PROSKILL: Segment-Level Skill Assessment in Procedural Videos Supplementary Material

Michele Mazzamuto*
University of Catania, Italy

Daniele Di Mauro*
Next Vision s.r.l., Italy

Gianpiero Francesca†
Toyota Motor Europe, Belgium

Giovanni Maria Farinella†
University of Catania, Italy

Antonino Furnari†
University of Catania, Italy

Abstract

This supplementary document provides additional details on the construction of the PROSKILL *dataset, including the annotation protocol, the pairwise comparison workflow, and the aggregation strategy used to derive absolute skill scores. We further report implementation details for all benchmarked baselines and additional qualitative examples not included in the main paper. This material is intended to complement the main manuscript titled:* PROSKILL*: Segment-Level Skill Assessment in Procedural Videos. All data and code are available at* https://fpv-iplab.github.io/ProSkill/.

1. Amazon Mechanical Turk

This section will detail the design and implementation of our annotation tasks on Amazon Mechanical Turk (AMT). We describe the user interface shown to workers, the iterative round-based annotation workflow, quality control mechanisms including qualification and inter-HIT tests, and the technical tools used to automate task management and data processing.

1.1. User Interface

Each HIT on AMT displays a pair of videos side by side, both showing the same sub-action (e.g., *Clean the chain*). Annotators are instructed to compare the two performances and select the individual who demonstrates superior skill. See Figure 1 for an example of the annotation interface.

Prompt shown to workers:

> *Compare the two videos performing the action "Clean the chain" and select the better one. Identify which performer is better based on factors like Skill/Expertise, Proper Technique/Form, Confidence.*

Instructions summary:

- Workers must **watch both videos completely**.
- They are instructed not to base their judgment solely on execution speed.
- Emphasis is placed on evaluating *technique*, *confidence*, and *correctness*.
- After watching, workers select one of:
  - Video 1 (Left)
  - Video 2 (Right)

1.2. Task Design and Implementation

The annotation workflow was organized into iterative rounds. At the end of each round, we extracted pairs of videos based on the updated global ranking to generate the next set of HITs. Each pair consisted of two video URLs hosted on our own servers, which were embedded within the AMT interface to provide seamless playback and consistent presentation.

Workers were asked to compare each pair and select which video demonstrated better skill, or indicate if there was no noticeable difference. After completing the annotations for a round, the global ranking was updated according to the collected preferences, and new pairs were generated for the subsequent round accordingly.

The user interface was developed using HTML, CSS, and JavaScript embedded within AMT's custom templates. To ensure data quality, a qualification test using gold-standard pairs with known correct answers was required before

*Equal contribution.
†Co-Principal Investigator role.

