Supplementary Material for "From Observation to Action: Latent Action-based Primitive Segmentation for VLA Pre-training in Industrial Settings"

Jiajie Zhang Sören Schwertfeger
School of Information Science and Technology
ShanghaiTech University, Shanghai, China
{zhangjj2023, soerensch}@shanghaitech.edu.cn

Alexander Kleiner
School of Automation
Hangzhou Dianzi University, China
alexander.kleiner@gmail.com

Contents

This supplementary material provides further details on our methodology, dataset, and experimental results, reinforcing the claims made in the main paper.

1. Methodology Details

1.1. Motion Tokenizer (M_θ) Architecture and Training

As stated in the main paper (Section 3.1), we provide a detailed breakdown of the Motion Tokenizer (M_θ). The architecture is visualized in Figure S1.

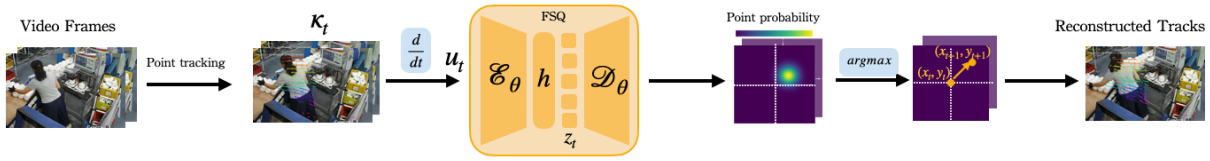
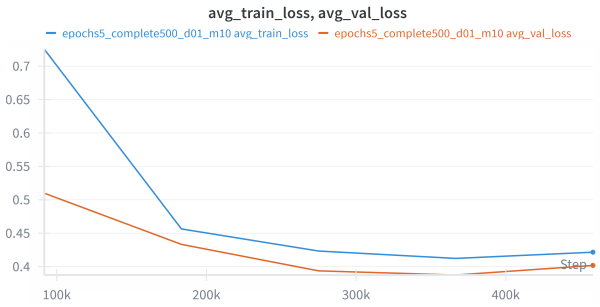


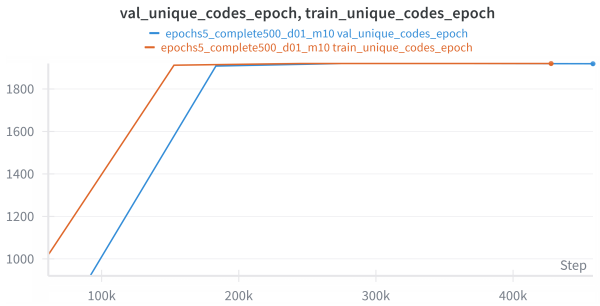
Figure S1. **Motion Tokenizer [2] (M_θ) Architecture.** This figure details the architecture introduced in Sec. 3.1. The model consists of a Transformer-based Encoder (E_θ) and Decoder (D_θ), with a Finite Scalar Quantization (FSQ) layer for discretization. The tokenizer is trained on keypoint velocities (derived from CoTracker) and uses a cross-entropy loss to predict the relative displacement for each track point, effectively modeling motion dynamics.

The tokenizer was trained until convergence, and its performance was validated on two primary metrics, as shown in Figure S2. We analyze both the training loss, to demonstrate stable convergence, and the FSQ codebook utilization, to confirm that the model learns a rich, non-collapsed vocabulary of motion dynamics. Key hyperparameters used for training and architecture are provided in Table S1.

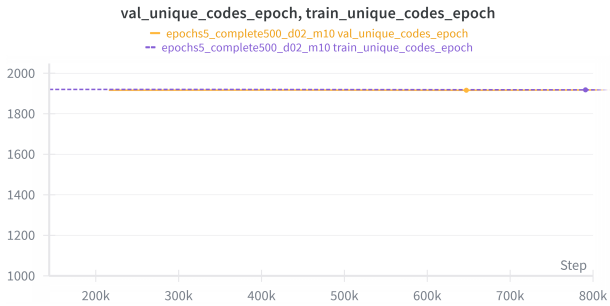


(a) Training/Validation Loss (Exocentric View)

(b) Training/Validation Loss (Top-down View)



(c) Codebook Utilization (Exocentric View)



(d) Codebook Utilization (Top-down View)

Figure S2. **Motion Tokenizer Training Analysis.** (a) Training and validation loss (Cross-Entropy Loss) for M_θ on the Exocentric view. (b) Training and validation loss on the Top-down view. The curves demonstrate stable convergence across both data sources. (c) FSQ codebook utilization analysis (Codebook Size=2048) for the Exocentric view. (d) FSQ codebook utilization for the Top-down view. High usage rates in both views confirm the model avoids "codebook collapse" and learns a rich, robust vocabulary of motion dynamics.

