Probabilistic Embeddings with Causal Constraint for Error Detection in Egocentric Procedural Videos

Anonymous ICME submission

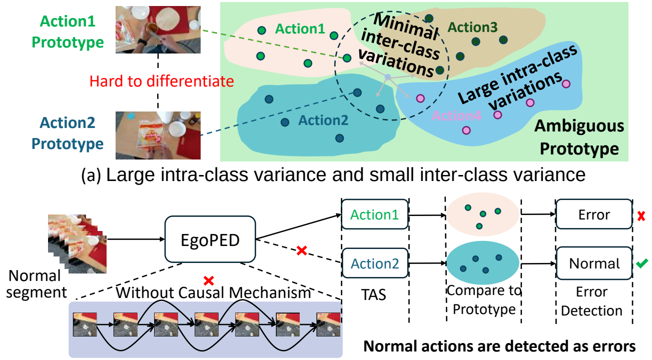
***Abstract*—Error detection in egocentric procedural task videos aims to identify deviations to support intelligent monitoring and task automation. Despite making significant progress, existing methods that leverage prototypes for egocentric error detection have two drawbacks: (1) The neglect of inherent data traits, i.e., large intra-class variance and minimal inter-class distinction. (2) The absence of causal consistency in temporal modeling. To address these challenges, we introduce a novel framework termed *Probabilistic Embeddings with Causal Constraint (PECC)* for error detection in egocentric procedural videos. Specifically, we first integrated a causal dilated convolution module in temporal action segmentation model to capture temporal causal consistency. We then train Gaussian Mixture Models (GMMs) for each action class to get frame-level probabilistic embeddings. Finally, We evaluate test frames using log-likelihood values to detect erroneous actions. Extensive experiments conducted on EgoPER and HoloAssist demonstrate that our method achieves state-of-the-art performance, significantly surpassing existing methods in error detection. Our anonymous code is available at https://anonymous.4open.science/r/PECC-07F0.**

***Index Terms*—Probabilistic Distribution Pepresentation, Gaussian Mixture Model, Egocentric, Error Detection**

I. Introduction

Driven by the need for intelligent monitoring, accurately detecting operational mistakes in egocentric videos has become a focal challenge. Traditional methods, typically based on anomaly detection, focus on frame-level analysis by learning normal behavior patterns and identifying deviations as anomalies. While these methods have shown effectiveness in controlled environments, they encounter difficulties in modeling the complex temporal dynamics and diverse action patterns characteristic of egocentric video streams. As a result, learning an appropriate probabilistic distribution to represent normal behavior has become a critical problem.

To address this problem, existing methods [7] employ a Temporal Action Segmentation (TAS) backbone for action segmentation, followed by a prototype learning module using K-means clustering. This approach allows the framework to learn multiple prototypes for each action class, serving as references for detecting deviations from normal task execution. As the first study to tackle this problem, it laid a foundational framework for detecting procedural errors in egocentric videos.



(a) Large intra-class variance and small inter-class variance

(b) Absence of causal consistency lead to wrong outputs in the model

Fig. 1. Illustration of the limitations of EgoPED [7], highlighting key challenges: (a) significant intra-class variance coupled with minimal inter-class variance, making distinction difficult, and (b) a lack of causal consistency, underscoring critical areas for improvement and innovation.

Despite existing methods achieving remarkable improvements over traditional methods, they still face two major challenges: (1) They failed to consider inherent data characteristics, such as considerable intra-class variation and weak inter-class separation. (2) The temporal modeling process lacks causality consistency. As illustrated in Fig. 1(a), previous methods overlook critical data characteristics, resulting in prototypes with significant intra-class variance and limited inter-class distinction, making them prone to ambiguities. Moreover, the close positioning of prototypes from different classes in the embedding space further increases the likelihood of incorrect error detection due to overlapping feature representations. Secondly, as shown in Fig. 1(b), the TAS backbone employed in EgoPED [7] lacks temporal causal consistency, which is crucial for procedural tasks in egocentric procedural task videos, leading to potential misclassifications of action segments. This, in turn, propagates errors in detecting errors associated with these segments, undermining the robustness of the error detection process.