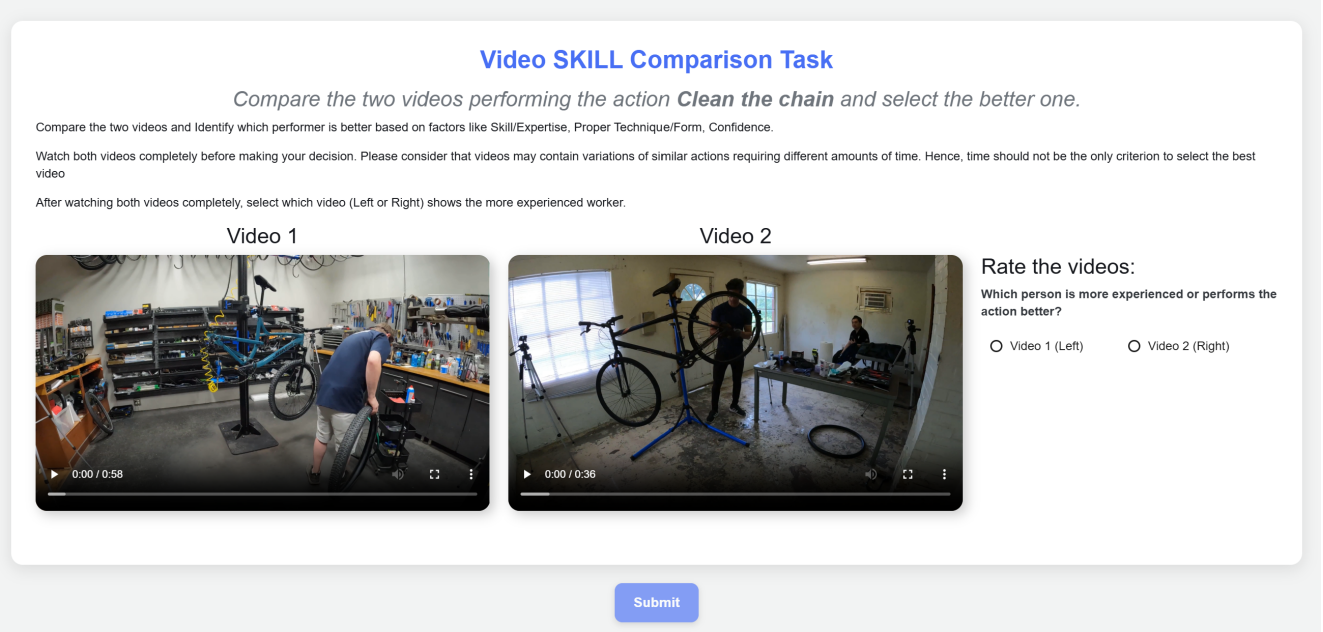


Figure 1. Screenshot of the annotation interface used in AMT. Annotators view two video segments corresponding to the same task and are asked to select which performer appears more skilled.

workers could participate. Additionally, inter-HIT quality controls were implemented by randomly inserting gold-standard pairs within HITs during the annotation process, allowing real-time monitoring of worker reliability and suspension of underperforming annotators.

Task management, including HIT creation, publication, and results retrieval, was automated through Python scripts using the AWS SDK for Python (`boto3`). These scripts enabled efficient batch processing and streamlined data handling.

Preprocessing scripts prepared video clips for annotation, while aggregation pipelines combined pairwise preferences into robust global rankings. All source code, including frontend templates and backend scripts, was maintained under version control in a dedicated Git repository to ensure reproducibility and ease of maintenance.

1.3. Annotation Statistics

We conducted a detailed analysis of the collected annotations across five datasets and present the results below.

Agreement Rate Distribution by Dataset The top-left boxplot in Fig. 2 illustrates the distribution of agreement rates among annotators for each dataset. Across the five datasets, agreement rates generally cluster around 0.7, with some variability. IKEA and EgoExo4D tend to have slightly higher median agreement, reflecting more consistent consensus among workers. Epic-Tents and Meccano show wider variability, indicating some tasks were more ambiguous or difficult to judge consistently. This plot helps visualize how reliably annotators agree on skill assessments depending on the dataset context.

Mean Agreement Rate by Dataset The top-center bar chart in Fig. 2 shows the average agreement rate for each dataset, summarizing annotator consensus in a single metric. IKEA scores the highest mean agreement (0.724), followed closely by EgoExo4D (0.716) and Assembly101 (0.705). Meccano and Epic-Tents have slightly lower means around 0.666 and 0.699 respectively. These differences may relate to task complexity, video quality, or inherent ambiguity in skill evaluation across different procedural tasks.

Distribution of Vote Differences The top-right histogram in Fig. 2 represents how many comparisons exhibit various levels of vote difference — the absolute difference in votes between the two videos compared. Most comparisons have vote differences centered around 2 or 3, indicating moderate consensus among annotators. Few comparisons have very low (0 or 1) or very high (5+) vote differences, highlighting that while some pairs are clear winners, others are more contested or balanced in skill demonstration.

