Supplementary Material for "Polyphony: Diffusion-based Dual-Hand Action Segmentation with Alternating Vision Transformer and Semantic Conditioning"

Hao Zheng[1] Hu Wang[2] Tiantian Zheng[1] Prajjwal Bhattarai[1] Tuka Alhanai[1]

[1]New York University Abu Dhabi, Abu Dhabi, United Arab Emirates
[2]Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates

{h.zheng, tz545, pb2276, tuka.alhanai}@nyu.edu hu.wang@mbzuai.ac.ae

This supplementary material provides comprehensive details and additional analyses to complement the main paper. It is organized as follows:

Section 1: Comprehensive implementation details including hyperparameters, training configurations, and dataset preprocessing procedures.

Section 2: Complete experimental results on HA-ViD and ATTACH, including additional baseline comparisons.

Section 3: Hand coordination accuracy analysis, demonstrating our method's superiority in modeling symmetric and asymmetric bimanual patterns.

Section 4: Detailed explanation and performance comparison of single-stream adaptation strategies.

Section 5: Comprehensive computational cost analysis, providing detailed breakdowns and fair comparisons with baseline methods.

Section 6: Specification of fine-grained action groups used to evaluate semantic disambiguation capability.

1. Additional Implementation Details

1.1. Detailed Hyperparameters

All experiments are conducted on a AMD Threadripper Pro 3395wx and a NVIDIA RTX A6000 GPU (48GB).

1.1.1. Stage 1: ADH-ViT Training

Table 1 presents the detailed hyperparameters for the ADH-ViT training.

1.1.2. Stage 2: Semantic Conditioning

Table 2 presents the detailed hyperparameters for the semantic conditioning stage.

1.1.3. Stage 3: Diffusion-based Action Segmentation

Table 3 presents the detailed hyperparameters for the diffusion-based dual-hand action segmentation.

Table 1. Detailed hyperparameters for ADH-ViT training.

Hyperparameter	Value
Backbone	VideoMAE V2 ViT-Base
Input resolution	224×224
Clip length (l_{clip})	16 frames
Tubelet size ($l \times p \times p$)	$2 \times 16 \times 16$
Embedding dimension (D)	768
Batch size per hand (B)	4
Alternation period (Δ)	50 steps
Total epochs	50
Optimizer	AdamW
Learning rate	1e-3
Layer-wise LR decay	0.9
Weight decay	0.1
LR scheduler	Cosine annealing scheduler
Warmup epochs	5
Data augmentation:	
Mixup	0.8
CutMix	1.0
Label smoothing	0.1
Random clips per video (n_{clip})	30 (HA-ViD), 20 (ATTACH), 5 (Breakfast)

Table 2. Detailed hyperparameters for semantic conditioning.

Hyperparameter	Value
Language model	MiniLM-L6-v2
Semantic embedding dim (D_{sem})	384
TCN layers (M)	3
TCN channels	[512, 128, 64]
TCN kernel size	3
Dilation rates	$[2^1, 2^2, 2^3]$
Dropout	0.5
Total epochs	100
Optimizer	AdamW
Learning rate	3e-4
Weight decay	1e-4
Batch size	8
Alignment loss weight (α)	0.7

1.2. Data Processing

Feature Extraction: For baseline comparisons using I3D features, we extract features from the pre-trained I3D model [1] on Kinetics-400 with input resolution 224×224 and

Table 3. Detailed hyperparameters for diffusion-based dual-hand action segmentation.

