Supplemental Material: Video Pose Distillation for Few-Shot, Fine-Grained Sports Action Recognition

James Hong[1] Matthew Fisher[2] Michaël Gharbi[2] Kayvon Fatahalian[1]
[1]Stanford University [2]Adobe Research

A. Implementation: Video Pose Distillation

This section provides additional implementation details for our method described in Section 3.

Pose: $\mathbf{p}_t$ definition. VPD is not dependent on a specific 2D pose estimator or joint definition. We use an off-the-shelf HRNet [22] to estimate pose in the detected region of the athlete, as is typical for top-down pose estimation. Heuristic tracking, described in Section F, can often provide bounding boxes in frames where person detection fails. We use only 13 of the 17 COCO [11] keypoints (ignoring LEye, REye, LEar, and REar), and we apply the same joint normalization procedure as in [21].

Student inputs. The RGB crops $\mathbf{x}_t$ are derived from the spatial bounding boxes of the athlete in frame t. We expand the bounding box to a square and then pad each side by 10% or 25 pixels, whichever is greater.

Optical flow ϕ_t is computed using RAFT [24] between $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$, where we crop the same location as $\mathbf{x}_t$ in the previous frame for $\mathbf{x}_{t-1}$. In datasets where the frame rate differs between videos, a target frame rate of 25 frames per second (fps) determines $\mathbf{x}_{t-1}$. To obtain the final ϕ_t, we subtract the median of the RAFT output, clip to ± 20 pixels, and quantize into 8-bits.

During training and inference, $\mathbf{x}_t$ is scaled to a range of ± 1 and standardized with respect to the dataset RGB mean and standard deviation; ϕ_t is also centered to ± 0.5. In video frames where the athlete was explicitly detected by Mask R-CNN with a score above 0.8 (see Section F), we use the predicted mask to jitter the background with Gaussian noise ($\sigma = 0.05$) as data augmentation.

For performance reasons, we pre-compute $\mathbf{p}_t$, $\mathbf{x}_t$, and ϕ_t in an offline manner for the entire corpus.

Auxiliary decoder D is a standard fully connected network, whose sole purpose is to provide supervision for training the student F. We use two hidden layers, each with dimension of 128. Note that the ablations without motion in Table 2 do not use D and directly optimize L_2 loss between the student's output and the teacher's $\mathbf{p}_t$.

Student training. The student is initialized with random weights. In each training epoch, we randomly sample 20,000 frames t that meet the pose selection criteria outlined in Section 3.3. We use an AdamW [13] optimizer with learning rate $5e^{-4}$ and a batch size of 100. The student is trained for 1,000 epochs, though in practice the model often converges sooner and using a higher learning rate is also possible. We use the loss on the held-out validation frames to select the best epoch. On a single Titan V GPU, the student model trains in approximately 8 hours.

B. Implementation: Action Recognition

This section provides details about our fine-grained action recognition models and baselines.

B1. BiGRU Architecture

This is a standard bidirectional-GRU [4] architecture. The model is trained on sequences of VI-VPD, 2D-VPD, VIPE*, and normalized 2D joint position features.

The inputs are variable length sequences of per-frame pose features (for each action). The features are sampled to 25 fps in FX35 and Diving48, where frame rate varies from 25 to 60 fps. FSJump6 is a small dataset and normalizing the features also reduces overfitting.

Architecture. We use a two-layer BiGRU as the backbone, with a hidden dimension $h = 128$. The output of the BiGRU is a sequence $H \in \mathbb{R}^{2h \times t}$ of hidden states from the final layer. To obtain a fixed size encoding of this sequence, we max-pool across the time steps in H. To output an action class, the pooled encoding is sent to a fully connected network consisting of BN-Dropout-FC-ReLU-BN-Dropout-FC, with the FC dimensions being $2h$ and the number of output classes.

Training. We train the network with AdamW [13] and a batch size of 50 for 500 epochs (200 on Diving48 due to the larger dataset). Learning rate is initially set to $1e^{-3}$ and adjusted with a cosine schedule. Dropout rate is 0.5 on the dense layers and 0.2 on the input sequence. Data augmentation consists of the horizontally flipped input sequences.

On a single Titan V GPU, our model takes 7 minutes to train for FSJump6, 25 minutes for Tennis7, 50 minutes for

FX35, and 100 minutes for Diving48 over the full datasets.

Inference. At inference time, we feed the input sequence and its horizontal flip to the model; sum the predictions; and output the top predicted class.

B2. Nearest-Neighbor Search

Our nearest-neighbor search (NNS) uses sequence alignment cost with dynamic time warping (DTW).

The inputs are the same as in Section B1, but with each feature vector normalized to unit length.

Inference. We treat the training set as an index. Alignment cost between two sequences of features, normalized by sequence length, is calculated using DTW with pairwise L_2 distance and the symmetricP2 step pattern [18]. Combinations of the regular and horizontally flipped pose sequences in the testing set and training set are considered, with the lowest cost match returned.

Because the computational complexity of inference grows linearly with training set size, this method is unsuited for larger datasets with more examples or classes. DTW is also sensitive to factors such as the precision of the temporal boundaries and the duration of the actions.