Winner Balance by Dataset The bottom-left winner balance plot in Fig. 2 visualizes how wins are distributed between the two videos compared within each dataset. In-

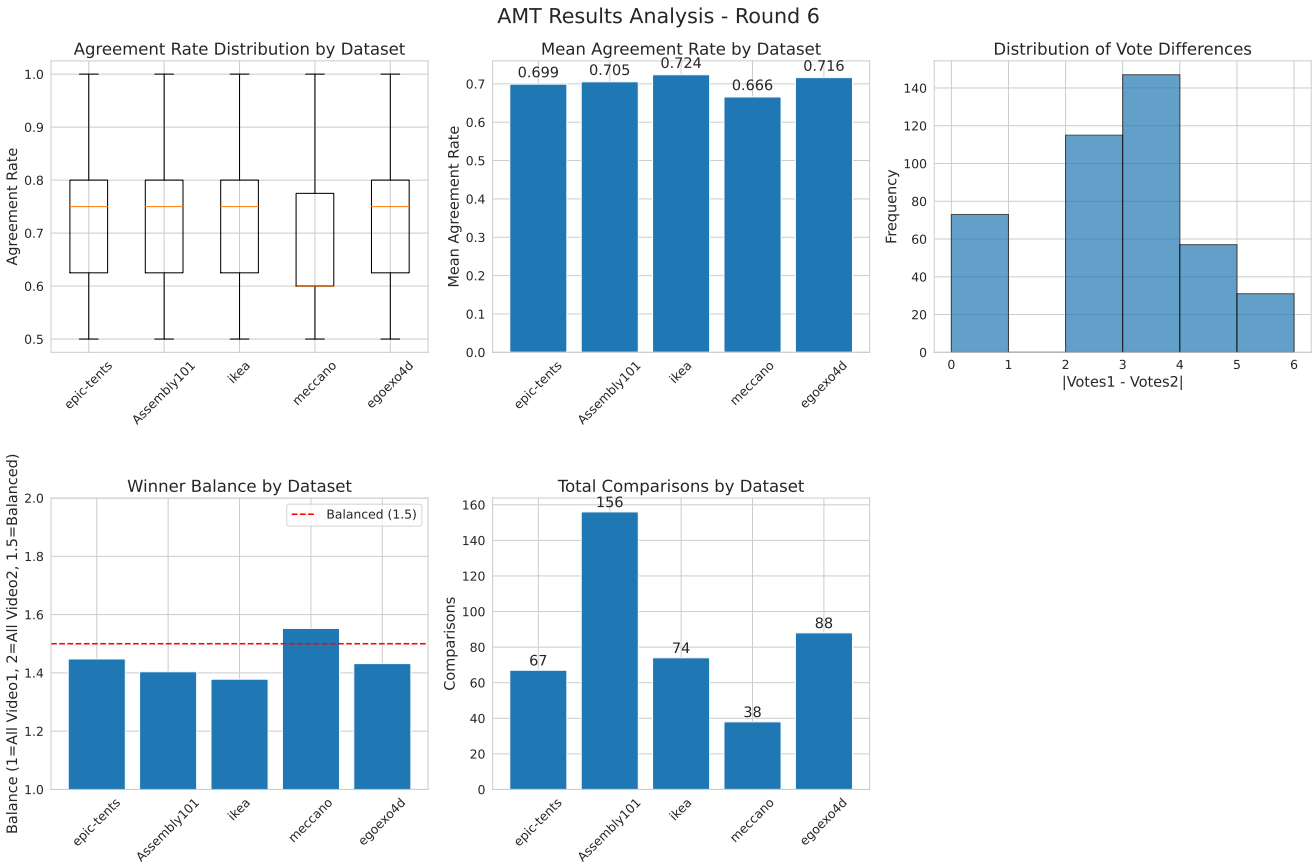


Figure 2. Analysis of AMT annotations across datasets. From left to right and top to bottom: (1) Agreement Rate Distribution, (2) Mean Agreement Rate, (3) Distribution of Vote Differences, (4) Winner Balance, (5) Total Comparisons.

stead of the previous ratio, the balance is now expressed on a scale from 1 to 2, where a value of 1 means all votes favored the first video, 2 means all votes favored the second video, and 1.5 represents a perfectly balanced split. Datasets like Epic-Tents, Meccano, and EgoExo show values close to 1.5, indicating relatively even competition between videos. Conversely, IKEA and Assembly101 display values skewed closer to 1, suggesting a tendency for the first video to be favored more often, which may reflect stronger or more consistent skill differences in these datasets.

