Supplementary Material for "HARP: Hierarchical Adaptive Ranking with Probabilistic Modeling for Skill Determination"

Hui Yu[1,2], Xiao Ke[1,2]*, Zhihong Zeng[3,4]*, Huangbiao Xu[1,2]*, Huanqi Wu[1,2]
[1]Fujian Provincial Key Laboratory of Networking Computing and Intelligent Information Processing, College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China
[2]Engineering Research Center of Big Data Intelligence, Ministry of Education, Fuzhou 350108, China
[3]College of Mathematics and Information Engineering, Longyan University, Longyan 364012, China
[4]Key Laboratory of Big Data Mining and Application (Longyan University), Fujian Province University, Longyan 364012, China
kex@fzu.edu.cn, 82009054@lyun.edu.cn, {hui.yu.xiamen.work, huangbiaoxu.chn}@gmail.com

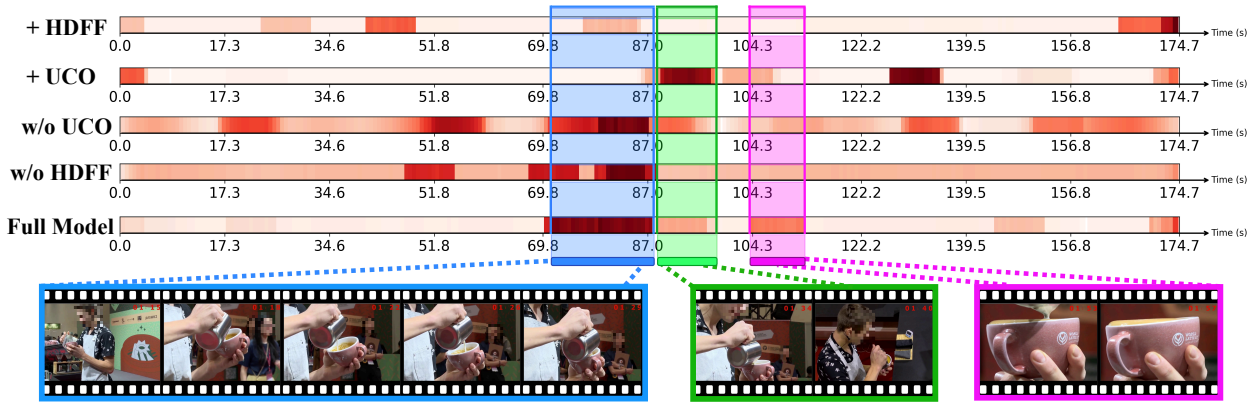


Figure 1. Visualization of attention weights under different ablation settings to analyze the role of each module. (a) Model with only the HDFF module. (b) Model with HDFF and UCO modules. (c) Model without UCO. (d) Model without HDFF. (e) Full model with all components.

1. Method Analysis and Derivations

1.1. Analysis of Module Contributions

Module-wise Ablation Analysis. To thoroughly evaluate the individual contributions of the HDFF, UCO, MVP, and DyM modules in the ranking-based skill determination task, we conducted ablation studies on the Latte Art task within the CoffeeCraft dataset. As shown in Figure 1, combined with the visualization of attention weights, we systematically analyzed the modules' effects on attention focus over key action segments and the accuracy of action boundary delineation, aiming to elucidate their functional distinctions and collaborative mechanisms. The results are detailed as follows.

In the configuration using only the HDFF module (a), the attention weight visualization shows that the model partially focuses on key action regions, reflecting HDFF's strength in extracting fine-grained semantic features. However, the attention boundaries remain relatively vague, failing to precisely separate the start and end points of actions, which indicates that HDFF alone lacks sufficient constraints for boundary localization.

In contrast, the configuration using both HDFF and UCO modules (b) demonstrates a more focused and coherent attention distribution. The addition of UCO helps the model better capture temporal dependencies and key

*Corresponding authors.

features across action sequences, improving fine-grained recognition and partially sharpening the attention boundaries. Nevertheless, some boundaries remain indistinct, suggesting that further mechanisms may be needed for precise action segmentation.