Hyperparameter	Value
Encoder layers (L_{seg})	10
Encoder channels	64
Encoder architecture	Mixed Conv-Attention
Decoder layers	3
Decoder architecture	Cross-attention Transformer
Feature fusion dim (D')	256
Diffusion steps (K)	1000
Diffusion schedule	Cosine
Noise scale (s_{de})	1.0
Offset (δ)	0.008
DDIM sampling steps (K')	5
Total epochs	1000
Optimizer	Adam
Learning rate	1e-3
Batch size	1 (full video)
Adaptive weighting window (w)	5
Performance gap threshold (Δ_{gap})	0.95
Weight bounds ($[\beta_{min}, \beta_{max}]$)	[1.0, 2.0]
Loss weights:	
Encoder CE (λ_{enc})	1.0
Decoder CE (λ_{dec})	15.0
Smoothness (λ_{sm})	0.15
Boundary (λ_{bd})	1.0

stride of 10 frames.

Action Description Construction: For each action class, we construct structured compositional descriptions following the format: "Action verb is [verb]; manipulated object is [object]; target object is [target]; tool is [tool]", following the definitions from HR-SAT [2]. When a component is not applicable, we use "null" as the value. The availability of action elements varies across datasets based on their annotation schemas: HA-ViD provides complete annotations for all four elements (verb, manipulated object, target object, tool); ATTACH annotations include action verb, manipulated object, and tool (manually parsed), with no target object component; Breakfast annotations contain action verb, manipulated object, and target object (manually parsed), with no tool component.

2. Additional Experimental Results

2.1. Additional Results on HA-ViD Dataset

Table 4 presents the complete results across all three viewpoints (side, front, top) of the HA-ViD [3] dataset, with additional experiments on DiffAct and FACT using MAS features.

Results with I3D Features. Using I3D features as input, our method achieves competitive performance across all views, with the best overall accuracy in side and front views. While performance differences with baselines are modest, our unified dual-hand model maintains more consistent performance across both hands. Furthermore, other methods train separate models for each hand, while our method simultaneously segments dual-hand actions using one model.

Results with MAS Features. To ensure fair comparison, we re-evaluate DiffAct and FACT using MAS features extracted from our trained ADH-ViT (Stage 1 and 2 features). With MAS features, all methods show substantial improvements over their I3D counterparts. Notably, our method achieves the best performance across most metrics, demonstrating that our contributions extend beyond feature engineering. Specifically, our method outperforms FACT with MAS by an average of 2.4 and 1.6 percentage points for left hand (LH) and right hand (RH) respectively, and surpasses DiffAct with MAS by 1.7 and 2.1 points. These improvements validate the effectiveness of our diffusion-based dual-hand action segmentation architecture with cross-hand feature fusion and adaptive loss weighting.

View-wise Analysis. The front view consistently achieves the highest accuracy (59.3% LH, 61.3% RH with MAS), likely due to optimal visibility of hand-object interactions from this perspective. Details about the multi-view camera setup are in [3].

2.2. Additional Results on ATTACH Dataset

Table 5 presents complete results on ATTACH with all evaluation metrics. With I3D features, our method achieves state-of-the-art performance across most metrics among I3D-based approaches.

With MAS features, our method achieves substantial improvements to 52.8% (LH) and 47.3% (RH), outperforming the best I3D baseline by +5.3 and +4.8 percentage points respectively. These gains extend across all metrics: F1@10 improves by 5.0/3.4 (LH/RH) points, F1@25 by 5.5/3.8 (LH/RH) points, and F1@50 by 3.9/3.3 (LH/RH) points over DiffAct. The consistent improvements across all metrics validate the effectiveness of our method.

2.3. Comparison with DuHa and DuCAS

DuHa [10] and DuCAS [11] also addressed the dual-hand action segmentation problem. However, they rely on the ground-truth bounding boxes of the objects. We evaluated our method on the same HA-ViD subset as DuHa [10] and DuCAS [11]. The results are shown in Table 6. It shows that our method produces more temporally coherent and balanced dual-hand segmentations, even without additional inputs. However, integrating object cues remains a promising future direction.