Total Comparisons by Dataset The bottom-center bar plot in Fig. 2 reports the total number of pairwise comparisons collected for each dataset in round 6. Assembly101 contributes the majority with 156 comparisons, reflecting its larger dataset or annotation emphasis. IKEA and EgoExo4D follow with 74 and 88 comparisons respectively, while Epic-Tents and Meccano have fewer at 67 and 38. This distribution indicates the relative annotation effort and data availability across procedural video sources. Overall, we collected 16,372 unique comparisons, annotated by 551 qualified workers.

1.4. Labeling Time Analysis

In this section, we analyze the time required by annotators to complete each HIT. Understanding annotation duration provides insight into the complexity and cognitive load of the skill assessment task. To gain insights into the annotation effort required per action, we analyzed the average time spent by workers to complete each HIT, normalized by the average video clip length of the corresponding dataset. This provides a scale-invariant measure of annotation time, accounting for differences in video duration across datasets. We further rescaled these values between 0 and 1 for comparability.

As shown in Figure 3, the normalized annotation time varies considerably depending on the dataset and the complexity of the action. We report below the mean and standard deviation of normalized time for each action. In general, actions from the *Egoexo4D* dataset were annotated more quickly (e.g., *Remove inner tube*: 0.108 ± 0.083), likely due to higher visual clarity or simpler temporal dynamics. Con-