Configuration (c), which integrates HDFF, MVP, and DyM modules while excluding UCO, exhibits a more focused attention distribution. Attention is largely concentrated on key action regions, and action boundaries are better defined. This suggests a strong synergy among HDFF's feature extraction, MVP's probabilistic ranking, and DyM's dynamic constraints, collectively enhancing key action recognition and boundary localization. Nevertheless, some attention still disperses to non-key regions, implying that the contrastive optimization introduced by MVP and DyM may help suppress irrelevant attention.

In configuration (d), which combines UCO, MVP, and DyM while excluding HDFF, attention is primarily concentrated on the first key action region with relatively clear boundaries. However, the boundaries of other key actions remain poorly defined, and the overall attention distribution is more dispersed. This configuration outperforms both HDFF-only (a) and HDFF+UCO (b), indicating that UCO, MVP, and DyM together enhance attention focus and boundary recognition. Nevertheless, it still underperforms compared to configurations that include HDFF, highlighting HDFF's pivotal role in improving feature representation and attention precision.

The full model (configuration e), integrating HDFF, UCO, MVP, and DyM, achieves the best attention distribution performance. Visualization indicates that over 92% of attention is concentrated on the three key action regions, with clearer boundary delineations. Specifically, the attention maps reveal that regions of high attention sharply correspond to the precise onset of key actions, while the gradual fading of attention aligns closely with the action termination points. Remarkably, despite the complete absence of any explicit boundary annotations during training, the model implicitly learns to localize and attend to the temporal boundaries of critical actions through self-supervised attention mechanisms. This emergent capability indicates that the integrated modules collaboratively enable the model to capture fine-grained temporal structures and effectively segment key actions without direct supervision on boundaries. This superior performance stems from the complementary collaboration of modules: HDFF provides fine-grained semantic features; UCO enhances training stability through contrastive optimization; MVP improves ranking accuracy; and DyM strengthens dynamic adaptability and boundary recognition. The full model surpasses other configurations in attention concentration and boundary clarity, confirming the synergistic gains among modules.

In summary, attention weight visualizations and ablation studies comprehensively reveal the distinct functionalities and contributions of HDFF, UCO, MVP, and DyM in ranking-based skill determination tasks. HDFF acts as the core driver enhancing key action capture and boundary clarity; UCO ensures stable attention allocation; MVP and DyM respectively enhance ranking accuracy and model adaptability through probabilistic ranking and dynamic constraints.

Table 1. Ablation study of the variance gating mechanism on the BEST and CoffeeCraft datasets (accuracy %).

Variance Gating	BEST	CoffeeCraft
×	86.51	73.13
✓	**87.53**	**74.84**

Variance Gating Ablation. Table 1 presents the impact of the variance gating mechanism on model performance. When this mechanism is enabled, accuracy increases from 86.51% to 87.53% on the BEST dataset and from 73.13% to 74.84% on the CoffeeCraft dataset. These improvements suggest that variance gating contributes to enhanced representational capacity and greater robustness of the model.

The variance gating mechanism computes the variance of normalized feature channels and applies a sigmoid function to obtain dynamic gating values, which weight the residual branch features adaptively. This approach suppresses potential irrelevant noise while emphasizing channels containing rich semantic information, facilitating accurate extraction of sub-action semantics.

Typically, the magnitude of variance in a feature channel correlates with the richness of the information it contains. Components exhibiting smaller variance generally have lower energy and may include irrelevant noise or redundant information; removing these components usually results in negligible information loss [1, 2]. This observation provides theoretical justification for variance gating: emphasizing channels with larger variance enables the model to dynamically retain more informative signals, reduce noise interference, and enhance the discriminative power and stability of feature representations.

Given the complex and diverse semantic patterns and noise present in action sequences, variance gating effectively mitigates the negative effects of noise on semantic decomposition by dynamically suppressing low-variance channels. This mechanism enables the model to more precisely capture key features related to sub-actions, thereby enhancing overall skill determination performance.