3. Hand Coordination Accuracy Analysis

A core motivation for Polyphony is to effectively model complex inter-hand coordination patterns. To directly evaluate this capability, we conduct a targeted analysis on HA-ViD front-view dataset, measuring how well different meth-

Table 4. Complete results on HA-ViD dataset across all three viewpoints. For fair comparison with our MAS features, we re-evaluate DiffAct and FACT using MAS features extracted from our trained ADH-ViT. Results show our method achieves best performance across all viewpoints and metrics. Results for MS-TCN, DTGRM, BCN, and C2F-TCN are from the original HA-ViD paper [3]; all other methods use official implementations under our experimental setup.

Method	View	Input	Left Hand					Right Hand				
			Acc	Edit	F1@10	F1@25	F1@50	Acc	Edit	F1@10	F1@25	F1@50
MS-TCN[4]	Side	I3D	37.6	37.4	41.2	/	/	31.1	32.5	37.4	/	/
	Front	I3D	35.2	36.3	38.8	/	/	36.7	36.2	39.3	/	/
	Top	I3D	37.1	38.9	40.4	/	/	36.1	35.6	41.3	/	/
DTGRM[5]	Side	I3D	38.5	36.5	40.9	/	/	35.9	35.2	37.6	/	/
	Front	I3D	38.5	37.2	39.0	/	/	38.8	39.6	40.5	/	/
	Top	I3D	40.4	38.8	40.8	/	/	38.7	37.0	41.2	/	/
BCN[6]	Side	I3D	43.1	40.4	43.7	/	/	38.6	36.3	42.4	/	/
	Front	I3D	44.4	43.1	44.4	/	/	41.3	37.0	44.0	/	/
	Top	I3D	43.5	40.7	44.3	/	/	44.0	40.7	43.7	/	/
C2F-TCN[7]	Side	I3D	18.0	20.7	38.2	/	/	21.7	20.9	38.8	/	/
	Front	I3D	23.3	21.4	39.1	/	/	20.5	22.2	38.8	/	/
	Top	I3D	26.6	24.0	41.1	/	/	25.4	22.7	39.5	/	/
DiffAct[8]	Side	I3D	42.2	34.4	37.1	30.9	19.4	43.4	46.29	47.92	40.61	27.02
	Front	I3D	43.1	45.7	46.7	40.3	28.0	45.0	47.07	47.82	41.67	26.99
	Top	I3D	45.8	47.5	48.6	43.1	30.5	44.8	50.60	50.66	44.87	31.29
FACT[9]	Side	I3D	44.8	49.4	50.8	45.0	30.5	42.1	46.0	46.4	38.0	24.4
	Front	I3D	44.7	46.2	48.3	42.0	29.1	44.1	44.0	47.0	39.4	26.7
	Top	I3D	45.7	46.1	51.3	42.1	29.1	45.2	45.8	49.3	43.4	30.1
Ours	Side	I3D	46.2	46.5	49.8	42.3	29.0	45.0	45.1	46.4	41.8	29.1
	Front	I3D	45.2	47.4	50.8	45.6	33.1	45.5	49.7	49.7	41.4	27.1
	Top	I3D	45.0	46.0	49.1	41.6	28.3	45.2	45.4	47.2	42.7	29.2
MS-TCN[4]	Side	MAS	52.1	46.0	49.3	43.4	30.0	54.7	45.5	49.8	43.1	29.2
	Front	MAS	52.7	45.2	50.6	43.7	29.1	56.7	46.3	50.5	43.3	30.7
	Top	MAS	54.5	48.8	52.0	47.2	33.7	56.9	47.4	50.3	44.3	30.9
DTGRM[5]	Side	MAS	52.1	27.8	29.8	26.3	18.8	55.3	28.1	29.7	25.4	16.8
	Front	MAS	54.3	27.3	31.4	26.9	18.8	56.9	32.7	36.6	31.4	22.5
	Top	MAS	55.0	41.8	44.4	39.5	28.7	55.2	39.8	40.7	34.7	24.1
BCN[6]	Side	MAS	52.2	51.7	52.5	47.5	36.7	53.5	51.4	52.3	46.6	37.2
	Front	MAS	52.3	48.6	49.5	46.3	36.7	54.5	50.6	52.7	47.1	37.8
	Top	MAS	52.3	49.7	51.9	47.3	38.8	55.8	49.4	52.4	47.7	37.4
C2F-TCN[7]	Side	MAS	54.1	39.9	44.1	39.1	28.0	56.7	37.7	43.9	38.3	27.4
	Front	MAS	55.4	37.1	45.1	40.7	30.8	57.9	41.8	47.3	40.9	29.2
	Top	MAS	55.7	39.4	47.0	42.1	31.2	57.8	42.2	46.9	41.5	29.2
DiffAct[8]	Side	MAS	55.6	**54.9**	57.2	**52.0**	38.3	57.3	51.5	57.4	49.7	36.1
	Front	MAS	54.2	**55.1**	59.6	52.7	38.9	58.2	53.3	59.9	53.5	38.6
	Top	MAS	**56.3**	**52.6**	58.3	**51.5**	**37.2**		55.8	61.2	55.8	43.9
FACT[9]	Side	MAS	56.3	52.1	**58.1**	50.7	39.4	58.1	52.2	57.2	50.9	36.6
	Front	MAS	53.4	53.3	60.4	53.1	40.3	57.0	52.0	58.3	53.2	39.2
	Top	MAS	54.4	52.4	56.6	50.0	35.1	61.9	**57.2**	62.3	**58.2**	44.6
Ours	Side	MAS	**56.7**	53.9	56.6	51.3	**40.1**	**58.1**	**53.0**	**58.2**	**51.5**	**37.6**
	Front	MAS	**59.3**	54.7	**61.4**	**56.4**	**42.8**	**61.3**	**55.0**	**63.0**	**56.4**	**41.1**
	Top	MAS	55.4	52.5	**58.4**	51.3	36.0	**62.5**	56.5	**63.4**	57.5	**45.2**