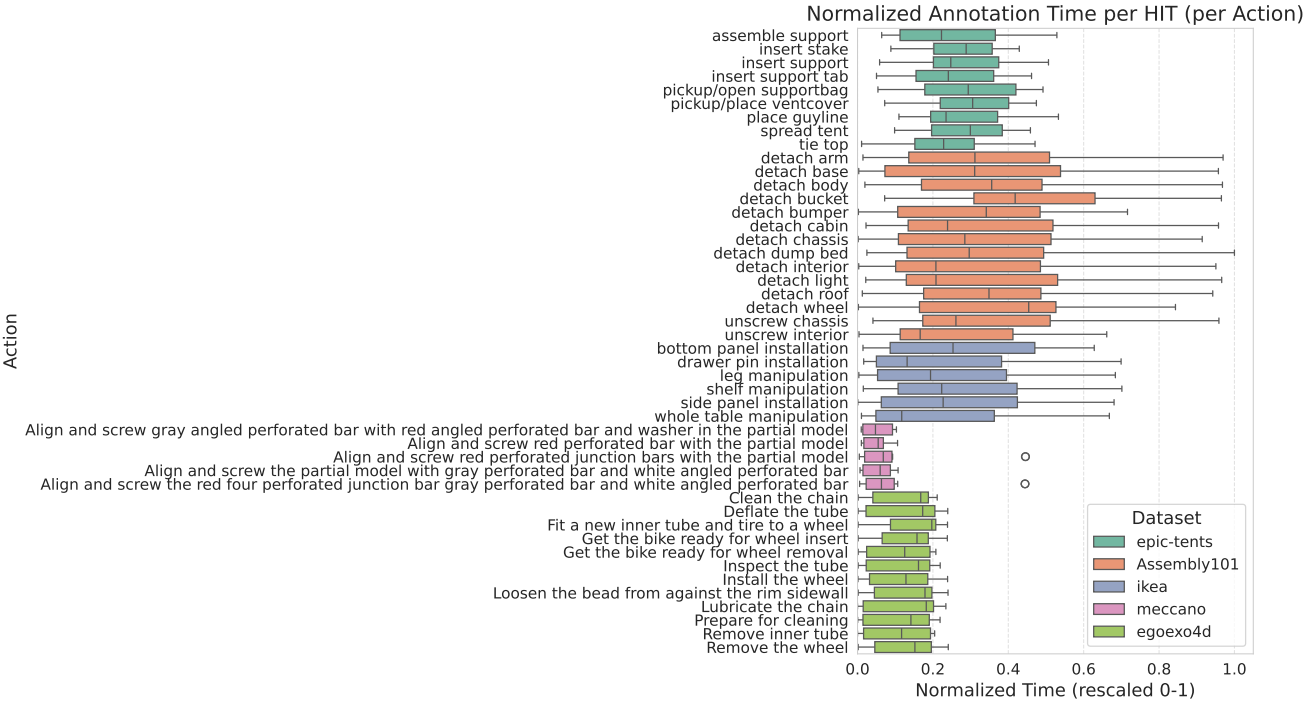


Figure 3. Distribution of normalized annotation time per HIT (rescaled to [0, 1]) across actions and datasets.

versely, actions in *Assembly101* show higher values, such as *detach bucket* (0.452 ± 0.244), reflecting their multistep nature and complex object manipulation.

Annotations for *Meccano* are particularly efficient, with most actions taking less than 0.13 normalized time units, suggesting clear, concise clips and high inter-rater consistency. For *EPIC-Tents*, most actions range between 0.24 and 0.30, indicating moderate complexity and uniform clip duration. *IKEA* annotations fall in a similar range but with slightly higher variance, particularly for actions involving furniture manipulation such as *bottom panel installation* and *shelf manipulation*.

These findings confirm that task complexity, action type, and visual clarity play a key role in determining annotation time. Importantly, the normalized time metric can be used to estimate annotation cost and design future annotation tasks with balanced load per HIT.

To further interpret these findings, we administered a short feedback questionnaire at the end of the pilot. Two main observations emerged: First, the notions of skill, technique, and confidence were perceived as highly correlated and often difficult to distinguish. Second, annotators reported that they rarely selected a single factor, instead opting for multiple checkboxes to reflect an intertwined rationale.

Based on these insights, we chose to omit the justification checkboxes in the final annotation protocol used in PROSKILL. While well-intentioned, these labels introduced ambiguity and cognitive overhead, and their high co-occurrence suggested they did not yield clearly disentangled supervisory signals.

2. Swiss Tournament Round Generation

To efficiently collect pairwise comparisons without exhaustively evaluating all video pairs, we adopt a Swiss-style tournament protocol that operates in rounds. After each round r, a global ranking is updated and used to generate the set of comparisons for the next round $r{+}1$.
Let $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ be the set of video segments to be ranked within a sub-task. At round r, we maintain:

- A set of match outcomes $\mathcal{M}^{(r)} = \{(s_i, s_j, y_{ij})\}$, where $y_{ij} = 1$ if s_i is preferred over s_j, 0 otherwise, in draw, i.e. no agreements on AMT labeling, 0.5.
- A ranking function $\pi^{(r)} : \mathcal{S} \to \mathbb{R}$ that assigns a score to each segment based on outcomes so far.

We estimate $\pi^{(r)}$ using a combination of ELO rating and a Swiss tournament point system. The ELO score for segment s_i after round r is recursively updated using:

$$\text{Elo}_i^{(r+1)} = \text{Elo}_i^{(r)} + K \cdot (y_{ij} - p_{ij}),$$

where

$$p_{ij} = \frac{1}{1 + 10^{(\text{Elo}_j^{(r)} - \text{Elo}_i^{(r)})/400}}$$

is the expected probability of s_i winning against s_j, and K is a constant determining the learning rate.