Table S1. **Motion Tokenizer Hyperparameters.** Key parameters used for training M_θ on Industrial Motor Assembly Dataset.

Parameter	Value
Keypoint Tracker	CoTracker [5]
Keypoints per window (N)	400
Track horizon (T)	16 frames
Encoder (E_θ) Layers	2 Transformer Layers
Encoder (E_θ) Heads	8 Attention Heads
Decoder (D_θ) Layers	4 Transformer Layers
FSQ Codebook	Size 2048
Training Dataset ($\mathcal{D}_{clips}$)	Unlabeled clips (40s per clips)
Batch Size	8
Learning Rate	0.0001
Optimizer	AdamW
Training Epochs	5

1.2. Action Segmentor Threshold Optimization

As described in Sec. 3.1.2, the primary threshold θ_{on} for our Action Segmentor is calibrated using a fully unsupervised, self-supervised process. We first generate a coarse "proxy signal" (e.g., temporal difference of keypoint velocities) and apply an automatic threshold (e.g., Otsu's method) to create binary pseudo-labels y_{pseudo}. Figure S3 visualizes this process.

We then perform a parameter sweep to find the θ_{on} for our E_{action} signal that maximizes the F1-score against these noisy pseudo-labels, as shown in Figure S4.

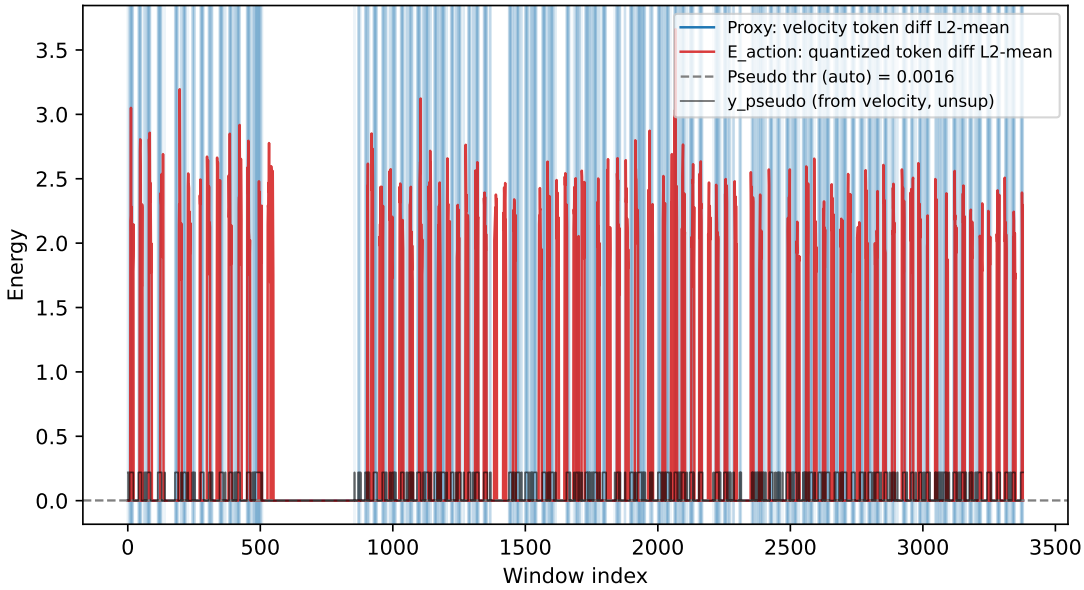


Figure S3. **Proxy Signal for Unsupervised Calibration.** This plot visualizes the first step of our unsupervised calibration procedure (Sec. 3.2.2). It shows: (1) The simple, low-level "velocity energy" (Proxy Signal, blue), (2) Our high-level E_{action} signal (red), and (3) The binary pseudo-labels y_{pseudo} (black steps). The y_{pseudo} labels are generated by applying a simple, automatic threshold (Otsu's method, gray dashed line) *only* to the low-level Proxy Signal. This plot demonstrates both the generation of these coarse target labels and the clear visual superiority (higher signal-to-noise ratio) of our E_{action} signal. These pseudo-labels are subsequently used as the optimization target to calibrate the threshold θ_{on} for E_{action}, as detailed in Figure S4.

