Addressing Data Scarcity in Depth-Based Human Action Recognition via Zero-Shot Depth Estimation

Supplementary Material

1. Implementation Details

All experiments use the same technical configuration to ensure results are comparable and reproducible.

1.1. Data Splits

We adhere to a a cross-subject evaluation strategy, ensuring that no subjects from the training set are included in the validation or test sets. For UTD-MHAD and UTKinect, a single subject from the original training partition is held out for the validation set. For NTU RGB+D and PKU-MMD datasets, the validation set is constructed by randomly sampling 20% of the available training subjects. An overview of the resulting data splits is provided in Table 1.

Table 1. **Dataset splits for training, validation, and testing.** Video counts across different splits for all datasets after filtering for overlapping actions and consistent viewpoints.

Dataset	Training	Validation	Testing	Total
UTD-MHAD	84	28	111	223
UTKinect	56	14	70	140
NTU RGB+D	7134	2050	3772	12956
PKU-MMD	1401	353	537	2291

1.2. Preprocessing

To ensure input consistency, we apply a standardized preprocessing pipeline. All depth values are unified to millimeters (mm). Frames are normalized by clipping values to the [0.5, 99.5] percentile range of non-zero pixels and linearly scaling them to 8-bit [0, 255]. An alternative "cutting" method, which sets out-of-range values to 0, is evaluated for domain adaptation. Processed frames are replicated into a three-channel grayscale format and encoded as .avi files using the `FFV1` lossless codec to maintain compatibility with pretrained RGB-based HAR models.

1.3. Similarity Metric Calculation

To quantify the domain gap between paired real and synthetic videos, we report globally aggregated metrics for each dataset. Pixel-wise errors (RMSE, AbsRel, and δ_τ) are computed across all pixels in the dataset to ensure uniform contribution and avoid bias from varying video lengths. SSIM is calculated at the frame level and reported as a dataset-wide mean. Finally, the 1-Wasserstein distance is calculated per video, capturing individual depth distributions, and averaged across the entire dataset.

1.4. Models

Our implementation relies on PyTorch [1] and Hugging Face Transformers [2]. We adapt the VideoMAE fine-tuning scripts[1] for VideoMAE, VideoMAE V2, and V-JEPA 2. We generate synthetic depth using Video Depth Anything[2], configured for metric depth estimation.

1.5. Data Augmentation

During training, we apply temporal and spatial augmentations: a fixed-duration clip is randomly sampled from each video, followed by random short-side scaling and cropping. For validation and testing, we use a clip sampled from the center of each video, applying resizing without spatial augmentations.

1.6. Training Hyperparameters

To ensure reproducibility, we use two hyperparameter configurations (Table 2) based on the dataset scale. All settings are derived from preliminary tuning, and all experiments use a fixed random seed of 42.

Table 2. **Training hyperparameters for small- and large-scale datasets.**

Hyperparameter	Small datasets	Large datasets
Learning rate	5e-5	3e-5
Number of epochs	100	50
LR Scheduler	Linear	Linear
Warmup ratio	0.1	0.1
Weight decay	0.0	0.01
Early stopping patience	15	10
Early Stopping threshold	0.01	0.01
Input sample rate	8	8

To accommodate memory constraints, V-JEPA 2 and VideoMAE V2 are trained with a batch size of 2, while VideoMAE uses a batch size of 8. For V-JEPA 2 and VideoMAE V2, we employ 4-step gradient accumulation to maintain a consistent effective batch size and enable mixed-precision (FP16) training for stability. We monitor validation loss during training and retain the checkpoint with the best performance for final evaluation.

[1] https://github.com/MCG-NJU/VideoMAE, last accessed: 28.09.2025

[2] https://github.com/DepthAnything/Video-Depth-Anything, last accessed: 27.07.2025

Table 3. **Within-dataset performance on UTKinect.**

ID	Config. (train data)	VideoMAE		VideoMAE V2		V-JEPA 2	
		Accuracy	F1	Accuracy	F1	Accuracy	F1
1	Full real-depth reference ($\alpha = 1$)	0.90	0.89	0.93	0.93	0.96	0.96
2	Baseline ($\alpha = 0.25$)	0.79 ± 0.08	0.76 ± 0.09	0.89 ± 0.02	0.88 ± 0.03	0.78 ± 0.08	0.75 ± 0.10
3	Domain gap ($\alpha = 0$)	0.59	0.52	0.74	0.75	0.91	0.91
4	Mixed training ($\alpha = 0.25$)	0.87 ± 0.02	0.87 ± 0.02	0.94 ± 0.02	0.94 ± 0.02	0.90 ± 0.04	0.90 ± 0.04

Table 4. **Within-dataset performance on UTD-MHAD.**

ID	Config. (train data)	VideoMAE		VideoMAE V2		V-JEPA 2	
		Accuracy	F1	Accuracy	F1	Accuracy	F1
1	Full real-depth reference ($\alpha = 1$)	0.93	0.93	0.99	0.99	0.95	0.95
2	Baseline ($\alpha = 0.2$)	0.74 ± 0.10	0.74 ± 0.09	0.92 ± 0.07	0.92 ± 0.07	0.87 ± 0.12	0.85 ± 0.15
3	Domain gap ($\alpha = 0$)	0.31	0.25	0.74	0.71	0.89	0.88
4	Mixed training ($\alpha = 0.2$)	0.89 ± 0.03	0.89 ± 0.03	0.96 ± 0.01	0.96 ± 0.01	0.96 ± 0.07	0.97 ± 0.05

1.7. Hardware and Computational Cost

NVIDIA A40 (48 GB) GPUs are used for small-scale dataset experiments with VideoMAE and VideoMAE V2, while V-JEPA 2 and all large-scale dataset runs are performed on NVIDIA A100 (80 GB) GPUs.