1.2. Subspace Decomposition via SGDW

Initial Subspace Count Captures Action Complexity. The initial layer is set to $K = 8$ to capture the semantic complexity of action sequences, which typically involve

multiple sub-actions such as grasping, twisting, or knotting, providing sufficient subspaces to represent diverse semantic expressions while balancing granularity and computational cost. A smaller K (e.g., $K = 4$) limits decomposition resolution and risks information loss, whereas a larger K (e.g., $K = 16$) may introduce redundancy and higher computational overhead. As shown in the ablation study in Fig. **??**, we evaluated propagation layers from Layer 2 to Layer 6, with corresponding subspace numbers Layer 2 $[2, 1]$, Layer 3 $[4, 2, 1]$, Layer 4 $[8, 4, 2, 1]$, Layer 5 $[16, 8, 4, 2, 1]$, and Layer 6 $[32, 16, 8, 4, 2, 1]$. The results indicate that an initial layer with $K = 8$ achieves the best trade-off between representation capacity and computational efficiency, consistently outperforming both smaller and larger initial layer settings.

Base-2 Hierarchical Subspace Configuration. The HDFF module employs a layered configuration $K = 8, 4, 2, 1$, reducing subspaces geometrically to transition from coarse-grained to fine-grained modeling. The initial layer ($K = 8$) captures a variety of distinct skill semantics, integrating information across all temporal steps T. Subsequent layers ($K = 4, 2, 1$) aggregate semantics, focusing on key sub-action interactions, e.g., grasping and knotting synergies. The base-2 geometric progression ensures smooth dimensional reduction, balancing representation capacity and efficiency, similar to hierarchical feature aggregation in deep learning. Halving subspace counts per layer facilitates a natural progression toward higher-order semantic interactions. In contrast, a base-4 progression, e.g., $K = 16, 4, 1$, results in an overly large initial layer ($K = 16$), increasing redundancy, and reaches holistic modeling ($K = 1$) too abruptly. Base-3 or base-5 progressions, e.g., $K = 9, 3, 1$ or $K = 25, 5, 1$, are impractical due to dimensional misalignment with feature structures, hindering smooth decomposition and geometric consistency. The base-2 progression aligns with the hierarchical nature of action sequences, optimizing skill determination modeling, as validated by ablation studies in the main text.

2. Subspace-Based Action Representation

Modeling complex action sequences, such as coffee-making, competitive sports [3–5, 7, 8], and human dances [6], requires capturing the semantics of multiple sub-actions, which may be partially overlapping and nonlinearly related. Directly modeling each sub-action is challenging because their number varies across tasks and their semantic boundaries are often ambiguous. To address this, HDFF decomposes the input feature into a limited number of K subspaces, where x_k denotes the feature in the k-th subspace, Z_k is the set of latent skill-relevant components captured by this subspace, $w_{z,k}$ encodes the contribution of component z within subspace k, and n_k represents subspace-specific noise. Each subspace encodes a subset of shared semantic components and assigns different weights to emphasize the relative contribution of each component, while n_k isolates task-irrelevant variations. This design removes the need to know the exact number of sub-actions, maintains consistency across subspaces through shared components, and allows nonlinear combinations of sub-action contributions, effectively representing overlapping or partially redundant actions. For example, in coffee-making, sub-actions such as grasping a cup, pouring milk, and stirring may share the semantic component rotation, which can be aggregated within subspaces for consistent representation. In the initial layer with $K = 8$ subspaces, each subspace captures fine-grained manipulations, such as tilting, rotation, or lifting, ensuring that even when the number of sub-actions exceeds K, the model can still express and distinguish all relevant sub-action semantics.

The hierarchical configuration of HDFF ($K = 8, 4, 2, 1$) enhances semantic differentiation by progressively reducing the number of subspaces, aggregating lower-level semantic components into higher-level representations. The initial layer ($K = 8$) performs fine-grained decomposition, capturing diverse skill-relevant patterns. The second and third layers ($K = 4, 2$) integrate and refine these components, forming more complex representations, such as coordinated movements or combined operations. The final layer ($K = 1$) consolidates all semantic components to provide a holistic evaluation of overall skill. This hierarchical aggregation is inspired by human perceptual mechanisms—gradually abstracting global semantics through multi-level local observations—and allows a limited number of subspaces K to represent a potentially larger variety of skill-relevant patterns across tasks. Visualizations show that the $K = 8, 4, 2, 1$ configuration captures complementary semantic components across subspaces, indicating that the hierarchical subspace approach effectively encodes diverse skill-relevant patterns. The low off-diagonal similarity in the matrices, together with robust performance, confirms that the HDFF subspace design provides a flexible and reliable framework for skill determination.