Finally, we analyze the segmentor's sensitivity to the hysteresis ratio r (where $\theta_{off} = r \cdot \theta_{on}$) and the debounce windows u (for ON) and d (for OFF). Figure S5 shows that our segmentor is highly stable across a wide range of these secondary

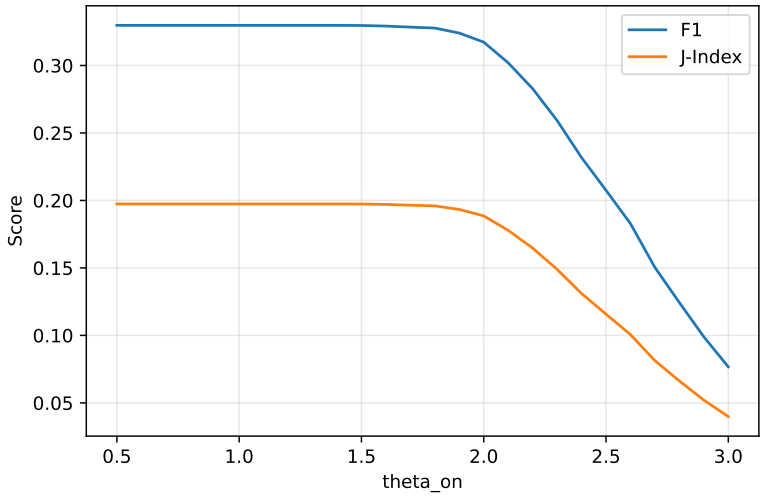


Figure S4. **Unsupervised Calibration of θ_{on}.** This plot visualizes the second and final step of our unsupervised calibration procedure. We plot the F1-score (Y-axis) that measures the agreement between our E_{action} signal (thresholded by θ_{on}) and the target y_{pseudo} labels (generated in Fig. S3). The X-axis represents the scanned θ_{on} value. The curve exhibits a distinct **plateau**, reaching a clear maximum F1-score (≈ 0.33) at $\theta_{on} \approx 0.5$ and maintaining this peak across a wide range (up to $\theta_{on} \approx 1.8$). This wide plateau confirms the robustness of our E_{action} signal; the final θ_{on} value can be reliably selected (e.g., the F1-maximizing $\theta_{on} = 0.5$) without sensitive fine-tuning.

parameters, making it well-suited for robust industrial deployment.

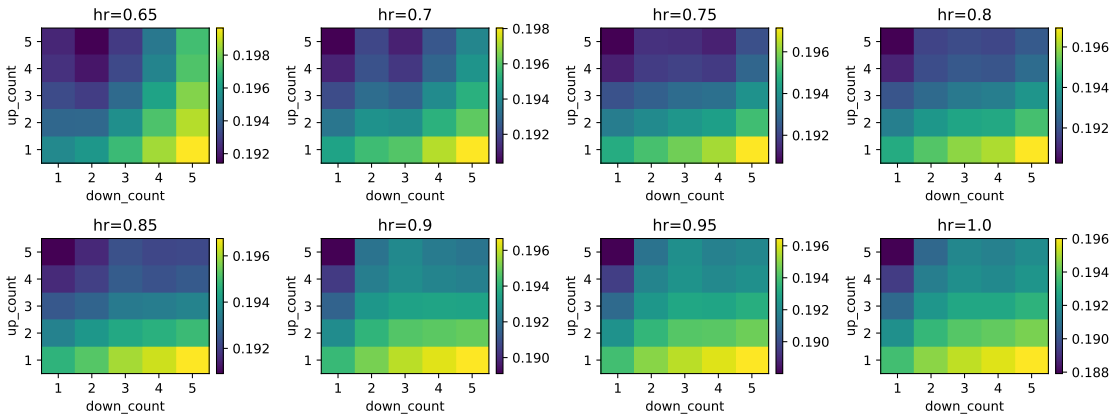


Figure S5. **Hysteresis and Debounce Sensitivity Analysis.** We analyze the F1-score (vs. Ground Truth) across a wide parameter sweep of the state-machine's secondary parameters: the hysteresis ratio r (where $\theta_{off} = r \cdot \theta_{on}$), the ON debounce u (up_count), and the OFF debounce d (down_count). Each subplot corresponds to a fixed r, while its axes represent u and d. The F1-score remains extremely stable across all tested parameter ranges (e.g., $r \in [0.65, 1.0]$, $u, d \in [1, 5]$), confirming the robustness of our segmentor design for industrial deployment.

1.3. Frozen Transformer Embedding Search

In Sec. 3.3.1, we selected Mean Pooling and a specific Transformer architecture ($L = 4, H = 4, d = 256$) for our training-free temporal embedding. Table S2 and Table S3 provide the experimental justification for these choices, based on unsupervised clustering metrics (Silhouette Score and Calinski-Harabasz Index).