To address these challenges, as illustrated in Fig. 2, we propose a novel framework termed ***P****robabilistic* ***E****mbeddings with* ***C****ausal* ***C****onstraint* (***PECC***) to enhance the ability for error detection in egocentric procedural task videos. This approach leverages causal mechanisms within the TAS model to improve the model's sensitivity to causal consistency. Furthermore, Gaussian Mixture Models (GMMs) are employed to model the probabilistic distribution of each normal action.

Specially, we first introduce a Causal Dilated Convolution (CDC) module into the TAS model to address the lack of causal consistency in egocentric procedural task videos. The CDC module effectively captures temporal causal consistency in videos, enhancing the model's ability to utilize causality. Second, we introduce a Gaussian-Based Probabilistic Model, wherein a separate Gaussian Mixture Model (GMM) is independently learned for each action class to capture the unique underlying statistical characteristics of error-free actions. During inference, log-likelihood scores computed by

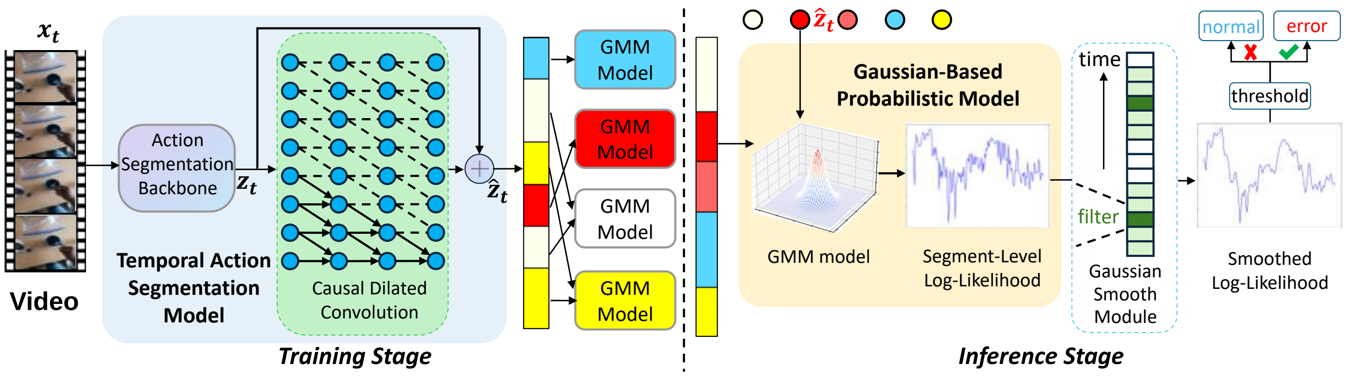


Fig. 2. We integrate a Causal Dilated Convolution module into the Temporal Action Segmentation (TAS) model to effectively capture the temporal consistency inherent in procedural task videos. After training the TAS model, we further train a distinct Gaussian Mixture Model (GMM) for each action class to capture its respective probabilistic distribution. For inference, we utilize the trained GMM for each segment to generate Log-Likelihood embeddings corresponding to its action class. Then we smooth the Segment-Level Log-likelihood and compare it to Log-likelihood thresholds for error detection.

the GMMs serve as Probabilistic Embeddings, identifying frames that deviate significantly from the learned distributions. This approach enables the model to distinguish erroneous frames effectively by leveraging probabilistic representations rather than relying on fixed prototypes. Finally, we introduce a Gaussian Smoothing Module, which utilizes a one-dimensional Gaussian filter, to smooth the raw log-likelihood scores generated by the GMM. This smoothing step reduces noisy fluctuations, producing more consistent and coherent segment-level mistake assessments. The resulting refinement enhances the robustness of the framework, enabling it to handle complex action sequences with higher accuracy and reliability. Our proposed method achieves state-of-the-art results on the EgoPER and HoloAssist datasets.