Table 5. Complete results on ATTACH dataset with all evaluation metrics. Our method with MAS features achieves substantial improvements over all I3D-based baselines across both hands and all metrics. All baselines use official implementations under our experimental setup.

Method	Input	Left Hand					Right Hand				
		Acc	Edit	F1@10	F1@25	F1@50	Acc	Edit	F1@10	F1@25	F1@50
MS-TCN[4]	I3D	43.5	46.2	38.8	29.2	13.1	36.6	46.7	37.7	27.7	13.7
C2F-TCN[7]	I3D	46.3	19.4	22.2	16.9	8.2	40.3	36.3	35.2	27.5	15.0
DiffAct[8]	I3D	47.5	44.1	40.7	31.4	15.3	42.5	46.9	43.5	33.9	17.5
FACT[9]	I3D	45.8	46.8	39.1	29.6	14.3	40.1	46.4	41.0	31.3	15.6
Ours	I3D	47.2	47.3	42.4	32.9	16.4	42.4	47.1	43.5	33.7	17.3
	MAS	**52.8**	**47.8**	**45.7**	**36.9**	**19.2**	**47.3**	**49.7**	**46.9**	**37.7**	**20.8**

ods predict two fundamental coordination modes that characterize bimanual activities.

Table 6. Comparison with DuHa and DuCAS. Averaged performance across three views.

Method	LH			RH		
	Acc	Edit	F1@10	Acc	Edit	F1@10
DuHa	51.7	32.1	36.0	31.5	31.7	34.7
DuCAS	**53.0**	29.8	31.9	34.6	28.0	29.5
Ours	42.3	**45.6**	**48.0**	**43.1**	**46.8**	**50.7**

Table 7. Bimanual coordination accuracy on HA-ViD front-view. Symmetric: both hands perform identical actions; Asymmetric: hands perform distinct actions simultaneously. Our method outperforms baselines in both coordination modes, with particularly strong improvements in the more challenging different-actions scenario.