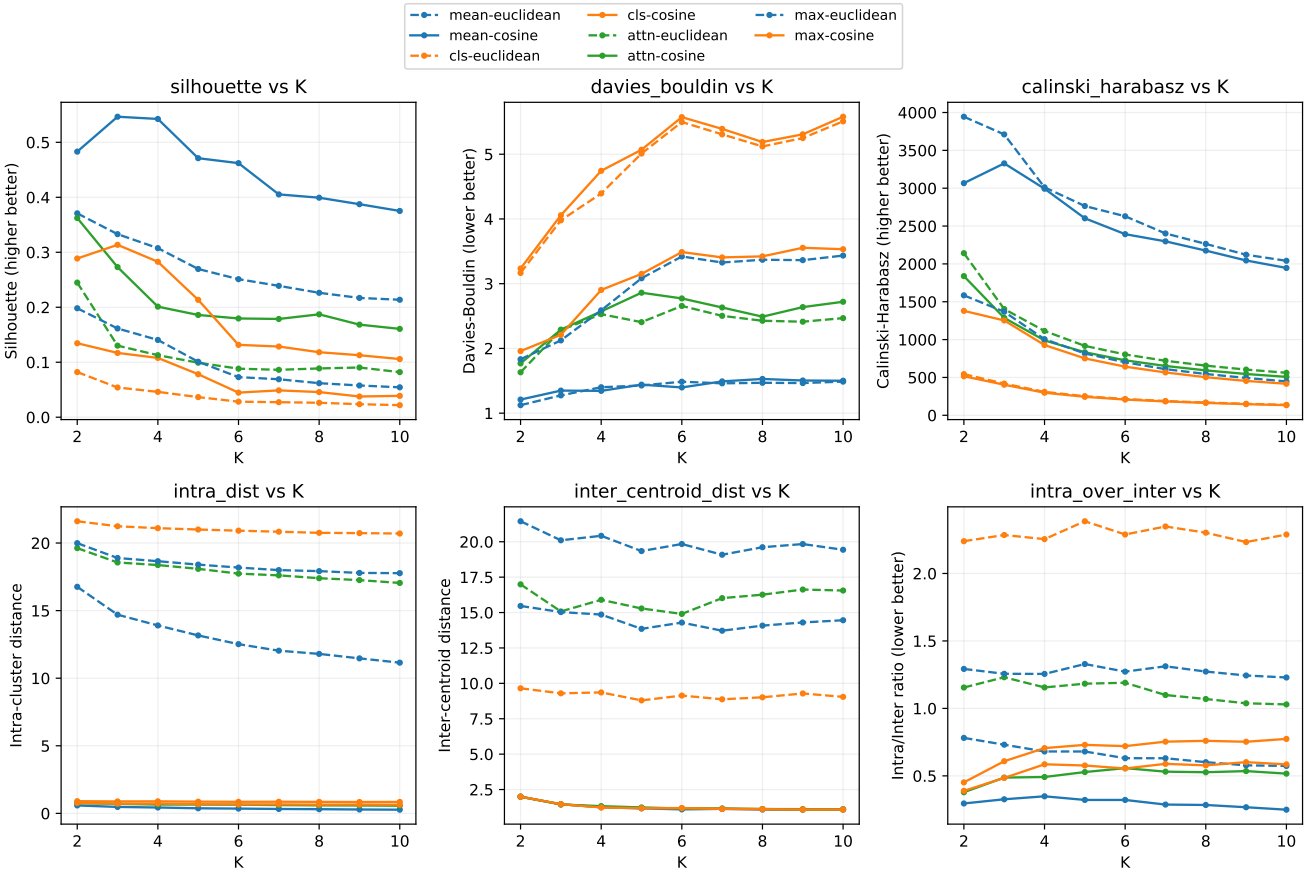


Figure S6. **Comprehensive Comparison of Pooling Strategies and Distance Metrics.** We conduct an exhaustive evaluation of different pooling strategies (color-coded: mean, cls, attn, max) and distance metrics (linestyle-coded: euclidean, cosine) for our Frozen Transformer embedding (Sec. 3.3.1). We plot six standard unsupervised clustering metrics against a K-range of [2, 10]. The results overwhelmingly show that **Mean Pooling** (blue lines) consistently and significantly outperforms all other strategies across all six metrics, producing embeddings with the highest inter-cluster separation (e.g., Silhouette, Inter-centroid dist) and intra-cluster compactness (e.g., Davies-Bouldin, Intra-dist). This provides robust justification for our choice of Mean Pooling.

Table S2. **Impact of Pooling Strategy on Clustering Quality (K=3).** As a summary of the comprehensive sweep shown in Figure S6, this table provides a snapshot of performance at $K = 3$ (our chosen K for the industrial dataset). Mean Pooling provides the best balance of compactness (Calinski-Harabasz) and separation (Silhouette), supporting our choice in Sec. 3.3.1.