Each new round selects a set of candidate pairs $\mathcal{C}^{(r+1)} \subset \mathcal{S} \times \mathcal{S}$ by:

$$\mathcal{C}^{(r+1)} = \big\{(s_i, s_j) \;\big|\; |\pi^{(r)}(s_i) - \pi^{(r)}(s_j)| \text{ is minimal}, \\ (s_i, s_j) \notin \mathcal{M}^{(\leq r)}\big\} \tag{1}$$

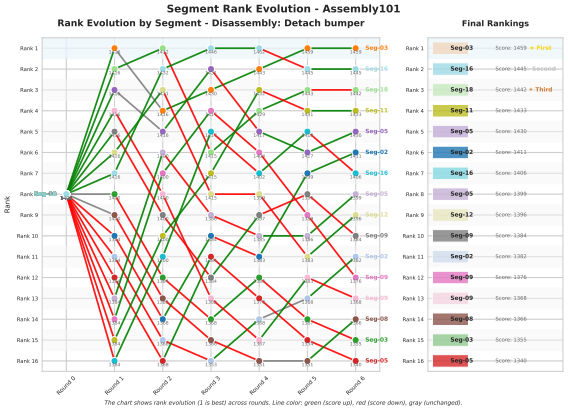
i.e., we pair segments with similar scores that have not been compared yet.

3. Hyperparameter optimization

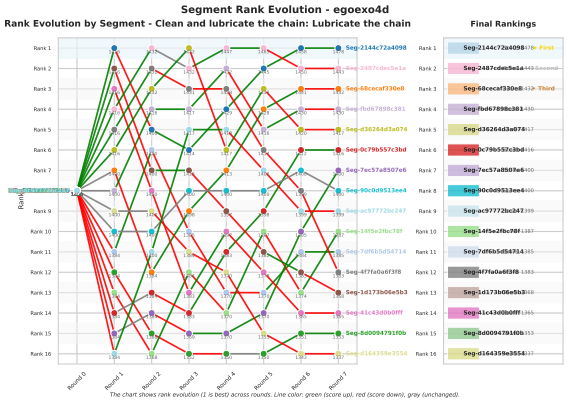
Table 1 summarizes the impact of three key hyperparameters, learning rate (θ), augmentation noise standard deviation (σ), and score normalization factor (τ),on the USDL baseline's performance across multiple datasets and feature types (I3D and VideoMAE). Each row aggregates results by varying a single hyperparameter while averaging over the others, reporting both mean and best Spearman's correlation coefficients on validation and test splits. The results reveal several insights: First, the optimal hyperparameters differ across datasets, reflecting their distinct data distributions and complexity. For example, Meccano and IKEA tend to favor moderate learning rates around 0.004–0.01 and normalization factors near 20–25, whereas datasets like EpicTents show greater sensitivity to augmentation noise, which can boost generalization on test data. Second, VideoMAE features consistently yield higher Spearman correlations than I3D, especially on more challenging datasets like EgoExo4D and EpicTents, suggesting stronger representation power for skill-related information. Third, augmentation noise and normalization play a crucial role in improving robustness and ranking consistency, sometimes more so than fine-tuning the learning rate.

4. Qualitative Examples

To better understand how segment rankings evolve across Swiss tournament rounds, we provide qualitative examples from two datasets in Figure 4. Each plot illustrates the rank trajectory of all segments involved in a specific sub-task. In the *Assembly101* example, we observe clear separation among high-performing segments early in the process, with minimal rank fluctuation in the final rounds. In contrast, *EgoExo4D* exhibits a high degree of stability for both top- and bottom-ranked segments from as early as round 3, confirming the Swiss format's effectiveness in identifying the extremes of the ranking spectrum with limited supervision. These examples visually reinforce the earlier quantitative findings: although only a subset of all possible comparisons is used, the Swiss tournament efficiently converges to rankings that closely approximate the full round-robin outcome.



(a) *Assembly101* – Detach Bumper



(b) *EgoExo4D* – Lubricate the Chain

Figure 4. Ranking evolution of individual segments over Swiss tournament rounds. Each line corresponds to a segment, with the y-axis representing its rank (1 is best). Color indicates whether the segment's final score increased (green), decreased (red), or remained unchanged (gray) compared to earlier rounds.