Coordination Mode	Ours	DiffAct	FACT
Symmetric	**61.4**	58.0	54.6
Asymmetric	**20.5**	17.4	15.8

3.1. Coordination Metrics

We define two complementary coordination scenarios based on the relationship between left-hand and right-hand actions at each frame:

(1) Symmetric Coordination: Both hands perform identical actions simultaneously (including synchronized idle states). This represents mirrored bimanual coordination, which is common in assembly tasks when both hands manipulate the same component type. Formally, we compute accuracy over frames where $y_t^{LH} = y_t^{RH}$.

(2) Asymmetric Coordination: Hands perform distinct actions simultaneously, representing complementary coordination where hands have independent functional roles. This is common in manipulation tasks requiring complementary hand roles or concurrent parallel actions. We compute accuracy over frames where $y_t^{LH} \neq y_t^{RH}$.

3.2. Results and Analysis

Table 7 presents coordination accuracy on HA-ViD front-view using MAS features.

Symmetric Coordination. Our method achieves 61.4% accuracy, outperforming DiffAct by 3.4 percentage points and FACT by 6.8 points. This improvement validates that our unified modeling of dual-hand actions enables more consistent recognition of synchronized bimanual patterns.

Asymmetric Coordination. Our method achieves 20.5% accuracy, representing substantial improvements of 3.1 points (17.8% relative) over DiffAct and 4.7 points (29.7% relative) over FACT. This larger relative improvement in the more challenging different-actions scenario is particularly significant: it demonstrates that our method excels precisely where inter-hand coordination modeling is critical.

These results confirm that Polyphony successfully addresses its core design goal: modeling complex inter-hand dependencies while respecting the independence of each hand's action. Critically, this performance is achieved through a single unified model that processes both hands simultaneously, whereas baseline methods (DiffAct, FACT) require training separate models for each hand and selecting the best results for comparison. The fact that our unified approach outperforms these independent per-hand models, particularly in the challenging different-actions coordination mode, provides strong empirical evidence for our core hypothesis: that dual-hand action understanding benefits from architectures that mirror human perceptual processing, where bimanual coordination is perceived holistically rather than as two independent motor streams [12].

4. Single-Stream Adaptation Strategies

Our dual-hand framework can be adapted to single-stream temporal action segmentation through two strategies:

4.1. Strategy 1: Single-Stream Mode (Blocking)

We deactivate one hand-specific stream entirely, processing the input video through only the left-hand (or right-hand) stream. The model disables cross-hand feature fusion and adaptive weighting. This strategy represents the most direct adaptation but sacrifices the dual-stream coordination mechanisms.

4.2. Strategy 2: Dual-Stream Mode

To fully utilize our dual-stream architecture, we feed identical visual input to both hand streams. We implement this through two methods:

Method 2a: Direct Duplication. Both streams receive identical visual input and identical ground-truth labels ($Y^{LH} = Y^{RH} = Y^{gt}$). Cross-hand fusion operates normally.

Method 2b: Label Perturbation. Both streams process the same visual input, but one stream (LH) receives the original ground-truth labels while the other stream (RH) receives temporally perturbed labels. We apply boundary-shift perturbation to generate temporally misaligned label sequences to enable robust learning:

1. **Boundary identification**: Extract all internal segment boundaries $\{b_i\}_{i=1}^{B}$ from the ground-truth label sequence.
2. **Boundary filtering**: Remove boundaries adjacent to segments shorter than minimum length L_{min} to prevent creating invalid short segments.
3. **Random selection**: Randomly select up to N boundaries from the filtered set for perturbation.
4. **Temporal shifting**: For each selected boundary b_i, apply a random shift $\Delta_i \sim \mathcal{U}(-K, K)$ frames, subject to the constraint that resulting segments maintain a length of at least L_{min} frames.

Table 8. Comparison of single-stream adaptation strategies on Breakfast dataset with I3D features. Dual-stream mode with perturbation mechanism achieves the best performance across most metrics, demonstrating the benefit of controlled temporal misalignment for regularization.