B3. Additional Baselines

We evaluated ST-GCN [30], MS-G3D [12], multiscale TRN [34], and GSM [20] on our datasets using the reference implementations released by the authors. For TSN [27], we used the code from the authors of GSM [20]. The GSM [20] codebase extends the TRN [34] and TSN frameworks, and we backported ancillary improvements (e.g., learning rate schedule) to the TRN codebase for fairness.

Skeleton based. The inputs to ST-GCN and MS-G3D are the tracked 2D skeletons of only the identified athlete. For MS-G3D, we trained both the bone and joint feature models and reported their ensemble accuracy. Ensemble accuracy exceeded the separate accuracies in all of our experiments.

End-to-end. We follow the best Diving48 configuration in the GSM [20] paper for the GSM, TSN, and TRNms baselines. This configuration uses 16 frames, compared to 3 to 7 in earlier work [34], and samples 2 clips at inference time. As seen in benchmarks by the authors of [19], additional frames are immensely beneficial for fine-grained action recognition tasks compared to coarse-grained tasks, where the class can often be guessed in a few frames from context [3, 28]. The backbone for these baselines is an InceptionV3 [23], initialized using pretrained weights.

When comparing to TSN and TRN with optical flow, we train using the same cropped flow images as VPD, described in Section A. Flow and RGB model predictions are ensembled to obtain the 2-stream result. Recent architectures that model temporal information in RGB, such as GSM, often perform as well as or better than earlier flow based work.

C. Implementation: Action Retrieval

The search algorithm for action retrieval is identical to nearest neighbor search described in Section B2, for action recognition, except that the pose sequence alignment scores are retained for ranking.

Query set. For FSJump6, Tennis7, and FX35 we evaluate with the entire corpus as queries. For the much larger Diving48 dataset, we use the 1,970 test videos as queries.

D. Implementation: Action Detection

We evaluated pose features for few-shot figure skating jump and tennis swing detection. Our method should be interpreted as a baseline approach to evaluate VPD features, given the lack of prior literature on temporally fine-grained, few-shot video action detection, using pose features. More sophisticated architectures for accomplishing tasks such as generating action proposals and refining boundaries are beyond the scope of this paper.

The inputs are the uncut, per-frame pose feature sequences. For figure skating, the sequences are entire, 160 second long, short programs. ISU [7] scoring rules require that each performance contains two individual jumps and a jump combination (two jumps). For tennis, each point yields two pose sequences, one for each player. The points sampled for training have at least five swings each per player.

For the ResNet-3D [25] baseline, we extracted features for each frame using a Kinetics-400 [8] pretrained model on the 128 × 128 subject crops, with a window of eight frames. A limitation of this baseline is that actions (e.g., tennis swings) can be shorter than the temporal window.

Architecture. We use a two-layer BiGRU as the backbone with a hidden dimension $h = 128$. The hidden states at each time step from the final GRU layer are sent to a fully connected network consisting of BN-Dropout-FC-ReLU-BN-Dropout-FC, with the FC dimensions being $2h$ and 2 (a binary label for whether the frame is part of an action).

Training. The BiGRU is trained on randomly sampled sequences of 250 frames from the training set. We use a batch size of 100, $1e^4$ steps with the AdamW [13] optimizer, and a learning rate of $1e^{-3}$. We apply dropout rates of 0.5 on the dense layers and 0.2 on the input sequence. Because only five examples are provided in this few-shot setting, we use five-fold cross validation to train an ensemble.

The reported results are an average of separate runs on five randomly sampled, fixed few-shot dataset splits.

Inference. We apply the trained BiGRU ensemble to the uncut test videos to obtain averaged frame-level activations. Consecutive activations above 0.2 are selected as proposals; the low threshold is due to the large class imbalance because actions represent only a small fraction of total time. A minimum proposal length of three frames is required. The mean action length in the training data was also used to expand or trim proposals that are too short (less than $0.67\times$) or too long (greater than $1.33\times$).

E. Additional Experiments

This section includes results of additional ablations, analysis, and baselines omitted from the main text.

E1. Ablation: Data Selection Criterion

Mean estimated joint score from the teacher pose estimator is used as the weak-pose selection criterion. Figure S1 shows the distribution of such scores in each of the four sports datasets. Notice that the teacher produces significantly less confident pose estimates on the floor exercise (FX35) and Diving48 datasets, and also on the labeled action portions of all four datasets.

While the optimal selection threshold is ultimately dependent on the calibration and quality of the pose estimator used, we evaluate the effect of tuning the weak-pose selection criterion on three of our datasets: Tennis7, FX35, and Diving48. Table S1 shows results with VI-VPD when various thresholds are applied. There is benefit to ignoring the least confident pose estimates, though setting the threshold too high also diminishes performance, as insufficient data remains to supervise the student.

E2. Ablation: NNS vs. BiGRU for Recognition

Figure 3 notes that the BiGRU classifier for action recognition generally performed better than NNS, except in extremely data-scarce settings, where there are simultaneously few classes and examples per class. Table S2 presents results for both the BiGRU and NNS.

E3. Ablation: Activation Threshold for Detection

In Section D, we use a frame-level activation threshold of 0.2 when proposing action intervals for few-shot action detection. Table S3 shows the impact on average precision (AP) of other thresholds, scored at 0.5 temporal intersection over union (tIoU). The results are similar at nearby thresholds and results at 0.2 are reported for consistency.

