Few-shot human atomic action recognition via 3d spatio-temporal skeletal alignment, 2022. 4, 7, 9

[73] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 1

[74] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021. 1

[75] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint rendering of moving people from monocular video. In *CVPR*, 2022. 1

[76] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *ICCV*, 2011. 4

[77] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang,

Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 1

[78] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 1, 11

[79] Weiyu Zhang, Menglong Zhu, and Konstantinos G Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, 2013. 4

[80] Hang Zhao, Zhicheng Yan, Lorenzo Torresani, and Antonio Torralba. Hacs: Human action clips and segments dataset for recognition and temporal localization. *arXiv preprint arXiv:1712.09374*, 2019. 4