Dataset	Feature	Aggregated by	θ	σ	τ	ρ_{val}^{mean}	ρ_{val}^{best}	ρ_{test}^{mean}	ρ_{test}^{best}
Assembly101	I3D	lr	0.003	0.047	20.5	**0.39**	0.48	0.14	0.20
Assembly101	I3D	noise	0.004	0.000	20.5	0.35	0.47	0.12	0.24
Assembly101	I3D	norm	0.005	0.045	25.0	0.34	**0.51**	0.11	0.26
Assembly101	VideoMAE	lr	0.010	0.047	20.5	0.21	0.39	**0.15**	**0.27**
Assembly101	VideoMAE	noise	0.004	0.100	20.5	0.22	0.37	0.13	0.22
Assembly101	VideoMAE	norm	0.007	0.044	16.0	0.18	0.38	0.12	0.24
Meccano	I3D	lr	0.010	0.056	23.5	0.57	0.77	**0.35**	0.64
Meccano	I3D	noise	0.004	0.004	20.9	0.60	0.77	**0.35**	0.64
Meccano	I3D	norm	0.004	0.047	25.0	0.60	0.80	0.32	0.64
Meccano	VideoMAE	lr	0.001	0.047	20.5	**0.61**	**0.84**	0.30	0.56
Meccano	VideoMAE	noise	0.004	0.005	20.5	0.48	0.83	0.28	0.62
Meccano	VideoMAE	norm	0.004	0.047	25.0	0.49	**0.84**	0.24	**0.65**
EpicTents	I3D	lr	0.001	0.035	19.4	0.05	0.45	0.13	0.21
EpicTents	I3D	noise	0.004	0.005	20.5	0.19	0.58	0.10	0.22
EpicTents	I3D	norm	0.004	0.012	25.0	0.20	0.65	0.04	0.21
EpicTents	VideoMAE	lr	0.008	0.032	19.4	**0.51**	0.64	0.06	0.24
EpicTents	VideoMAE	noise	0.004	0.200	16.0	**0.51**	0.65	**0.14**	**0.34**
EpicTents	VideoMAE	norm	0.004	0.047	16.0	0.49	**0.68**	0.12	**0.34**
IKEA	I3D	lr	0.010	0.047	20.5	0.62	0.77	0.17	0.27
IKEA	I3D	noise	0.004	0.001	20.5	0.62	0.71	0.17	0.24
IKEA	I3D	norm	0.004	0.046	16.0	0.59	0.73	0.15	0.27
IKEA	VideoMAE	lr	0.005	0.047	20.5	0.58	0.75	**0.28**	0.34
IKEA	VideoMAE	noise	0.004	0.000	20.2	**0.65**	**0.79**	0.26	0.34
IKEA	VideoMAE	norm	0.004	0.046	16.0	0.63	0.78	0.24	**0.36**
EgoExo4D	I3D	lr	0.0100	0.047	20.5	0.36	0.55	0.28	0.43
EgoExo4D	I3D	noise	0.0044	0.100	20.5	0.41	0.56	0.29	0.55
EgoExo4D	I3D	norm	0.0044	0.047	16.0	0.40	0.56	0.25	0.45
EgoExo4D	VideoMAE	lr	0.0005	0.047	20.5	0.44	0.58	**0.43**	0.50
EgoExo4D	VideoMAE	noise	0.0044	0.075	20.5	0.42	**0.62**	0.38	0.46
EgoExo4D	VideoMAE	norm	0.0044	0.047	25.0	**0.44**	0.64	0.33	**0.53**

Table 1. Grid search results for the USDL baseline.
θ: learning rate; σ: augmentation noise std; τ: normalization parameter; ρ: Spearman's correlation.
Aggregation (Aggregated by): learning rate (lr), augmentation noise (noise), normalization (norm).
Bold highlights the best value per aggregation group within each dataset.