E4. Ablation: Action Recognition Architectures

The BiGRU described in Section B1 was used in our experiments for consistency. This section includes a number of additional simple, well-studied architectures that we also tested. Results from these models are given in Table S4 and are often similar; the BiGRU is not necessarily the best performing model in all situations. As Section 4.1 shows, however, the BiGRU is competitive with recent, state-of-the-art methods when trained with VIPE* or our VI-VPD features.

E5. Baseline: GSM Without Cropping on Diving48

In Section 4.1.1, on few-shot action recognition, we reported results from GSM [20] with cropping. This is despite GSM, without cropping, having higher accuracy in the full supervision setting on Diving48 [10] (see Table 1). Table S5 shows that GSM, with cropping, is the stronger baseline when limited supervision is available.

We speculate that cropping forces the GSM model focus on the diver in few-shot settings. In the full supervision setting, the GSM model can learn this information by itself and is limited by noise in the crops and the loss of other information from the frame (e.g., the other diver in synchronized diving; the 3 metre springboard or 10 metre platform; and spatial information).

E6. Analysis: Visualizing Distilled 2D Pose

Although the goal of this paper is to distill pose features for downstream tasks, this section provides preliminary qualitative results on how well distilled features mimic their teachers and reflect the explicit 2D pose. Because the learned VIPE* and VPD features are not designed to be human interpretable, we use normalized 2D joint positions (described in Section A) as the teacher instead, and we train an ablated student without the auxiliary decoder for motion.

Figure S2 compares the teacher's normalized 2D joint features to the student's distilled outputs. Visible errors in the student's predictions show that our distillation method presented in this paper does not solve the explicit 2D pose estimation problem in challenging sports data. However, solving this explicit task is not necessarily required to improve results in downstream tasks that depend on pose.

F. Additional Dataset Details

This section provides additional details about the fine-grained sports video datasets used in the results section.

Figure skating is a new dataset that contains the jumps in 371 singles short programs. Because professional skaters often repeat the same routine in a competitive season, all performances from 2018 are held out for testing.

The six jump types that occur in the FSJump6 dataset are: Axel, flip, loop, Lutz, Salchow, and toe-loop (see Table S6). The labels are verified against the ISU's [7] publicly accessible scoring data. For the classification task, the average label duration is 3.3 seconds and includes the poses from before taking off and after landing.

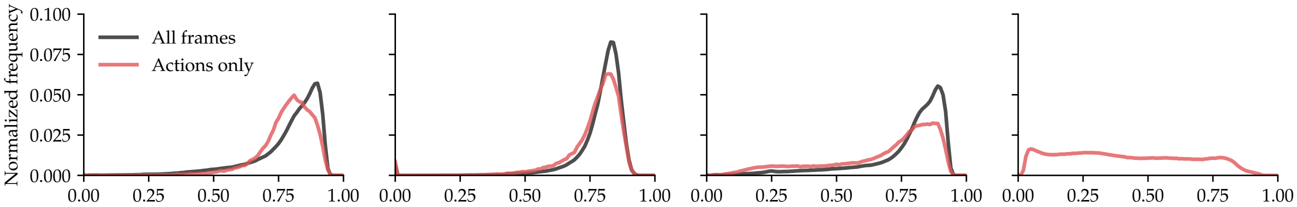
Tennis consists of Vid2Player's [31] swing annotations in nine matches. For action recognition, Tennis7 has seven swing types: forehand topspin, backhand topspin, forehand slice, backhand slice, forehand volley, backhand volley, and overhead. Note that the distribution of actions in tennis is unbalanced, with forehand topspin being the most common.

Dataset	Tennis7				FX35				Diving48		
Score	% All	% Action	Full	16-shot	% All	% Action	Full	16-shot	% All	Full	16-shot
VIPE*	-	-	91.8	67.0	-	-	90.8	75.7	-	78.6	35.0
$\geq$ 0.1	99.4	99.2	93.4	67.5	99.7	99.5	93.9	82.9	89	81.1	43.9
$\geq$ 0.3	99.2	99.0	93.0	67.7	96	91	93.8	84.0	62	85.7	51.8
$\geq$ 0.5	97.9	97.2	**93.4**	69.5	90	79	**94.6**	**84.9**	38	**88.9**	**58.8**
$\geq$ 0.7	89.9	83.9	93.3	**71.1**	78	61	93.9	83.0	17	87.7	50.6
$\geq$ 0.9	1.5	1.3	91.2	65.4	17	8	93.1	79.9	$<$1	73.8	25.7

Table S1: **Top-1 accuracy on action recognition using VI-VPD when varying the weak-pose selection threshold.** For consistency, all results are using the BiGRU (Section B1). Excluding the least confident poses improves accuracy; these poses are most likely to be incorrect. However, setting the threshold too high also decreases accuracy if the supervision becomes too sparse. The percent of poses in all frames (% All) and in action frames (% Action) that are retained at each threshold is also shown. Note: Diving48 [10] only contains action frames.