Overall, our contributions can be summarized as follows:

- We integrate a causal dilated convolution module into the TAS model to model temporal dependencies and causal relations effectively, improving temporal consistency.
- We propose a GMM-based framework for error recognition in egocentric video analysis, leveraging probabilistic log-likelihood evaluation to achieve enhanced robustness.
- We introduce a smoothing technique using Gaussian filters on GMM log-likelihood scores, enabling coherent and accurate segment-level error detection.

II. Related Work

Error Detection in Egocentric Procedural Task Videos. Error detection in procedural tasks seeks to identify mistakes that occur during the execution of a specified sequence of steps. This task differs from conventional anomaly detection, which focuses on global deviations within the data, by instead emphasizing the correctness of individual steps within a task. Recent advancements in procedural error detection include various novel methods. For instance, Lee et al. [7] proposed EgoPED, a framework that uses Contrastive Step Prototype Learning (CSPL) for error detection in procedural tasks. They also introduced the EgoPER dataset, specifically designed for this task, which includes multimodal data from cooking tasks with both normal and erroneous steps. Similarly, Flaborea et al. [4] introduced the PREGO model, a pioneering online error detection model tailored for open-set scenarios. PREGO integrates online action recognition and prediction modules, using large language models (LLMs) for symbolic reasoning to forecast whether upcoming steps conform to expected procedures. Furthermore, the Assembly101 [12] dataset, containing multi-view videos and coarse-grained action annotations, offers valuable resources for error identification research in assembly tasks. Unlike these existing approaches, our method focuses on modeling fine-grained temporal dependencies and leveraging probabilistic frameworks to enhance error detection accuracy in egocentric procedural tasks.

Probability Distribution Representation. In computer vision and multimodal learning, probabilistic representations are extensively used to capture data uncertainty and variability. Unlike deterministic models, probability distribution representations provide a richer characterization of intrinsic data variation, which strengthens model robustness and generalization. For example, Chun et al. [2] proposed the Probabilistic Cross-modal Embedding framework, which represents images and texts as probability distributions within a shared embedding space, effectively addressing the one-to-many correspondence problem between images and text. Ji et al. [6] introduced the MAP model, which employs a probabilistic distribution encoder to improve multimodal data handling and uncertainty management. In another advancement, Ahmine et al. [1] presented PNeRF, a probabilistic neural scene representation model capable of generating high-quality image renderings and accurate geometric reconstructions, even under sensor and pose uncertainty. Inspired by these works, we introduce probabilistic distribution representations into the task of error detection in egocentric procedural videos, modeling normal action distributions to enhance the differentiation between normal and erroneous behaviors.

Causal Mechanism. Causal dilated convolution has been widely adopted for temporal sequence modeling, capturing long-range dependencies and maintaining causality. Ding et al. [3] proposed using causal convolution networks to capture temporal dependencies, highlighting the role of causal relationships in mechanical system degradation. Mehta and Yang introduced NAC-TCN [10], combining dilated causal convolution with neighborhood attention to improve emotion recognition, reducing computation costs while achieving state-of-the-art performance. Hamad et al. [5] employed dilated causal convolutions to preserve temporal order, leveraging causal mechanisms for more effective human activity recognition. Inspired by these advancements, we incorporate dilated causal convolutions into our framework to model long-range temporal dependencies while maintaining the causality essential for detecting procedural errors in egocentric videos.