Table 2. Ablation study on the BEST dataset (accuracy %). AE, BH, OR, SE, and TT denote the tasks Apply Eyeliner, Braid Hair, Origami, Scrambled Eggs, and Tie Tie, respectively. Bold font represents the best performance.

Activation	AE	BH	OR	SE	TT	Avg. Acc
Sigmoid	82.55	80.34	80.19	83.12	**90.13**	83.27
Softplus	81.70	**82.48**	79.72	90.26	**90.13**	84.86
Tanh	**85.11**	**82.48**	**84.91**	**95.45**	89.70	**87.53**

2.1. Ablation Study of Activation Functions in DyM

The Dynamic Margin Ranking Loss (DyM) addresses the challenge of adapting margin adjustments to diverse skill

differences in video-based skill determination, with its performance critically dependent on the choice of activation function in dynamic margin computation. To this end, we conducted an ablation study on the BEST dataset, comparing the effects of Sigmoid, Softplus, and Tanh activation functions on the DyM loss, with results presented in Table 2. The core of the DyM loss lies in the dynamic margin, defined as:

$$m = \alpha f(\beta(y_i - y_j)) + m_0, \tag{1}$$

where y_i and y_j are scalar predictions for a video pair, α and β are learnable parameters controlling the margin scaling and sensitivity, respectively. m_0 is a fixed base margin, and f denotes the activation function. To ensure α and β remain strictly positive, they are reparameterized as $\alpha = \exp(\tilde{\alpha})$ and $\beta = \exp(\tilde{\beta})$, where $\tilde{\alpha}$ and $\tilde{\beta}$ are unconstrained scalar parameters optimized via backpropagation. The loss function is defined as:

$$\mathcal{L}_{\text{DyM}}(y_i, y_j, m_0) = \max(0, m - (y_i - y_j)). \tag{2}$$

The non-linear mapping properties of the activation function directly influence the adaptability of the margin, thereby determining the capability of the model to distinguish skill differences across varied scenarios.

The Tanh activation function, with its symmetric output range of $(-1, 1)$, demonstrates superior adaptability in dynamic margin computation. Its bidirectional non-linear mapping effectively handles prediction differences $y_i - y_j$, enabling flexible margin adjustments across diverse tasks, such as Scrambled Eggs and Origami in the BEST dataset, where it adeptly captures subtle variations in skill levels. In contrast, the Sigmoid activation, with an output range of $(0, 1)$, tends to compress prediction differences, resulting in overly conservative margin adjustments that limit its flexibility in tasks requiring fine-grained differentiation. Similarly, the Softplus activation, which produces non-negative smooth outputs, offers stability in high-variance scenarios but is constrained by its unidirectional non-linearity, reducing its generalization across tasks with small prediction differences.

The experimental results highlight the robustness of Tanh across the diverse tasks included in the BEST dataset, consistently achieving superior performance over Sigmoid and Softplus, particularly under conditions involving high action complexity. Although Sigmoid and Softplus demonstrate stability in certain tasks, limitations in their non-linear mapping reduce their discriminative capacity relative to Tanh. This ablation study confirms that Tanh serves as the most effective activation function for the DyM loss, offering robust and adaptable support for video-based skill determination.

2.2. Detailed Derivation of the MVP Loss

Building upon the main text, where the Mean-Variance Probability Loss (MVP) was introduced to model uncertainty inherent in subjective skill annotations, we provide here a detailed derivation of the loss function to rigorously justify its formulation.

Our model represents the predicted scores for videos i and j as independent Gaussian random variables:

$$\hat{y}_i \sim \mathcal{N}(\mu_i, \sigma_i^2), \quad \hat{y}_j \sim \mathcal{N}(\mu_j, \sigma_j^2), \tag{3}$$

where μ_i and μ_j denote the expected scores, and σ_i^2 and σ_j^2 represent the associated predictive uncertainties. This probabilistic framework effectively accommodates the subjective noise and annotation variability inherent in skill determination tasks.