Training data	4-shot		8-shot		16-shot		32-shot	
Features \ Model	BiGRU	NNS	BiGRU	NNS	BiGRU	NNS	BiGRU	NNS
FSJump6								
Normalized 2D joints	38.5 ± 3.7	50.8 ± 6.1	60.1 ± 4.5	65.3 ± 4.5	72.5 ± 3.9	71.7 ± 3.9	89.7 ± 0.9	79.7 ± 1.8
(Ours) 2D-VPD	43.2 ± 5.2	50.7 ± 5.8	66.1 ± 1.1	70.3 ± 3.7	74.4 ± 3.0	75.7 ± 1.5	90.8 ± 1.9	84.1 ± 1.2
VIPE*	51.1 ± 3.0	64.3 ± 5.0	69.7 ± 2.9	75.7 ± 3.6	80.5 ± 3.5	78.3 ± 2.6	91.3 ± 1.7	84.5 ± 1.3
(Ours) VI-VPD	54.4 ± 5.0	**65.9 ± 5.5**	71.4 ± 1.7	**78.4 ± 2.5**	80.2 ± 1.9	**81.1 ± 2.5**	**92.2 ± 1.2**	86.2 ± 0.7
Tennis7								
Normalized 2D joints	48.0 ± 1.9	54.2 ± 3.4	58.5 ± 3.0	57.0 ± 5.5	64.4 ± 2.6	63.0 ± 2.8	69.7 ± 2.6	64.6 ± 2.3
(Ours) 2D-VPD	53.0 ± 3.3	57.0 ± 3.4	62.0 ± 1.7	61.3 ± 4.8	66.9 ± 1.7	65.0 ± 2.0	71.5 ± 2.4	67.2 ± 1.5
VIPE*	61.4 ± 4.1	62.4 ± 4.4	65.8 ± 3.4	65.6 ± 3.5	67.0 ± 2.8	68.8 ± 4.3	73.2 ± 2.3	70.1 ± 2.0
(Ours) VI-VPD	**63.9 ± 6.1**	62.4 ± 4.5	65.5 ± 4.5	**66.1 ± 3.5**	**71.1 ± 2.4**	68.4 ± 3.5	**76.3 ± 2.0**	70.3 ± 1.8
FX35								
Normalized 2D joints	37.6 ± 1.2	38.0 ± 1.9	54.8 ± 2.6	45.8 ± 1.2	65.6 ± 0.9	52.8 ± 1.4	75.3 ± 0.9	59.0 ± 0.6
(Ours) 2D-VPD	51.2 ± 1.0	47.4 ± 2.1	70.0 ± 1.2	54.9 ± 1.5	82.7 ± 0.6	63.9 ± 1.4	88.8 ± 0.8	69.7 ± 0.5
VIPE*	49.7 ± 0.7	43.0 ± 1.7	62.5 ± 2.1	49.1 ± 0.9	75.7 ± 0.4	54.3 ± 1.2	81.8 ± 0.5	59.7 ± 1.3
(Ours) VI-VPD	**59.3 ± 1.9**	51.0 ± 1.1	**73.0 ± 0.6**	57.1 ± 1.3	**84.9 ± 0.5**	65.4 ± 1.5	**89.1 ± 0.6**	70.6 ± 0.7
Diving48								
Normalized 2D joints	12.6 ± 1.2	13.3 ± 1.4	13.3 ± 1.2	15.3 ± 0.8	25.5 ± 3.5	-	44.2 ± 0.9	-
(Ours) 2D-VPD	27.6 ± 2.6	18.4 ± 2.4	29.4 ± 1.2	22.8 ± 1.4	57.6 ± 6.5	-	76.6 ± 0.9	-
VIPE*	17.0 ± 1.6	12.9 ± 1.6	18.8 ± 1.0	16.1 ± 1.3	35.0 ± 4.5	-	53.2 ± 1.4	-
(Ours) VI-VPD	**29.2 ± 2.5**	16.9 ± 2.1	**34.0 ± 1.2**	21.2 ± 1.0	**58.8 ± 3.6**	-	**76.7 ± 0.8**	-

Table S2: **Results with NNS, under L_2 distance and DTW, compared to the BiGRU in the few-shot setting.** NNS can perform competitively in label-poor settings, though the results are dataset dependent. We did not evaluate NNS on Diving48 past $k = 8$ due to the large number of classes (48), longer average clip length, and the inference time scaling linearly with the number of training examples.



(a) Figure skating (b) Tennis (c) Floor exercise (d) Diving48

Figure S1: **Distribution of mean estimated joint scores in each dataset.** A flatter distribution with more mass to the left indicates greater uncertainty in the estimates. The distribution of joint scores produced by the pose estimator varies by dataset and whether the frames are part of actions or not.

	Figure skating jumps								
Activation threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Pretrained R3D [26]	23.3	23.1	18.2	16.6	14.2	12.8	10.0	7.4	5.9
Normalized 2D Joints	57.1	53.4	50.4	46.9	42.2	37.9	33.3	27.0	20.3
2D-VPD	**63.3**	**61.5**	58.7	56.3	55.2	**53.7**	**51.8**	49.2	44.4
VIPE*	61.5	59.3	58.1	**57.8**	**56.1**	52.0	50.0	44.6	40.0
VI-VPD	61.7	60.7	**59.6**	57.5	54.9	53.0	51.2	**49.9**	**45.7**
	Tennis swings at 200 ms								
Activation threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45
Pretrained R3D [26]	32.1	31.5	30.9	29.9	28.5	27.4	26.4	25.1	22.3
Normalized 2D Joints	46.8	45.3	44.4	43.7	43.6	43.1	41.8	40.1	37.7
2D-VPD	51.2	52.1	53.5	54.0	54.7	54.5	53.6	52.0	49.0
VIPE*	49.6	50.1	50.8	51.2	51.8	52.1	51.5	50.3	48.0
VI-VPD	**55.6**	**56.3**	**57.8**	**58.6**	**59.3**	**59.8**	**59.7**	**58.9**	**56.9**

Table S3: **Few-shot action detection: Average precision (AP) at tIoU** $= 0.5$ **with various frame-level activation thresholds.** These per-frame activations are produced by the BiGRU ensemble described in Section D, and consecutive activations above the threshold are predicted as actions. In Table 4, we use a threshold of 0.2.