Embedding Method	Silhouette Score ↑	Calinski-Harabasz Index ↑
Mean Pooling (Ours)	**0.547**	**3327.9**
CLS Token Pooling	0.135	515.6
Max Pooling	0.314	1253.0
Attention Pooling	0.362	1838.7

Table S3. **Impact of Frozen Transformer Architecture on Clustering Quality.** We evaluate frozen Transformer configurations varying the number of (L)ayers, (H)eads, and hidden (d)imension. For each configuration, we extract mean-pooled sequence embeddings (cosine distance) and run KMeans with $K \in [2, 10]$; we report the Silhouette Score and Calinski–Harabasz Index at the K that maximizes silhouette (higher is better). The highlighted setup (L=4, H=4, d=256) is adopted in our method and achieves strong performance on both metrics.

Architecture			Clustering Metrics	
L (Layers)	H (Heads)	d (Dimension)	Silhouette Score ↑	Calinski-Harabasz Index ↑
2	2	128	0.541	176.84
4	4	128	0.478	134.80
4	**4**	**256**	**0.600**	**4015.65**
6	8	256	0.514	164.10
6	8	512	0.524	161.44

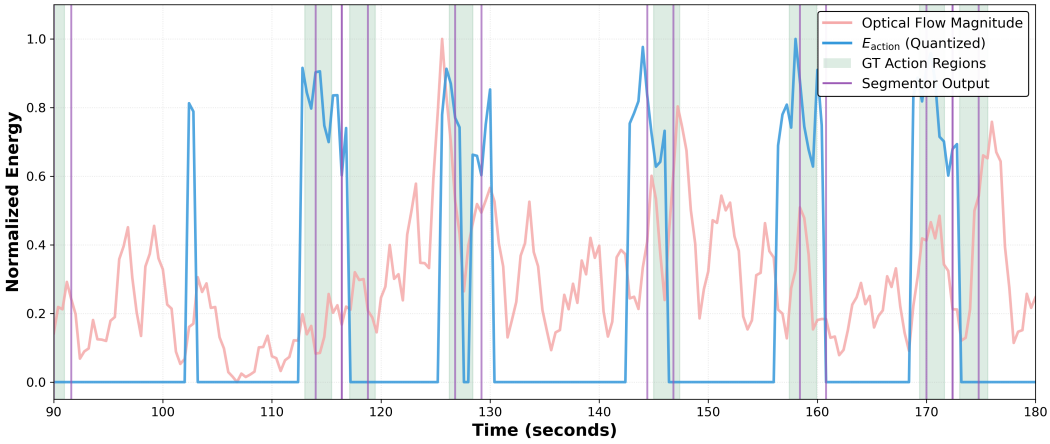
2. Extended Experimental Results

2.1. Extensive Qualitative Segmentation Results

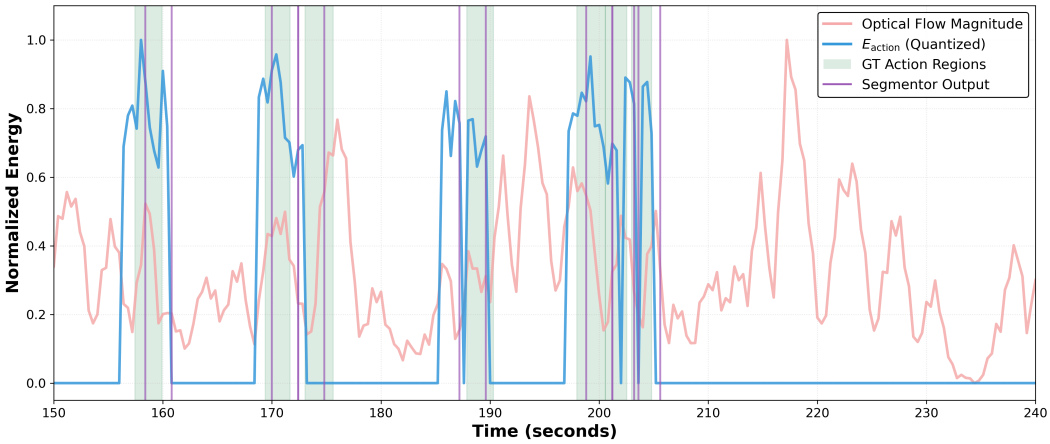
To further validate the superiority of our **Latent Action Energy** (E_{action}) signal (Sec 4.2), we provide extensive qualitative comparisons against the Optical Flow baseline. Figure S7 through Figure S8 show various segments from our test set. Across different actions, our E_{action} signal consistently exhibits clearer, more stable activations that align with semantic action boundaries, while the Optical Flow signal is noisy and correlates with low-level physical motion rather than task intent.