The primary focus is the probability that the model correctly ranks video i above video j, expressed as:

$$P(\hat{y}_i > \hat{y}_j) = P(\Delta\hat{y} > 0), \quad \text{where} \quad \Delta\hat{y} = \hat{y}_i - \hat{y}_j. \tag{4}$$

Given that the difference of two independent Gaussian variables is also Gaussian, $\Delta\hat{y}$ follows the distribution:

$$\Delta\hat{y} \sim \mathcal{N}(\mu_i - \mu_j, \sigma_i^2 + \sigma_j^2). \tag{5}$$

To evaluate $P(\Delta\hat{y} > 0)$, we standardize $\Delta\hat{y}$ into a standard normal variable Z:

$$Z = \frac{\Delta\hat{y} - (\mu_i - \mu_j)}{\sqrt{\sigma_i^2 + \sigma_j^2}} \sim \mathcal{N}(0, 1). \tag{6}$$

This standardization enables the probability to be reformulated as:

$$\begin{aligned} P(\Delta\hat{y} > 0) &= P\left(Z > \frac{0 - (\mu_i - \mu_j)}{\sqrt{\sigma_i^2 + \sigma_j^2}}\right) \\ P(\Delta\hat{y} > 0) &= P\left(Z > -\frac{\mu_i - \mu_j}{\sqrt{\sigma_i^2 + \sigma_j^2}}\right). \end{aligned} \tag{7}$$

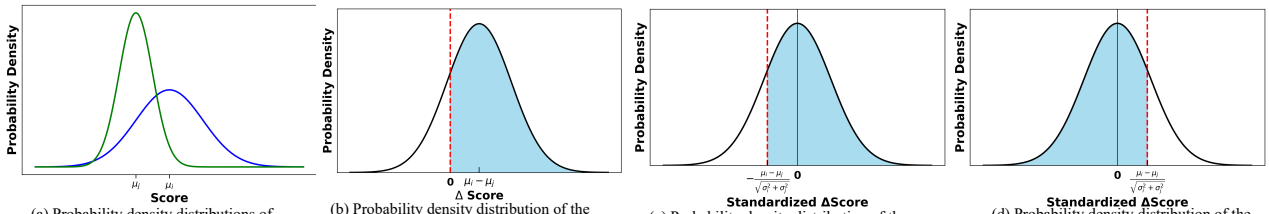
Utilizing the symmetry property of the standard normal distribution, where $P(Z > -x) = P(Z < x)$, we derive the final closed-form expression:

$$P(\hat{y}_i > \hat{y}_j) = \Phi\left(\frac{\mu_i - \mu_j}{\sqrt{\sigma_i^2 + \sigma_j^2}}\right), \tag{8}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

This derivation is graphically summarized in Figure 2, illustrating the stepwise transformation from individual Gaussian score distributions to the standardized variable used for probability evaluation.

$$\mathcal{L}_{\text{prob}} = -\log P(\hat{y}_i > \hat{y}_j), \quad \mathcal{L}_{\text{var}} = \log(\eta + \sigma_i^2 + \sigma_j^2), \tag{9}$$



(a) Probability density distributions of predicted scores $\hat{y}_i$ and $\hat{y}_j$ for videos i and j, modeled as normal distributions $\mathcal{N}(\mu_i, \sigma_i^2)$ and $\mathcal{N}(\mu_j, \sigma_j^2)$, illustrating the individual score predictions for the two videos.

(b) Probability density distribution of the score difference $\Delta\hat{y} = \hat{y}_i - \hat{y}_j$ for videos i and j, following a normal distribution $\mathcal{N}(\mu_i - \mu_j, \sigma_i^2 + \sigma_j^2)$, reflecting the distribution characteristics of the difference between the predicted scores.

(c) Probability density distribution of the standardized score difference $\Delta\hat{y}$ for videos i and j, standardized as $\frac{\Delta\hat{y}-(\mu_i-\mu_j)}{\sqrt{\sigma_i^2+\sigma_j^2}}$ with mean $\mu_i - \mu_j$, showing the standardization process based on the original difference.

(d) Probability density distribution of the further standardized score difference for videos i and j, transformed into a standard normal distribution $Z = \frac{\Delta\hat{y}-(\mu_i-\mu_j)}{\sqrt{\sigma_i^2+\sigma_j^2}}$, with the red line indicating $Z = 0$, representing the basis for calculating the probability of correct ranking $\Delta\hat{y} > 0$.