Dataset	FSJump6		Tennis7		FX35		Diving48	
Architecture \ Features	VIPE*	VI-VPD	VIPE*	VI-VPD	VIPE*	VI-VPD	VIPE*	VI-VPD
NNS (w/ DTW) [Section B2]	90.6	92.7	89.1	88.6	71.8	81.2	-	-
CNN [9]	93.8	96.0	91.6	93.0	87.8	93.4	58.8	81.3
BiLSTM	97.7	98.1	92.2	**93.4**	90.9	94.3	77.9	88.2
BiLSTM (w/ attn)	97.3	97.9	90.7	92.0	88.8	93.9	76.8	87.5
BiGRU [Section B1]	96.8	97.4	91.8	93.3	90.8	**94.6**	78.6	**88.6**
BiGRU (w/ attn)	96.8	**98.3**	91.1	92.5	89.5	94.3	77.5	88.0

Table S4: **Action recognition architectures: Top-1 accuracy using VIPE*** and VI-VPD features in the full supervision setting.** We experimented with a number of standard architectures for classifying sequences of pose features. The CNN is based on early work on text classification with word vectors [9]. The BiLSTM is similar to the BiGRU described in Section B1. For the BiLSTM and BiGRU with attention, we use an attention mechanism similar to [5]. Results are often similar when comparing across architectures, showing that the improvement from VI-VPD is not reliant on the downstream architecture. For consistency, we use the BiGRU, without attention, for the main results in the paper.

k	Not cropped	Cropped	Difference
8	9.5 ± 1.3	15.1 ± 0.8	+5.5
16	21.8 ± 1.5	33.0 ± 0.9	+11.2
32	49.5 ± 3.1	59.3 ± 2.4	+9.8
64	72.6 ± 1.0	75.6 ± 1.2	+3.0

Table S5: **GSM [20] on Diving48 [10], with and without cropping, in the $k = 8$ to $k = 64$ shot settings.** GSM with subject cropping is the stronger baseline and is used in the few-shot action recognition experiments in Section 4.1.1.

Serves are intentionally excluded from the action recognition task because they always occur at the start of points and do not need to be classified. For swing detection, however, serves are included.

All action recognition models receive a one second interval, centered around the frame of ball contact for the swing.

Floor exercise. We use the videos, labels, and official train/validation split from the floor exercise event of FineGym99 [19]. We focus on floor exercises (FX35) because the data is readily tracked and because the [19] authors report accuracies on this subset. Because actions are often short, for each action, we extracted frames from 250 ms prior to the annotated start time to the end time, and we use these frames as the inputs to our methods and the baselines.

Diving48 [10] contains both individual and synchronized diving. We use the standard train/validation split. For synchronized diving, we track either diver as the subject and tracks can flicker between divers due to missed detections. Tracking is the most challenging in this dataset because of the low resolution, motion blur, and occlusion upon entering the water. Also, because the clips are short, it is more difficult to initialize tracking heuristics that utilize periods of video before and after an action, where the athlete is more static and can be more easily detected and identified.

Subject Tracking

To focus on the athletes, we introduce subject tracking to the figure skating, floor exercises [19], and Diving48 [10] datasets. Our annotations are created with off-the-shelf person detection and tracking algorithms. First, we run a Mask R-CNN detector with a ResNeXt-152-32x8d backbone [29] on every frame to detect instances of people. We use heuristics such as "the largest person in the frame" (e.g., in figure skating, floor exercise, and diving) and "upside down pose" (e.g., in floor exercise and diving) to select the athlete. These selections are tracked across nearby frames with bounding box intersection-over-union, SORT [1], and OpenCV [2] object tracking (CSRT [14]) when detections are missed. This heuristic approach is similar to the one taken by the authors of Vid2Player [31].

Class	Count
Axel	371
Flip	179
Loop	94
Lutz	244
Salchow	61
Toe-loop	497
Total	1,446

Table S6: **Distribution of action classes in FSJump6.**

Class	Count
Backhand slice	812
Backhand topspin	3,134
Backhand volley	140
Forehand slice	215
Forehand topspin	3,732
Forehand volley	123
Overhead	87
Total	8,243

Table S7: **Distribution of action classes in Tennis7.**

Example images of tracked and cropped athletes are shown in Figure S3. We run pose estimation on the pixels contained in and around the tracked boxes.

G. VIPE* Details

We provide details of VIPE*, which is used as the teacher for our view-invariant VI-VPD student. VIPE* is used because the evaluation code and documentation for Pr-VIPE [21] is not released at the time of development. The experiments in this section are to demonstrate that VIPE* is a suitable substitute, based on [21]'s evaluation on coarse-grained action recognition.