Figure S7 highlights the fundamental difference in signal quality between our approach and the baseline. The Optical Flow signal (red), being a low-level metric based purely on pixel changes, is highly volatile and reacts indiscriminately to *any* visual motion. Conversely, our E_{action} (blue) is derived from the latent space of the Motion Tokenizer (Sec. 3.1), which is trained to model semantic motion dynamics.

As the figure clearly demonstrates, E_{action} remains near baseline during non-semantic or minor movements, providing clear, sustained activations only during periods that the tokenizer recognizes as meaningful action. Consequently, our signal robustly and cleanly captures the true semantic action boundaries in both perspectives, demonstrating the effectiveness and superiority of our latent representation.

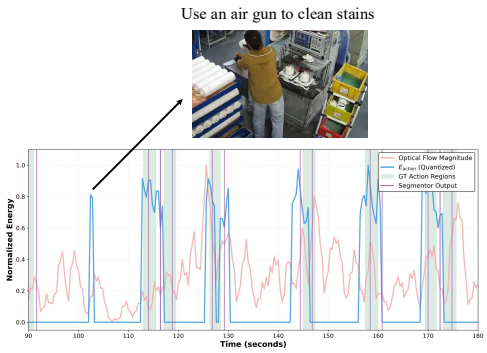


(a) Top-down View

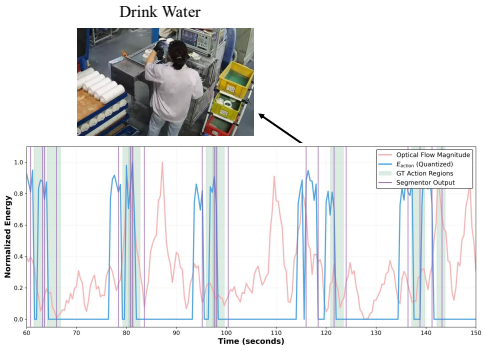


(b) Exocentric View

Figure S7. **Extended Qualitative Comparison.** Similar to Fig. 4 in the main paper, we provide qualitative results for both the (a) Top-down and (b) Exocentric views from our industrial dataset. In both plots, we show our **Latent Action Energy** E_{action} (blue), the **Optical Flow** baseline (red), Ground Truth Boundaries (green), and our Segmentor's output (purple).



(a) False Negative (missed detection). An action (e.g., "use an air gun") failed to generate enough latent energy to cross θ_{on}. In this case, the action's motion was largely occluded as the worker's back was turned to the camera, causing the motion tokenizer to register minimal dynamic change and thus low energy.



(b) False Positive. A rapid, non-task-related motion (e.g., worker drinks water) was incorrectly identified as an action primitive, introducing noise that violates our high-precision objective.

Figure S8. **Failure Case Analysis and Design Trade-offs.** Qualitative examples illustrating the system's behavior on challenging segments. **(Left)** Missed action due to heavy occlusion from the worker's back—an intentional precision–recall trade-off given abundant industrial data. **(Right)** Non-task motion (drinking water) falsely detected as an action primitive, representing a true limitation. These cases suggest future improvements including object-centric features or higher-level task grammar.

2.2. Qualitative Visualization of Discovered Action Clusters

To support our quantitative clustering results (Table 2, Table 3) and the UMAP visualization (Fig. 5), we provide direct qualitative evidence of the discovered clusters. Figure S9 show grids of randomly sampled primitives from each of the $K = 3$ clusters found by K-Means.

These figures visually confirm our core hypothesis: the unsupervised pipeline discovers a "countable" set of semantically coherent actions. The visual consistency within each cluster is high at a **coarse-grained semantic level**.

This observation is crucial: it validates that our embedding (Phase 3) successfully identifies the primary "verb" of the action (e.g., 'grasp', 'move'), creating semantically pure, high-level groupings. While it is less sensitive to fine-grained context (e.g., 'with which object'), for finer-grained semantic annotations, these well-segmented action clips generated by our LAPS pipeline can serve as high-quality inputs for automated labeling by a state-of-the-art VLM.