Strategy	Acc	Edit	F1@10	F1@25	F1@50
Single-stream	77.8	**79.2**	81.3	77.4	67.1
Dual-stream (duplication)	78.2	78.9	80.8	77.2	67.4
Dual-stream (perturbation)	**78.6**	79.0	**81.9**	**77.8**	**68.3**

5. **Sequence reconstruction**: Generate the perturbed label sequence Y^{RH} with shifted boundaries while maintaining the original action class assignments.

During training, the left-hand stream uses original ground-truth labels ($Y^{LH} = Y^{gt}$) while the right-hand stream uses the temporally shifted labels ($Y^{RH} = \text{Perturb}(Y^{gt})$). The perturbation is controlled by three hyperparameters: maximum temporal shift K (frames), maximum number of perturbed boundaries N, and minimum segment length L_{min} (frames). For our experiments, we use $K = 10$, $N = 5$, and $L_{min} = 15$. At inference, perturbation is closed.

Table 8 compares the above three methods on Breakfast. Method 2b (perturbation) achieves the best overall performance, outperforming both single-stream and direct duplication methods on most metrics. This suggests that controlled temporal misalignment provides beneficial regularization, forcing the model to handle temporal variations. Cross-hand fusion can leverage complementary temporal contexts from differently supervised streams for more robust predictions. Method 2a (direct duplication) shows modest gains over single-stream (+0.4 accuracy), validating dual-stream coordination benefits even with redundant inputs.

Main paper results (Table 3 of the main paper) use Method 2b (label perturbation). Notably, even Strategy 1 (single-stream) achieves 77.8% accuracy, setting a new state-of-the-art among I3D-based baselines, validating our architecture design.

5. Computational Cost Analysis

Our method consists of three stages trained separately, each with distinct computational characteristics and input granularities. We provide detailed cost breakdowns and fair comparisons with baseline methods.

5.1. Detailed Computational Costs

Table 9 reports the computational costs for each stage of our pipeline, with different reporting granularities reflecting their distinct input characteristics.

Stage 1 (ADH-ViT): Operates on fixed-length 16-frame clips. We report costs per clip, as this is the atomic unit of processing. During training, approximately 46 clips are sampled from each HA-ViD video on average (combining segment-based and random clip sampling strategies, see Section 3.2.1 in the main paper). For dense feature extraction on untrimmed videos, we apply a sliding window with stride-1 and symmetric padding to extract frame-wise features for every frame, resulting in T forward passes for a T-frame video.

Stages 2 & 3: Process full untrimmed videos of variable length. We report costs for a representative 1227-frame video, which is the average video length in HA-ViD. Stage 2 (semantic conditioning) aligns visual features (768D) from stage 1 with semantic embeddings, while Stage 3 (diffusion-based action segmentation) performs temporal action segmentation on the concatenated MAS features (1227D = 768D motion + 75D action logits + 384D semantic).

5.2. Fair Comparison with Baselines

Table 10 compares the computational cost of our method with baseline temporal action segmentation methods on a 1227-frame video (HA-ViD average length).

Our unified dual-hand model demonstrates favorable cost-efficiency across multiple dimensions.

- **Parameters:** At 18.76M parameters, our model is more compact than doubled DiffAct (23.86M) and FACT (238.92M), though larger than lightweight MS-TCN (1.66M) and C2F-TCN (13.58M).
- **Computational cost:** Our 13.35 GFLOPs inference cost is 25% lower than DiffAct (17.96 GFLOPs) and 73% lower than FACT (50.28 GFLOPs), while achieving +13.4/+16.2 (LH/RH) and +12.0/+16.8 (LH/RH) points higher accuracy respectively, demonstrating that unified dual-hand modeling is more efficient and effective than separate per-hand approaches.
- **Feature representation:** When using the same Stage 3 architecture, our compact MAS features (1227D) reduce computational cost by 8.8% compared to I3D features (2048D) — from 14.64 to 13.35 GFLOPs — while improving accuracy by +11.7/+15.4 (LH/RH) points, establishing superior cost-performance efficiency.