Overview. View-invariant pose embedding (VIPE) methods embed 2D joints such that different camera views of the same pose in 3D are similar in the embedding space. VIPE* is trained via 3D lifting to canonicalized features (w.r.t. rotation and body shape). We designed VIPE* to train on multiple (publicly available) datasets with differing 3D joint semantics; we use Human3.6M [6] as well as synthetic pose data from 3DPeople [17], AMASS [15], and NBA2K [35].

Inputs. VIPE* learns view-invariant embeddings by regressing 3D joint features from 2D joint pose. The 2D joint inputs are the 13 COCO [11] keypoints (excluding eyes and ears) normalized as in [21]. To obtain canonicalized 3D features, first, we rotate the 3D pose around the vertical-axis, aligning the torso-normal vector to the depth-axis. Then, we normalize each joint as two unit length offsets from its parent and from the hip (centered to 0). We also concatenate the cosine bone angle at each 3D joint. These transformations standardize 3D poses with respect to body appearance and camera view.

Model. VIPE* uses a similar neural network backbone to [16, 21] and is trained with two losses:

- *3D feature reconstruction loss.* We use a fully-connected decoder that takes embeddings as input. This decoder is discarded after training. To support multi-task training with 3D datasets with different ground-truth joint semantics, we specialize the output layer weights for each dataset.
- *Contrastive embedding loss.* We minimize the pairwise L_2 distance between embeddings of different 2D views of the same 3D pose (positive pairs). We also negatively sample pairs of 2D poses, corresponding to different 3D poses in each action sequence, and maximize their embedding distance. Two 3D poses are considered to be different if one of their joint-bone angles differs by 45° or more.

Substitute for Pr-VIPE. We compare VIPE*'s performance to the coarse-grained action recognition results reported by [21, 33] on the Penn Action [32] dataset. Our results suggest parity with Pr-VIPE when trained with Human3.6M only and a small improvement from extra synthetic data. VIPE* has 98.2% top-1 accuracy (compared to 98.4%, the best result for Pr-VIPE [33]) when trained on the same subjects of the Human3.6M dataset and using nearest-neighbor search as the action recognition method (see Section B2). VIPE* obtains 98.6% accuracy when trained with extra synthetic 3D data. The saturated accuracies of VIPE*, Pr-VIPE [21], and other prior work [33] on the Penn Action dataset suggest that more challenging datasets, such as fine-grained sports, are needed to evaluate new techniques.

For fine-grained action recognition in sports, additional synthetic 3D data improves VIPE* (Table S8). This is especially notable on FX35 and Diving48, which contain a variety of poses that are not well represented by Human3.6M. We use VIPE*, improved with the synthetic 3D data, as the teacher for all of our VI-VPD experiments.

	VIPE* training data		
Dataset	Human3.6M	All	Difference
FSJump6	95.8	96.8	+1.0
Tennis7	91.9	91.8	-0.1
FX35	87.7	90.8	+3.1
Diving48	66.8	76.8	+10.0

Table S8: **Effect of additional synthetic 3D data (all) for VIPE* on fine-grained sports datasets.** Top-1 accuracy on fine-grained sports action recognition is shown. The improvement is largest on FX35 and Diving48, which differ the most from the common poses in Human 3.6M [6]. We use VIPE* trained with all of the 3D data as the teacher for VI-VPD and the VIPE* baselines in Section 4.1. Note that vertically augmenting VIPE* for Diving48, as described in Section 4.1, further increases accuracy to 78.6% (over the 76.8% shown above).