Figure 2. Derivation process of the probability of correct model prediction, illustrating the step-by-step transformation from individual score distributions of videos i and j (modeled as $\mathcal{N}(\mu_i, \sigma_i^2)$ and $\mathcal{N}(\mu_j, \sigma_j^2)$) to the standardized score difference $\Delta\hat{y} = \hat{y}_i - \hat{y}_j$, and finally to the standard normal distribution $Z = \frac{\Delta\hat{y}-(\mu_i-\mu_j)}{\sqrt{\sigma_i^2+\sigma_j^2}}$ for evaluating $P(\Delta\hat{y} > 0)$.

Algorithm 1 Uniform Contrastive Optimization (UCO)

Input: Input feature sequence x, configuration C (training flag, model dim d_m, length T)
Output: Unified feature $f_{\mathcal{U}}$, auxiliary contrast score s (empty if inference)

Initialize convolutional projection
$x_c \leftarrow \text{Conv1D}(x) \rightarrow$ dimension d_m
$x_a \leftarrow \text{SGDW}(x_c, \text{subspace count } K{=}1)$ ▷ single-subspace dynamic weighting
$f_{\mathcal{U}} \leftarrow \text{TemporalPool}(\text{Project}(x_a))$
if training mode **then**
    $s \leftarrow \text{MLP}_s(f_{\mathcal{U}})$
else
    $s \leftarrow \emptyset$
end if
return $f_{\mathcal{U}}$, s

$$\mathcal{L}_{\text{MVP}} = \lambda \cdot \frac{\mathcal{L}_{\text{prob}} + \mathcal{L}_{\text{DyM}}(\mu_i, \mu_j, m_1)}{\sigma_i^2 + \sigma_j^2} + \kappa \cdot \mathcal{L}_{\text{var}} \tag{10}$$

In the loss function (Eq. (10)), the hyperparameters λ, κ, and η are initialized with predefined values and treated as learnable scalars during training. To ensure λ and κ remain strictly positive, they are reparameterized as $\lambda = \exp(\tilde{\lambda})$ and $\kappa = \exp(\tilde{\kappa})$, where $\tilde{\lambda}$ and $\tilde{\kappa}$ are unconstrained scalar parameters optimized via backpropagation. Similarly, to constrain η within the range $(0, 1)$, it is reparameterized as $\eta = \sigma(\tilde{\eta})$, where $\tilde{\eta}$ is a learnable scalar and σ denotes the sigmoid function. To mitigate numerical instability when the predicted variances σ_i^2 and σ_j^2 are small, a lower-bound constant η is introduced in the variance regularization term: $\mathcal{L}_{\text{var}} = \log(\eta + \sigma_i^2 + \sigma_j^2)$. This term, combined with the positivity constraints on λ, κ, and η, enabling robust modeling of skill-relevant semantics.

Algorithm 2 HDFF: Hierarchical Dynamic Feature Fusion

Input: Temporally enhanced feature x_t, unified feature $f_{\mathcal{U}}$ (optional), configuration C (node array $N = \{n_1, \ldots\}$, model dim d_m)
Output: Fused feature y, SGDW weight list $W = \{w_1, \ldots\}$

Project and normalize x_t to x_i
if $f_{\mathcal{U}}$ is not null **then**
    Fuse x_i and $f_{\mathcal{U}}$ via projection $\rightarrow$ updated x_i
end if
Initialize $x_c \leftarrow x_i$, $W \leftarrow \emptyset$
for all $n \in N$ **do**
    Partition x_c into n sub-features x_s
    Encode and normalize $x_s \rightarrow x_e$
    Apply depth-wise convolution on $x_e \rightarrow$ contextualized x_{ctx}
    for all $x_{\text{sub}} \in x_s$ **do**
        Compute raw SGDW logit on $x_{\text{sub}} + x_{\text{ctx}}$
        Apply temperature-scaled softmax $\rightarrow$ SGDW weight w
    end for
    $x_w \leftarrow x_{\text{ctx}} \odot w$ ▷ SGDW-weighted aggregation
    $x_r \leftarrow \sigma(\text{Var}(x_e)) \odot x_e$ ▷ variance gating residual
    $x_t \leftarrow \text{FFN}(\text{LayerNorm}(x_r + x_w))$
    $x_c \leftarrow x_t + \sigma(\text{Var}(x_c)) \odot x_c$ ▷ hierarchical residual
    Append normalized w to W
end for
$x_m \leftarrow \text{ResidualBlock}(\text{LayerNorm}(x_c))$
$x_f \leftarrow x_m + \sigma(\text{Var}(x_t)) \odot x_t$
$y \leftarrow \text{ProjectAndNorm}(x_f)$
return y, W