III. Our PECC Method

A. Preliminary

We adopt the task setup from the EgoPED framework [7], which originally proposed the procedural error detection task in egocentric videos of sequential tasks. The objective of this task is twofold: first, to segment the test video into task-related steps and background, and second, to identify frames containing procedural errors. A distinctive aspect of this task is that training is conducted using only normal (error-free) videos, while testing involves both normal and erroneous videos. Formally, let the dataset consist of N videos, each represented as a sequence of frames with pre-extracted features and corresponding step labels. For video n, the frame-wise features are denoted as $\mathbf{X}_n = \{\mathbf{x}_{n,1}, \mathbf{x}_{n,2}, \dots\}$ and their ground-truth step labels as $\mathbf{Y}_n = \{\mathbf{y}_{n,1}, \mathbf{y}_{n,2}, \dots\}$, where $\mathbf{x}_{n,t} \in \mathbb{R}^d$ is the d-dimensional feature vector of the t-th frame, and $\mathbf{y}_{n,t}$ is its action label. The TAS model is trained on normal videos using the action label sequence $\mathbf{Y}$, enabling the subsequent error detection process.

B. Causal Dilated Convolution

First, we focus on temporal action segmentation (TAS), where the goal is to classify each frame of the video into its corresponding action class. TAS typically consists of two main components: an action segmentation backbone and a classifier head. The backbone takes pre-extracted features $\mathbf{X}_n = \{\mathbf{x}_{n,1}, \mathbf{x}_{n,2}, \dots\}$ as input and learns refined, holistic frame-wise features $\mathbf{Z}_n = \{\mathbf{z}_{n,1}, \mathbf{z}_{n,2}, \dots\}$ by capturing long-range temporal dependencies across frames, where $\mathbf{z}_{n,t} \in \mathbb{R}^{d'}$ is a d'-dimensional feature vector representing the refined features of frame t in video n. The classifier head then assigns a label to each frame based on its corresponding refined feature vector. For clarity, we omit the video index n in the subsequent formulas, focusing on the frame-wise features $\mathbf{x}_t$ and their corresponding refined features $\mathbf{z}_t$ for a single video.

We incorporate a Causal Dilated Convolution (CDC) module between the two main components of TAS. This CDC module is specifically designed to enhance the model's capability to capture long-term dependencies and maintain temporal causal consistency. The CDC module refines the temporal features output by the backbone network. By sequentially processing frame features in temporal order, it enhances the temporal modeling capabilities while ensuring the crucial causal relationships between actions in procedural task videos.

For a convolution kernel with size k and dilation rate d, the receptive field of a CDC module is given by $(k-1) \times d + 1$, enabling the model to capture long-range causal relationships while maintaining computational efficiency. The operation performed by the CDC module can be formalized as:

$$\mathcal{F}_{\text{CDC}}(\mathbf{z}_t) = \sum_{i=0}^{k-1} \mathbf{w}_i \cdot \mathbf{z}_{t-i\cdot d}, \tag{1}$$

where $\mathcal{F}_{\text{CDC}}(\mathbf{z}_t)$ represents the output of the CDC module at time t, $\mathbf{w}_i$ denotes the convolutional kernel weights, and $\mathbf{z}_{t-i\cdot d}$ is the input feature at the dilated position.

By exponentially increasing d across layers (e.g., $d = 1, 2, 4, \dots$), the receptive field expands rapidly without a proportional increase in the number of parameters.

Furthermore, the output from the CDC module is combined with the backbone's output through a residual connection. The residual connection is defined as:

$$\hat{\mathbf{z}}_t = \mathbf{z}_t + \mathcal{F}_{\text{CDC}}(\mathbf{z}_t), \tag{2}$$

C. Gaussian-Based Probabilistic Model

To model the probabilistic distribution within actions, we propose a Gaussian-Based Probabilistic Model. In this model, Gaussian Mixture Models (GMMs) are employed as fundamental components. The intermediate features $\hat{z}_t$ from a trained TAS model are used to train a GMM for each action class. The GMM models the probability distributions of frame features, effectively capturing normal variations for each action. We utilize its output to obtain probabilistic embeddings for each frame. Furthermore, these embeddings can be employed for the purpose of error detection.