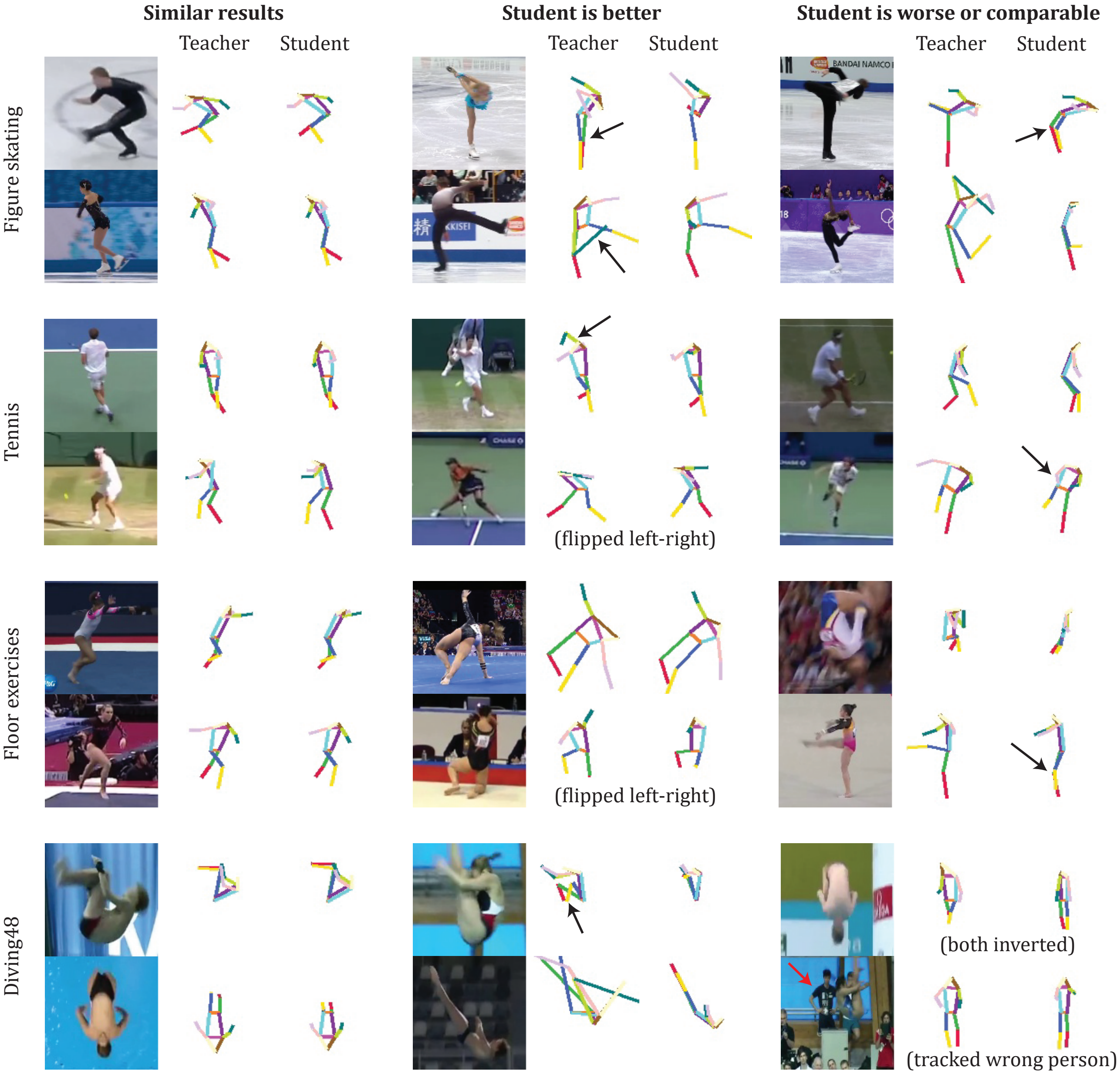
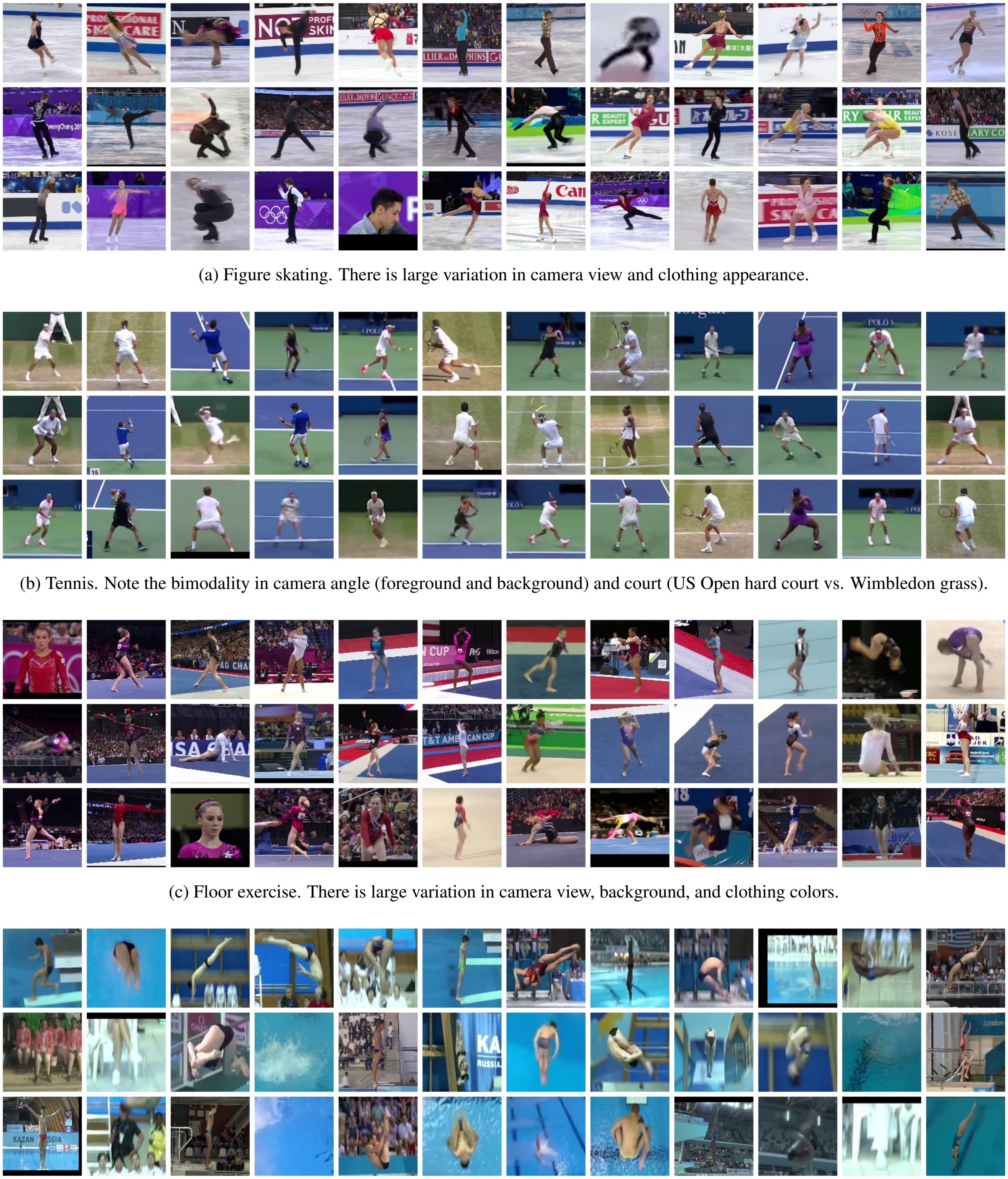