The overall loss combines: (1) a variance-aware ranking

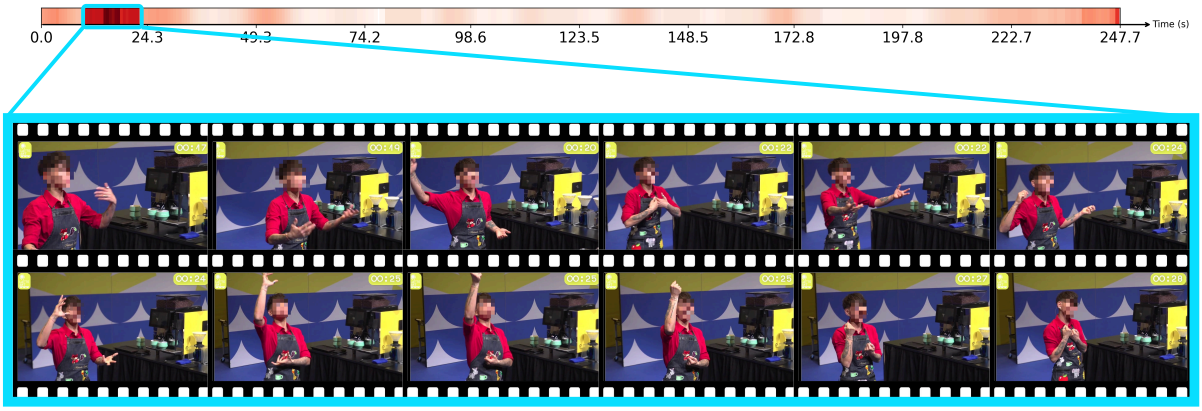


Figure 3. Case featuring the competitor introducing the performance with pronounced hand movements, which attracts concentrated attention.

loss, where the probability loss $\mathcal{L}_{\text{prob}}$ and dynamic margin loss $\mathcal{L}_{\text{DyM}}$ are weighted by $(\sigma_i^2+\sigma_j^2)^{-1}$, encouraging confident predictions; and (2) a smooth regularization penalty $\mathcal{L}_{\text{var}}$ to avoid degenerate solutions with overly large predicted uncertainties. This formulation enables the model to automatically trade off between ranking fidelity and uncertainty control during optimization.

3. Experiments

3.1. Implementation Details

We propose the **Uniform Contrastive Optimization (UCO)** scheme, as detailed in Algorithm 1, to achieve feature consistency optimization. Given the input feature vector and configuration parameters, the algorithm extracts a feature representation of dimension d_m via convolutional transformation, followed by the SGDW module to effectively isolate and aggregate key skill-relevant semantics. A unified feature vector $f_{\mathcal{U}} \in \mathbb{R}^{d_m}$ is then obtained through pooling and projection. During training, an auxiliary contrast score s is computed by a neural network based on the unified feature; during inference, s is set to `None`, thus distinguishing the training and testing phases.

Building upon the unified feature from UCO, we introduce the **Hierarchical Dynamic Feature Fusion (HDFF)** mechanism, presented in Algorithm 2. HDFF takes as input the temporally enhanced feature x_t and the unified feature $f_{\mathcal{U}}$. It first projects and normalizes x_t, and if $f_{\mathcal{U}}$ is available, fuses and updates x_t accordingly. The algorithm then iterates over a predefined node array N, segmenting the input feature into multiple subspaces. After applying positional encoding, the number of subspaces is set to match the subspace count used in the SGDW module. Subsequently, sub-action semantics within each subspace are isolated and their relative importance is dynamically evaluated. Convolutional networks and attention mechanisms are employed to weight and fuse local features adaptively. Residual variance normalization and feedforward layers further refine feature representations, maintaining orthogonal weight constraints. Finally, the fused features are processed through residual neural networks and normalization layers to produce the output feature y along with the corresponding set of fusion weights W.