5.3. Feature Extraction Cost Comparison

Table 11 compares the computational cost of I3D and MAS feature extraction (ADH-ViT + Semantic Conditioning).

For HA-ViD, I3D features are extracted using a PyTorch implementation[1] with 10-frame clips. Despite having only 14% of MAS's parameters (12.68M vs. 89.32M), I3D's computational cost is substantially higher because it requires both RGB feature extraction and optical flow computation via PWC-Net [13]. The optical flow computation dominates the overall cost, accounting for approximately 92.8% of the total computational burden of I3D.

[1] https://github.com/v-iashin/video_features

Table 9. Detailed computational costs of our three-stage pipeline. Stage 1 processes fixed-length clips (16 frames), with costs reported per clip; Stages 2-3 process full untrimmed videos, with costs reported for an example of 1227 frames (the average video length in HA-ViD).

Component	Parameters	Input	Inference GFLOPs	Training GFLOPs
Stage 1: ADH-ViT Feature Extraction (per clip)				
ViT-Base (per clip)	86.34M	16×224×224	157.78	473.33
Stage 2: Semantic Feature Conditioning (per video)				
TCN [512,128,64]	2.98M	1227×768	3.69	11.08
Stage 3: Diffusion-based Segmentation (per video)				
Encoder + Dual Decoders	18.76M	1227×1227	13.35	40.06
Total Parameters	**108.08M**	-	-	-

Table 10. Computational cost comparison for a 1227-frame video (HA-ViD average length). Baselines require separate models per hand (×2). Our Stage 3 cost is directly comparable to baseline segmentation costs when pre-extracted features are provided. I3D features: 2048D; MAS features: 1227D.

Method	Parameters	Input	Inference GFLOPs	Training GFLOPs
Baseline Methods (separate per-hand models)				
MS-TCN (I3D)	0.83M × 2	1227 × 2048	1.01 × 2	3.02 × 2
C2F-TCN (I3D)	6.79M × 2	1227 × 2048	5.07 × 2	15.22 × 2
DiffAct (I3D)	11.93M × 2	1227 × 2048	8.98 × 2	26.94 × 2
FACT (I3D)	119.46M × 2	1227 × 2048	25.14 × 2	75.43 × 2
Our Method (unified dual-hand model)				
Stage 3 only† (I3D)	18.97M	1227 × 2048	14.64	43.93
Stage 3 only† (MAS)	18.76M	1227×1227	13.35	40.06

†Assumes pre-extracted MAS features are provided

Per-frame, ADH-ViT is 9.84× more efficient than I3D (9.86 vs. 97.0 GFLOPs/frame), as it operates directly on RGB frames without requiring separate optical flow computation. For a complete 1227-frame video, MAS feature extraction (Stages 1+2: ADH-ViT + semantic conditioning) costs 193.60 TFLOPs, which is 6.15× more efficient than I3D with optical flow (1,190.3 TFLOPs). This efficiency advantage, combined with +11.7/+15.4 (LH/RH) points accuracy improvement, demonstrates that MAS features provide superior cost-performance trade-offs.

6. Fine-Grained Action Groups for Semantic Disambiguation Analysis

Our semantic conditioning is designed to distinguish actions that are visually similar but semantically distinct—a common challenge in fine-grained action understanding. To directly evaluate this capability, we identify 8 action groups (43 actions total) from HA-ViD where actions within each group exhibit similar visual patterns but differ in some semantic attributes. Table 12 details these action groups. Performance on these challenging actions is reported in Table 8 of the main paper.

Table 11. Feature extraction costs. I3D includes optical flow computation (PWC-Net). MAS combines ADH-ViT and Semantic Conditioning (Stages 1+2).