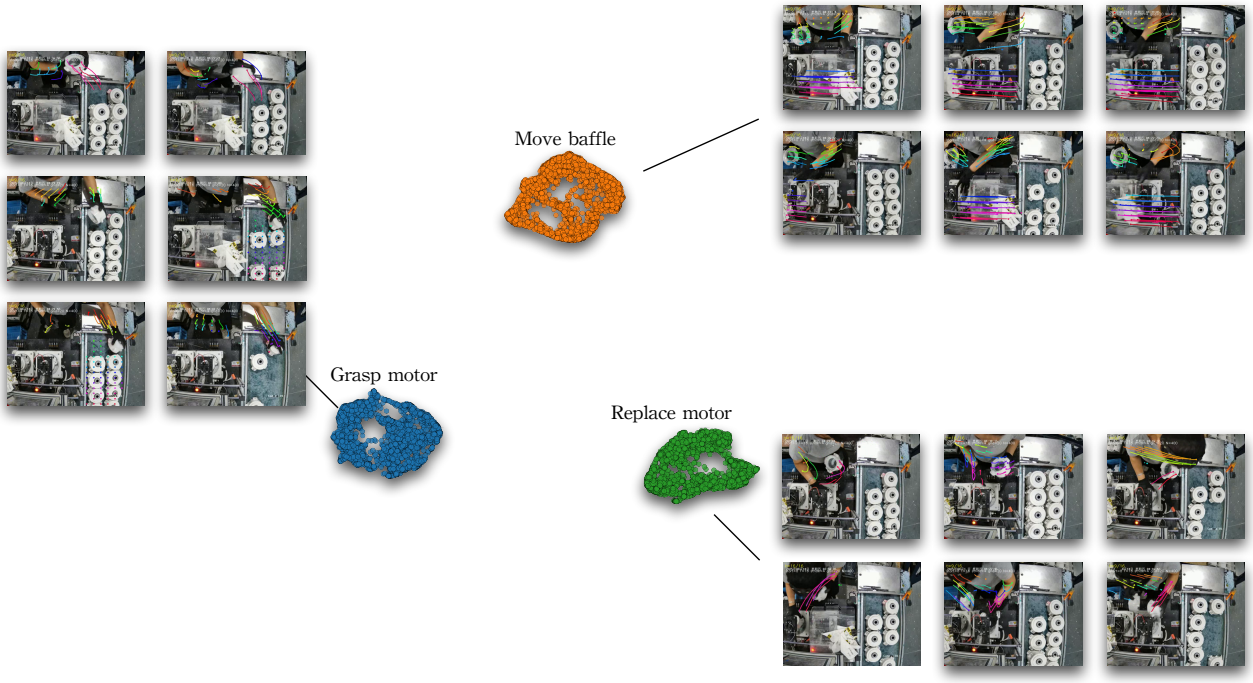
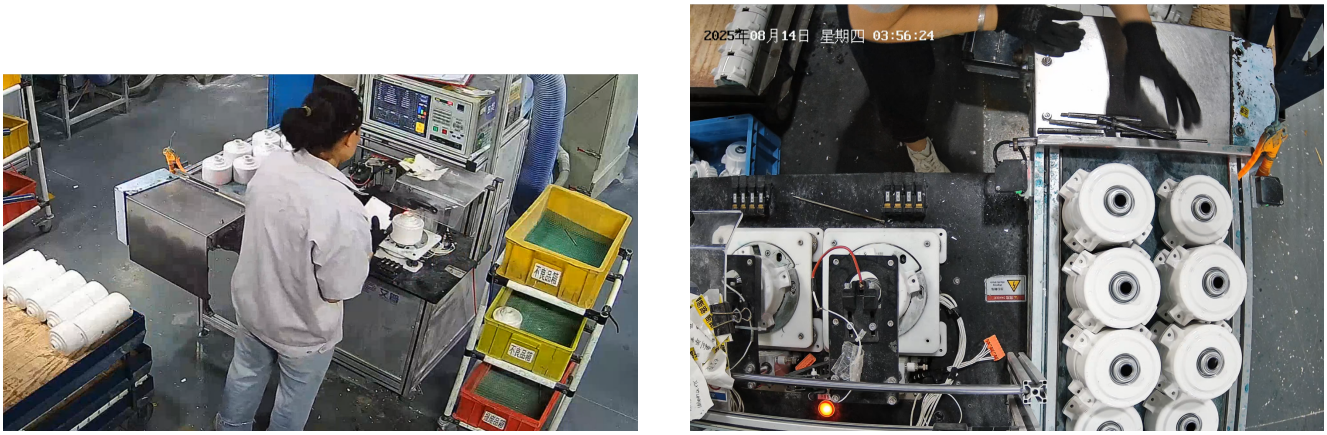


Figure S9. **Qualitative Visualization of Discovered Clusters (Top-down View).** This figure provides qualitative validation for the semantic coherence of action primitives discovered from the **Top-down View**. The central plot shows the UMAP embedding of all segmented primitives (Sec. 4.4), demonstrating distinct, well-separated clusters. Lines connect each abstract cluster to representative keyframes of its constituent primitives, visually confirming high intra-cluster similarity. As discussed in the main paper (Sec. 4.4), the K=3 clusters effectively disambiguate **coarse-grained action types**. Manual inspection confirms these clusters correspond to semantically similar groupings, such as **'Move Baffle'** (Cluster 1), **'Replace Motor'** (Cluster 2), and **'Grasp Motor'** (Cluster 3).