The GMM is a probabilistic model that assumes each data point is generated from a mixture of Gaussian distributions, with K representing the total number of Gaussian components in the mixture, each associated with a weight. The probability density function of the GMM, defined over a multi-dimensional vector $\mathbf{x}$, is:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \omega_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{3}$$

where $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ represents the parameters of the GMM: ω_k is the weight of the k-th Gaussian component, $\boldsymbol{\mu}_k$ is its mean vector, and $\boldsymbol{\Sigma}_k$ is its covariance matrix.

GMM Training via Expectation-Maximization (EM). The EM algorithm optimizes the GMM parameters $\boldsymbol{\theta}$, by maximizing the total expected log-likelihood of the data. Let $\hat{\mathbf{Z}}^i = \{\hat{\mathbf{z}}_1^i, \hat{\mathbf{z}}_2^i, \dots\ , \hat{\mathbf{z}}_T^i\}$ represent all intermediate frame-wise features for a specific action $i \in \{1, 2, \dots, S+1\}$, where T is the total number of frames for action i and S denotes the total number of action classes, label $S+1$ for the background class. The expected log-likelihood is expressed as:

$$\log L(\boldsymbol{\theta} \mid \hat{\mathbf{Z}}^i) = \sum_{t=1}^{T} \log \left(\sum_{k=1}^{K} \omega_k \mathcal{N}(\hat{\mathbf{z}}_t^i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right), \tag{4}$$

The optimization alternates between the following two steps:

- **E-Step**: Computes the responsibility $\gamma_{t,k}$, which represents the posterior probability of the k-th Gaussian component given $\hat{\mathbf{z}}_t^i$:

$$\gamma_{t,k} = \frac{\omega_k \mathcal{N}(\hat{\mathbf{z}}_t^i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \omega_j \mathcal{N}(\hat{\mathbf{z}}_t^i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}, \tag{5}$$

- **M-Step**: updates the parameters to maximize the expected log-likelihood:

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_\gamma \left[\log L(\boldsymbol{\theta} \mid \hat{\mathbf{Z}}^i) \right], \tag{6}$$

This yields the following updates:

$$\boldsymbol{\omega}_k = \frac{1}{T} \sum_{t=1}^{T} \gamma_{t,k}, \quad \mu_k = \frac{\sum_{t=1}^{T} \gamma_{t,k} \hat{\mathbf{z}}_t^i}{\sum_{t=1}^{T} \gamma_{t,k}}, \tag{7}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{t=1}^{T} \gamma_{t,k} (\hat{\mathbf{z}}_t^i - \mu_k)(\hat{\mathbf{z}}_t^i - \mu_k)^T}{\sum_{t=1}^{T} \gamma_{t,k}}, \tag{8}$$

The algorithm iteratively alternates between these steps until the parameters converge.

Log-Likelihood for Error Detection. At inference stage, through the TAS model, each test frame $\mathbf{x}_t$ can be classified into a specific action class, denoted by j, where $j \in \{1, 2, \ldots, S+1\}$. Additionally, for each action class j, a GMM is trained, denoted as $\boldsymbol{\theta}_j = \{\pi_k^j, \boldsymbol{\omega}_k^j, \boldsymbol{\Sigma}_k^j\}_{k=1}^K$. Given the intermediate feature $\hat{\mathbf{z}}_t$ of frame $\mathbf{x}_t$ belonging to class j, the log-likelihood of $\hat{\mathbf{z}}_t$ under the GMM $\boldsymbol{\theta}_j$ is computed as:

$$\log p(\hat{\mathbf{z}}_t | \boldsymbol{\theta}_j) = \log \left(\sum_{k=1}^{K} \omega_k \mathcal{N}(\hat{\mathbf{z}}_t | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right). \tag{9}$$

This log-likelihood value is used to assess the likelihood that the frame $\mathbf{x}_t$, follows the distribution of the action class j. By calculating this value for each frame, we can identify frames that deviate significantly from the expected distributions, thus enabling error detection.