3.2. Experiment Settings

All experimental evaluations were executed on a Tesla K80 GPU, selected to maintain uniform computational performance across the study. To ensure reproducibility, a consistent random seed of 42 was enforced throughout all experimental iterations, aligning with the methodology detailed in the main text. The training regimen comprised 4000 epochs with a batch size of 16, facilitating robust model convergence and comprehensive parameter optimization. To avoid manual bias in hyperparameter selection and to ensure fair model comparison, we employed the Optuna framework for automated hyperparameter tuning. All hyperparameter configurations were selected based on performance on the validation set, enabling systematic exploration of the search space while ensuring the generalizability of resulting models.

To mitigate the impact of inherent variability in action execution, a supplementary data augmentation strategy was integrated into the training pipeline, incorporating Gaussian noise with a mean of 0 and a standard deviation of 0.01 applied to the video features. This approach emulates realistic perturbations, including subtle motion dynamics and environmental fluctuations characteristic of competitive environments, thereby fostering the development of resilient

action pattern representations. Such enhancements, as elaborated in the primary analysis, contribute to the stability and generalizability of the model across diverse operational scenarios.

Since most methods evaluated on EPIC-Skills and BEST are not publicly available, we report HARP's parameter count and computational cost (11.9 million parameters and 0.085 GFLOPs) for reference, with FLOPs estimated using the PyTorch profiler on input features of size $(400, 1024)$. This relatively low computational cost indicates that HARP can be executed with modest processing resources, making it feasible for deployment in a range of practical scenarios.

3.3. Expressiveness and Attention

Attention of the model predominantly focuses on critical segments within the coffee-making process, demonstrating effective capture of core technical actions. However, in certain specific cases, as shown in Fig. 3, competitors may exhibit phase transitions during coffee preparation, such as abruptly shifting to an introductory speech accompanied by dynamic and expressive gestures. These visually salient and context-rich segments naturally attract attention, thereby diverting focus toward these prominent features and potentially impacting continuous tracking of subsequent preparation actions. It is noteworthy that such occurrences constitute a relatively small proportion of the overall dataset, as not all competitors display phase transitions or exaggerated expressive behavior. In light of this, future work will emphasize precise annotation of key actions to enhance capability in recognizing and discriminating core coffee-making steps, ultimately improving accuracy and comprehensiveness of skill determination and further optimizing overall performance.

References

[1] Junjie He, Bohua Chen, Yinzhang Ding, and Dongxiao Li. Feature variance ratio-guided channel pruning for deep convolutional network acceleration. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2

[2] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[3] Xiao Ke, Huangbiao Xu, Xiaofeng Lin, and Wenzhong Guo. Two-path target-aware contrastive regression for action quality assessment. *Inf. Sci.*, 664:120347, 2024. 3

[4] Huangbiao Xu, Xiao Ke, Yuezhou Li, Rui Xu, Huanqi Wu, Xiaofeng Lin, and Wenzhong Guo. Vision-language action knowledge learning for semantic-aware action quality assessment. In *European Conference on Computer Vision*, pages 423–440, 2024. 3

[5] Huangbiao Xu, Xiao Ke, Huanqi Wu, Rui Xu, Yuezhou Li, and Wenzhong Guo. Language-guided audio-visual learning for long-term sports assessment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23967–23977, 2025. 3

[6] Huangbiao Xu, Xiao Ke, Huanqi Wu, Rui Xu, Yuezhou Li, Peirong Xu, and Wenzhong Guo. Dancefix: An exploration in group dance neatness assessment through fixing abnormal challenges of human pose. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8869–8877, 2025. 3

[7] Huangbiao Xu, Huanqi Wu, Xiao Ke, Yuezhou Li, Rui Xu, and Wenzhong Guo. Quality-guided vision-language learning for long-term action quality assessment. *IEEE Transactions on Multimedia*, 27:7326–7339, 2025. 3

[8] Huangbiao Xu, Huanqi Wu, Xiao Ke, Junyi Wu, Rui Xu, and Jinglin Xu. Mcmoe: Completing missing modalities with mixture of experts for incomplete multimodal action quality assessment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11241–11249, 2026. 3