3. Dataset and Implementation Details

3.1. Industrial Dataset and Annotation

Our method was evaluated on a real-world industrial dataset collected from an electric motor assembly line, as described in Sec. 4.1. Figure S10 provides a visual overview of the data collection environment.



(a) Exocentric View (b) Top-down View

Figure S10. **Industrial Motor Assembly Workstation Setup.** This workstation facilitates a multi-stage process for household electric motor inspection and sorting: the worker picks a motor from the conveyor belt, conducts a visual appearance check, places it into an electrical testing rig for performance validation, and subsequently sorts the product into designated bins. The figure shows example frames from (a) the **Exocentric View**, which captures the global context of this entire process and the worker's full-body motion; and (b) the **Top-down View**, which captures the detailed, fine-grained hand manipulations involved in the task.

Table S4 provides a summary of the dataset statistics.

Table S4. **Dataset and Annotation Details.** Summary of our real-world industrial dataset (Sec. 4.1).

Property	Value
Environment	Electric Motor Assembly Line
Dataset Total Duration	∼10 hours
Annotated Test Duration	∼2 hours
Video Views	2 (Top-down, Exocentric)
Video Resolution	3840×2160 (Top-down); 1442×898 (Exocentric)
Video FPS	60 (Top-down); 25 (Exocentric)

3.2. Baseline Implementation Details

We provide implementation details for the baselines used in Table 1 and Table 2.

- **Optical Flow Baseline**: We computed dense optical flow using OpenCV's Dual TV-L1 method. The per-frame magnitude was averaged and fed into the *exact same* causal state-machine (Sec. 3.1.2) as our E_{action} signal, ensuring a fair comparison of the signals themselves.
- **ABD** [3]: We reproduced the offline ABD algorithm strictly following the paper (official code unavailable). The choice of input features varied by dataset. For the GTEA [4] and Breakfast [6] benchmarks, we utilized the available pre-computed I3D [1] features. However, for our large-scale Industrial Motor Assembly Dataset, extracting I3D features was computationally prohibitive. Consequently, for the industrial dataset, we utilized CPU-based HOF features (via OpenCV; 2.0 s window, 0.4 s stride, 16 orientation bins). This decision reflects a practical limitation of many real-world industrial settings, where leveraging expensive, pre-computed deep features (e.g., I3D) is often infeasible, especially for online segmentation scenarios. Given the respective clip-wise features $\mathbf{X} \in \mathbb{R}^{N\times D}$, the algorithm proceeds by applying temporal mean filtering with window $L \approx \alpha \cdot N/K$ ($\alpha = 0.5$), computing adjacent-frame cosine similarity, detecting change points by NMS on local minima with the same L, and refining segments via bottom-up merging by

cosine similarity until exactly K segments remain (K set to the dataset's average via "auto" mode). We export LAPS-compatible segments and evaluate with the same script as other methods; no supervision or tuning on test videos.

- **OTAS** [7]: We use the official open-source implementation (global "tf" variant): a ResNet-50 backbone with a Transformer encoder that predicts next-step features from a window of `seq_len`=5 frames sampled at `ds`=3. We train on the training split (Adam, $\mathrm{lr} = 10^{-4}$) and compute per-window MSE on test, form a negative mean-error series, then detect boundaries as local minima via `scipy.signal.argrelextrema` (`order`=15), re-align indices by offset $=$ `seq_len` $\times$ `ds`, and convert to seconds using video FPS. We do not use the object-aware or GAT variants; all other hyperparameters follow the defaults, and evaluation uses the same pipeline as our method.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 10

[2] Jeremy A Collins, Loránd Cheng, Kunal Aneja, Albert Wilcox, Benjamin Joffe, and Animesh Garg. Amplify: Actionless motion priors for robot learning from videos. *arXiv preprint arXiv:2506.14198*, 2025. 2

[3] Zexing Du, Xue Wang, Guoqing Zhou, and Qing Wang. Fast and unsupervised action boundary detection for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3323–3332, 2022. 10

[4] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*, pages 3281–3288. IEEE, 2011. 10

[5] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6013–6022, 2025. 3

[6] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. 10

[7] Yuerong Li, Zhengrong Xue, and Huazhe Xu. Otas: unsupervised boundary detection for object-centric temporal action segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6437–6446, 2024. 11