Log-Likelihood Thresholds. To classify frames as normal or erroneous, we define 41 log-likelihood thresholds for each action, denoted as the set $\boldsymbol{\Gamma} = \{\gamma_1, \gamma_2, \ldots, \gamma_{41}\}$. These thresholds are evenly sampled between the minimum and median log-likelihood values from the training set. The choice of 41 aligns with the configuration of the EgoPED [7].

D. Gaussian Smooth Module

We get log-likelihood values of every frame in inference stage for error detection. However, Frame-level log-likelihood values can fluctuate due to noise or transient inconsistencies, potentially leading to false positives or negatives. To mitigate this, we apply a Gaussian Smooth Module (GSM) using a one-dimensional Gaussian filter to smooth the log-likelihood values across each action segment. This smoothing is crucial because it aggregates frame-level errors into more stable segment-level assessments, ensuring that errors are detected more consistently over time. Specifically, we perform the smoothing by applying the Gaussian kernel across the frames within a segment, which takes into account not only the current frame's likelihood but also the surrounding frames.

As illustrated on the right side of Fig. 2, the smoothed log-likelihood values are evaluated against a predefined threshold. Given a threshold $\gamma \in \boldsymbol{\Gamma}$, a frame is classified as **normal** if its log-likelihood value is greater than or equal to γ; otherwise, it is classified as an **error**.

Finally, segment-level decisions are made using a majority voting mechanism:

- If the majority of frames in a segment are classified as normal, the segment is considered normal.
- Otherwise, the segment is classified as erroneous.

The detailed pseudocode of our framework is provided in the supplementary materials.

IV. Experiment

A. Experimental Setup

Datasets. We evaluate our model on the EgoPER [7] and HoloAssist [13] datasets. The EgoPER dataset comprises five independent tasks, each of which is trained and tested separately. For each of these five tasks, we utilize RGB data and active object detection (AOD) information to train our model. The HoloAssist dataset is a large-scale egocentric human interaction dataset containing 2,221 egocentric videos, with annotations provided for error detection tasks. We only use RGB data for training in HoloAssist. Specially, we separately train and evaluate the model using verbs or nouns as action labels in HoloAssist. For EgoPER and HoloAssist, data is split into 80% normal videos for training, 10% for validation, and the remaining 10%, along with all error videos, for testing.

Evaluation Metrics. We use multiple metrics to evaluate the performance of error detection and action segmentation. For error detection, we first calculate the Segment-level Error Detection Accuracy (EDA), defined as De/GTe, where De and GTe represent the number of correctly predicted segments and the total number of segments in all test videos, respectively. Additionally, we employ micro AUC based on frame-level error predictions to assess frame-level error detection capability; hereafter, AUC refers exclusively to micro AUC.

Implementation Details. We utilize the I3D model to extract RGB features. Additionally, for the EgoPER dataset, we employ Graph Convolutional Networks (GCNs), as in EgoPED, to extract active object detection (AOD) information. We use ActionFormer [15] and MSTCN++ [8] as backbone models for action segmentation, with CDC modules incorporated into each backbone. During training, we first train the TAS backbone. Subsequently, we train a Gaussian-Based Probabilistic Model by learning a separate GMM for each action class using intermediate features extracted from all videos in the training set. The log-likelihood of each frame in the training set is computed to establish appropriate thresholds for each action class. During inference, these thresholds are used by comparing them with the GMM predictions for each frame of the test set videos to detect errors.

B. Overall Comparsion Results

We compare our proposed method PECC with recent state-of-the-art methods, including: (1) Traditional anomaly detec-

TABLE I

Error detection results of different methods on the EgoPER dataset for each task and the average over all tasks.