Synthetic depth generation using Video Depth Anything is performed on NVIDIA A100 (80 GB) GPUs. Tables 5 and 6 report the generation time and storage requirements. Generation times vary depending on the total number of frames per dataset, GPU type, and overall cluster load during processing. The preprocessing step (described in Section 1.2) significantly reduces storage requirements from raw outputs (NTU RGB+D: 2.82 TB $\rightarrow$ 101.0 GB).

Table 5. Total generation time for synthetic depth using Video Depth Anything on NVIDIA A100 (80 GB) GPUs. Times vary based on total frame count, GPU load, and cluster availability.

Dataset	Videos	Generation Time (HH:MM:SS)
UTKinect	140	00:31:28
UTD-MHAD	223	01:07:19
PKU-MMD	2,291	19:59:04
NTU RGB+D	12,956	128:06:57

Table 6. Storage requirements for synthetic depth before and after preprocessing.

Dataset	Videos	Raw Size	Preprocessed Size
UTKinect	140	5.54 GB	168 MB
UTD-MHAD	223	8.54 GB	576 MB
PKU-MMD	2,291	421.8 GB	18.0 GB
NTU RGB+D	12,956	2.82 TB	101.0 GB

2. Additional Tables for Within-Dataset Experiments

Tables 3, 4, 7 and 8 report detailed within-dataset performance for all four datasets, including accuracy and F1 scores across all training configurations. The four configurations follow the protocol defined in the main paper: config. 1 establishes the target performance using full real depth; config. 2 defines the low-data baseline; config. 3 quantifies the domain gap under direct synthetic transfer; and config. 4 evaluates whether synthetic data as augmentation recovers the performance lost under data scarcity. Across all datasets, *mixed training* (config. 4) consistently closes the gap between the low-data baseline and the full real-depth reference. For NTU RGB+D, we additionally report results for $\alpha = 0.05$ to examine a more challenging low-data regime. This is motivated by V-JEPA 2's high data efficiency at $\alpha = 0.25$, where its baseline (0.80) already approaches the full real-depth reference (0.81), leaving little room to demonstrate the benefit of synthetic augmentation. At $\alpha = 0.05$, the gap widens and the value of mixed training becomes more apparent: for VideoMAE, mixing synthetic data with 5% real data improves accuracy from 0.13 to 0.60, and for V-JEPA 2, from 0.57 to 0.77.

3. Measuring Domain Gap with Thresholding

Table 9 reports results for direct transfer (config. 5), where models are trained exclusively on thresholded synthetic data ($\alpha = 0$) and evaluated on real depth, using the best-performing thresholding strategy per dataset: *Cut to ratio* for small-scale and *Clip to ratio* for large-scale datasets. For small-scale datasets, thresholding improves direct transfer, with VideoMAE gaining +0.22 on UTKinect and +0.58

Table 7. **Within-dataset performance on PKU-MMD.**

ID	Config. (train data)	VideoMAE		VideoMAE V2		V-JEPA 2	
		Accuracy	**F1**	**Accuracy**	**F1**	**Accuracy**	**F1**
1	Full real-depth reference ($\alpha = 1$)	0.48	0.45	0.57	0.56	0.62	0.59
2	Baseline ($\alpha = 0.25$)	0.30 ± 0.01	0.25 ± 0.02	0.44 ± 0.02	0.41 ± 0.03	0.45 ± 0.03	0.39 ± 0.02
3	Domain gap ($\alpha = 0$)	0.11	0.09	0.50	0.47	0.59	0.57
4	Mixed training ($\alpha = 0.25$)	0.43 ± 0.02	0.40 ± 0.02	0.57 ± 0.02	0.55 ± 0.03	0.67 ± 0.03	0.65 ± 0.04

Table 8. **Within-dataset performance on NTU RGB+D.**

ID	Config. (train data)	VideoMAE		VideoMAE V2		V-JEPA 2	
		Accuracy	**F1**	**Accuracy**	**F1**	**Accuracy**	**F1**
1	Full real-depth reference ($\alpha = 1$)	0.70	0.71	0.75	0.75	0.81	0.81
2	Baseline ($\alpha = 0.25$)	0.58 ± 0.01	0.58 ± 0.01	0.70 ± 0.01	0.70 ± 0.02	0.80 ± 0.05	0.80 ± 0.05
	Baseline ($\alpha = 0.05$)	0.13 ± 0.19	0.12 ± 0.20	0.58 ± 0.02	0.58 ± 0.02	0.57 ± 0.02	0.56 ± 0.02
3	Domain gap ($\alpha = 0$)	0.29	0.29	0.63	0.64	0.78	0.79
4	Mixed training ($\alpha = 0.25$)	0.69 ± 0.02	0.69 ± 0.02	0.74 ± 0.04	0.74 ± 0.04	0.82 ± 0.09	0.82 ± 0.09
	Mixed training ($\alpha = 0.05$)	0.60 ± 0.01	0.60 ± 0.01	0.68 ± 0.02	0.68 ± 0.02	0.77 ± 0.09	0.76 ± 0.09

on UTD-MHAD. For large-scale datasets, thresholding provides no benefit or degrades performance, except for V-JEPA 2 on PKU-MMD with a gain of 0.03.