Method	Parameters	Per-Clip GFLOPs	Per-Frame GFLOPs	Full Video (1227f) TFLOPs
I3D	12.68M	970.02	97.0	1,190.21
MAS (Stages 1+2)	(86.34+2.98)M	157.78 (Stages 1)	9.86 (Stages 1)	193.60 + 0.003
Ratio (I3D / MAS)	0.14×	6.15×	9.84×	6.15×

Table 12. Fine-grained action groups used for semantic disambiguation analysis (Table 8 in main paper). Each group contains actions with similar visual patterns but distinct semantic attributes.

Action Group	# Actions	Semantic Distinctions
Insert Gear/Shaft	3	Insert {gear, shaft} into {shaft, gear}
Insert Placer	2	Insert {small placer, large placer} into shaft
Screw Shaft	4	Screw {shaft} onto gear hole {1, 2} {with, without} wrench
Screw Nut	6	Screw nut onto {shaft, nut hole 5, bolt} {with, without} wrench
Screw Hex Screw	12	Screw hex screw into screw hole {1, 2, 3, 4} {with, without} {Hex screwdriver, Phillip screwdriver}
Screw Phillips	4	Screw phillips screw into {gear hole 3, hole 4} {with, without} Phillip screwdriver
Insert Hex Screw	6	Insert hex screw into {screw hole 1, screw hole 2, screw hole 3, screw hole 4, cylinder bracket}
Insert Cylinder	6	Insert cylinder base, cylinder bracket, cylinder cap into {ball set, cylinder bracket, cylinder base, cylinder cap}
Total	**43**	

References

[1] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, 2017. 1

[2] R. K.-J. Lee, H. Zheng, and Y. Lu, "Human-robot shared assembly taxonomy: A step toward seamless human-robot knowledge transfer," *Robotics and Computer-Integrated Manufacturing*, vol. 86, p. 102686, 2024. 2

[3] H. Zheng, R. Lee, and Y. Lu, "Ha-vid: A human assembly video dataset for comprehensive assembly knowledge understanding," in *Advances in Neural Information Processing Systems* (A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds.), vol. 36, pp. 67069–67081, Curran Associates, Inc., 2023. 2, 3

[4] Y. A. Farha and J. Gall, " MS-TCN: Multi-Stage Temporal Convolutional Network for Action Segmentation ," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 3570–3579, IEEE Computer Society, June 2019. 3

[5] D. Wang, D. Hu, X. Li, and D. Dou, "Temporal relational modeling with self-supervision for action segmentation," 2020. 3

[6] Z. Wang, Z. Gao, L. Wang, Z. Li, and G. Wu, "Boundary-aware cascade networks for temporal action segmentation," in *ECCV (25)*, vol. 12370 of *Lecture Notes in Computer Science*, pp. 34–51, Springer, 2020. 3

[7] D. Singhania, R. Rahaman, and A. Yao, "C2f-tcn: A framework for semi- and fully-supervised temporal action segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 11484–11501, 2023. 3

[8] D. Liu, Q. Li, A.-D. Dinh, T. Jiang, M. Shah, and C. Xu, " Diffusion Action Segmentation ," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Los Alamitos, CA, USA), pp. 10105–10115, IEEE Computer Society, Oct. 2023. 3

[9] Z. Lu and E. Elhamifar, "Fact: Frame-action cross-attention temporal modeling for efficient action segmentation," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18175–18185, 2024. 3

[10] H. Zheng, R. Lee, Y. Lu, and X. Xu, "Duha: a dual-hand action segmentation method for human-robot collaborative assembly," in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pp. 522–527, 2024. 2

[11] H. Zheng, R. Lee, H. Liang, Y. Lu, and X. Xu, "Ducas: a knowledge-enhanced dual-hand compositional action segmentation method for human-robot collaborative assembly," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7175–7180, 2024. 2

[12] S. P. Swinnen and N. Wenderoth, "Two hands, one brain: cognitive neuroscience of bimanual skill," vol. 8, no. 1, pp. 18–25. 4

[13] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," 2018. 5