Figure S2: **Qualitative examples of distilled 2D joints.** To investigate whether distillation improves the more general, explicit 2D pose estimation problem, we visualize the output of an ablated student trained to distill 2D poses, without the motion component. The sole learning objective in this experiment is to mimic the normalized 2D joint offsets produced by the teacher. As seen above, the student's output often is similar to the teacher's, especially when the teacher performs well (left column). The student can sometimes produce more plausible results by avoiding extreme errors by the teacher, such as inverting left and right or when pose keypoints are jumbled (middle column). However, as the example failures in the right column show, this student is far from perfect. Ground-truth 2D pose is not available for these datasets, making further quantitative evaluation difficult.



(a) Figure skating. There is large variation in camera view and clothing appearance.

(b) Tennis. Note the bimodality in camera angle (foreground and background) and court (US Open hard court vs. Wimbledon grass).

(c) Floor exercise. There is large variation in camera view, background, and clothing colors.

(d) Diving48. There are frequent errors and lapses in pose tracking, especially after entry into the water.

Figure S3: **Examples of cropped athletes**, based on tracking in the four sports datasets.

References

[1] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple Online and Realtime Tracking. In *ICIP*, 2016. 6

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 6

[3] Jinwoo Choi, Chen Gao, C. E. Joseph Messou, and Jia-Bin Huang. Why Can't I Dance in the Mall? Learning to Mitigate Scene Bias in Action Recognition. In *NeurIPS*, 2019. 2

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*, chapter Sequence Modeling: Recurrent and Recursive Nets. MIT Press, 2016. `http://www.deeplearningbook.org`. 1

[5] IBM. Pytorch-seq2seq, 2019. 5

[6] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014. 6, 7

[7] ISU. International Skating Union. 2, 3

[8] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics Human Action Video Dataset, 2017. arXiv:1705.06950. 2

[9] Yoon Kim. Convolutional Neural Networks for Sentence Classification. In *EMNLP*, 2014. 5

[10] Yingwei Li, Yi Li, and Nuno Vasconcelos. RESOUND: Towards Action Recognition without Representation Bias. In *ECCV*, 2018. 3, 4, 6

[11] 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. 1, 6

[12] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition. In *CVPR*, 2020. 2

[13] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019. 1, 2

[14] Alan Lukežič, Tomáš Vojíř, Luka Čehovin Zajc, Jiř'i Matas, and Matej Kristan. Discriminative Correlation Filter with Channel and Spatial Reliability. In *CVPR*, 2017. 6

[15] 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. 6

[16] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017. 7

[17] Albert Pumarola, Jordi Sanchez, Gary Choi, Alberto Sanfeliu, and Francesc Moreno-Noguer. 3DPeople: Modeling the Geometry of Dressed Humans. In *ICCV*, 2019. 6

[18] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978. 2

[19] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. FineGym: A Hierarchical Video Dataset for Fine-Grained Action Understanding. In *CVPR*, 2020. 2, 6

[20] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-Shift Networks for Video Action Recognition. In *CVPR*, 2020. 2, 3, 6

[21] Jennifer J Sun, Jiaping Zhao, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, and Ting Liu. View-Invariant Probabilistic Embedding for Human Pose. In *ECCV*, 2020. 1, 6, 7

[22] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep High-Resolution Representation Learning for Human Pose Estimation. In *CVPR*, 2019. 1

[23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *CVPR*, 2016. 2

[24] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *ECCV*, 2020. 1

[25] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *CVPR*, 2018. 2

[26] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *CVPR*, 2018. 5

[27] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Val Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *ECCV*, 2016. 2

[28] Philippe Weinzaepfel and Grégory Rogez. Mimetics: Towards understanding human actions out of context, 2021. arXiv:1912.07249. 2

[29] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 6

[30] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *AAAI*, 2018. 2

[31] Haotian Zhang, Cristobal Sciutto, Maneesh Agrawala, and Kayvon Fatahalian. Vid2Player: Controllable Video Sprites That Behave and Appear Like Professional Tennis Players. *ACM Transactions on Graphics*, 40(3), 2021. 3, 6

[32] Weiyu Zhang, Menglong Zhu, and Konstantinos G. Derpanis. From Actemes to Action: A Strongly-Supervised Representation for Detailed Action Understanding. In *ICCV*, 2013. 7

[33] Long Zhao, Yuxiao Wang, Jiaping Zhao, Liangzhe Yuan, Jennifer J. Sun, Florian Schroff, Hartwig Adam, Xi Peng, Dimitris Metaxas, and Ting Liu. Learning View-Disentangled Human Pose Representation by Contrastive Cross-View Mutual Information Maximization. In *CVPR*, 2021. 7

[34] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal Relational Reasoning in Videos. In *ECCV*, 2018. 2

[35] Luyang Zhu, Konstantinos Rematas, Brian Curless, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Reconstructing NBA Players. In *ECCV*, 2020. 6