Table 9. **Domain gap with input-level adaptation (config. 5).** Baseline (no threshold) vs. best thresholding strategy under direct transfer, where models are trained on 100% synthetic data and evaluated on real depth. Best strategy for small-scale datasets: *Cut to ratio*; for large-scale: *Clip to ratio*.

Model	TH	UTKinect	UTD-MHAD	PKU-MMD	NTU RGB+D
VideoMAE	None	0.59	0.31	**0.11**	**0.29**
	Best	**0.81**	**0.89**	0.09	0.14
VideoMAE V2	None	0.74	0.74	**0.50**	0.63
	Best	**0.93**	**0.95**	0.49	0.63
V-JEPA 2	None	0.91	0.89	0.59	**0.78**
	Best	**0.94**	**0.99**	**0.62**	0.76

4. Cross-Dataset Generalization with Thresholding

Table 10 reports cross-dataset generalization results with thresholding (config. 8) for small-scale datasets, comparing the no-threshold baseline against the best-performing strategy (*Cut to Kinect range*) in the cross-dataset setting.

Unlike the within-dataset domain adaptation setting where thresholding improved performance on small-scale datasets, the cross-dataset mixed setting shows inconsistent effects: thresholding reduces performance for VideoMAE and VideoMAE V2 on both target datasets, while V-JEPA 2 shows a marginal gain on UTD-MHAD. This suggests that when real target data is already present in training (mixed setting), thresholding the synthetic source data offers limited additional benefit. Given these inconsistent results, thresholding experiments for large-scale cross-dataset settings were not conducted.

Table 10. **Cross-dataset generalization with thresholding.** Baseline (no threshold) vs. best thresholding strategy for cross-dataset generalization on small-scale datasets (Cut to Kinect range).

Model	TH	UTKinect (target)	UTD-MHAD (target)
VideoMAE	None	$\mathbf{0.84 \pm 0.04}$	$\mathbf{0.85 \pm 0.02}$
	Best	0.79 ± 0.04	0.82 ± 0.11
VideoMAE V2	None	$\mathbf{0.89 \pm 0.01}$	$\mathbf{0.91 \pm 0.03}$
	Best	0.85 ± 0.04	0.90 ± 0.08
V-JEPA 2	None	$\mathbf{0.88 \pm 0.01}$	0.87 ± 0.01
	Best	0.83 ± 0.05	$\mathbf{0.91 \pm 0.07}$

5. Ablation Study

We ablate the fine-tuning strategy by comparing *mixed training* and *sequential fine-tuning*, as shown in Table 11. Mixed training outperforms sequential fine-tuning for VideoMAE (0.87 vs 0.85 on UTKinect and 0.89 vs 0.82 on UTD-MHAD) and VideoMAE V2 (0.94 vs 0.92 on UTKinect and 0.96 vs 0.95 on UTD-MHAD), while results are comparable for V-JEPA 2. This suggests that VideoMAE and VideoMAE V2 benefit from joint optimization on both modalities, while V-JEPA 2's robust features transfer effectively regardless of fine-tuning strategy. Given its simplicity (single training stage) and consistent performance across all three backbones, we adopt mixed training as our default approach.

Table 11. **Fine-tuning strategy ablation.** Accuracy on real test sets comparing mixed and sequential training strategies ($\alpha = 0.25$ for UTKinect, $\alpha = 0.20$ for UTD-MHAD).

Model	Strategy	UTKinect	UTD-MHAD
VideoMAE	Mixed ($\alpha + (1-\alpha)$)	$\mathbf{0.87 \pm 0.02}$	$\mathbf{0.89 \pm 0.03}$
	Seq. ($(1-\alpha) \to \alpha$)	0.85 ± 0.07	0.82 ± 0.03
VideoMAE V2	Mixed ($\alpha + (1-\alpha)$)	$\mathbf{0.94 \pm 0.02}$	$\mathbf{0.96 \pm 0.01}$
	Seq. ($(1-\alpha) \to \alpha$)	0.92 ± 0.01	0.95 ± 0.01
V-JEPA 2	Mixed ($\alpha + (1-\alpha)$)	0.90 ± 0.04	0.96 ± 0.07
	Seq. ($(1-\alpha) \to \alpha$)	$\mathbf{0.93 \pm 0.01}$	$\mathbf{0.97 \pm 0.04}$

References

[1] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*, 2017. 1

[2] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020. 1