Method	Quesadilla		Oatmeal		Pinwheel		Coffee		Tea		Average	
	EDA	**AUC**	**EDA**	**AUC**	**EDA**	**AUC**	**EDA**	**AUC**	**EDA**	**AUC**	**EDA**	**AUC**
Random	19.9	50.0	11.8	50.0	15.7	50.0	8.20	50.0	17.0	50.0	14.5	50.0
HF^2-VAD [9] (ICCV'21)	34.5	62.6	25.4	62.3	29.1	52.7	10.0	59.6	36.6	62.1	27.1	59.9
SSPCAB [11] (CVPR'22)	30.4	60.9	25.3	61.9	33.9	51.7	10.0	60.1	35.4	63.2	27.0	59.6
S3R [14] (ECCV'22)	52.6	51.8	47.8	61.6	50.5	52.4	16.3	51.0	47.8	57.9	43.0	54.9
EgoPED [7] (CVPR'24)	62.7	65.6	51.4	65.1	59.6	55.0	55.3	58.3	56.0	66.0	57.0	62.0
PECC (Ours)	**79.4**	**75.4**	**88.4**	**67.1**	**77.8**	**63.7**	**83.1**	**64.5**	**77.1**	**66.6**	**81.2**	**67.5**

TABLE II

The results of error detection on HoloAssist.

Method	Verb		Noun	
	EDA	**AUC**	**EDA**	**AUC**
Random	11.2	50.0	13.0	50.0
HF^2-VAD [9] (ICCV'21)	24.0	38.0	23.2	38.2
SSPCAB [11] (CVPR'22)	23.7	38.0	22.9	39.1
S3R [14] (ECCV'22)	51.2	48.6	51.6	49.5
EgoPED [7] (CVPR'24)	68.0	47.3	71.0	50.8
PECC (Ours)	**93.1**	**54.7**	**77.1**	**55.0**

TABLE III

Ablation Study of Method Components on EgoPER

Method	GBP	CDC	GSM	EDA	AUC
w/o GBP&GSM		✓		66.1	63.5
w/o CDC&GSM	✓			73.1	64.0
w/o CDC	✓		✓	75.2	64.9
w/o GSM	✓	✓		74.1	65.9
PECC (Ours)	✓	✓	✓	**81.2**	**67.5**

tion methods that is known for their strong generalizability and robustness across diverse tasks and datasets, such as HF2-VAD [9], SSPCAB [11] , S3R [14] and (2) EgoPED [7], considered the pioneer in error detection for egocentric procedural task videos, is a key method we focus on for comparison.

Error Detection Results. Tables I and II demonstrate that our method achieves state-of-the-art performance on both the EgoPED and HoloAssist datasets, consistently surpassing all baselines in EDA and AUC scores. For instance, our method achieves superior average EDA and AUC on EgoPED, significantly outperforming EgoPED and other traditional methods. Similarly, it achieves the highest scores on HoloAssist when using verbs or nouns as action class labels, further highlighting its robustness and effectiveness. The superior performance of our method is attributable to the inclusion of the CDC module, which effectively captures the causal relationships in action sequences. Additionally, the probabilistic representation learned from the GMMs model enables the model to better learn the distributional characteristics of action classes.

C. Further Analysis

Ablation Studies. Table III shows the ablation study results for the key components of our method on EgoPER dataset. Here, **GBP** refers to the Gaussian-Based Probabilistic Model for error detection, while omitting GBP implies a prototype-based approach. **CDC** denotes the Causal Dilated Convolution module in the TAS backbone, and **GSM** represents the Gaussian Smooth Module. From the table, we observe that the inclusion of each module contributes positively to the overall performance, with **Ours** (using all three components) achieving the highest metrics. Notably, removing the CDC or GSM modules results in a notable performance drop, highlighting their importance in enhancing error detection. Additionally, replacing GBP with a prototype-based approach leads to a further decline in performance, indicating that the Gaussian-Based Probabilistic Model is more effective for this task. Overall, these results underline the complementary roles of GBP, CDC, and GSM in improving both EDA and AUC metrics. We also evaluated the impact of TAS backbone (ActionFormer [15] and MSTCN++ [8]) on framework performance, with detailed results in the supplementary materials.

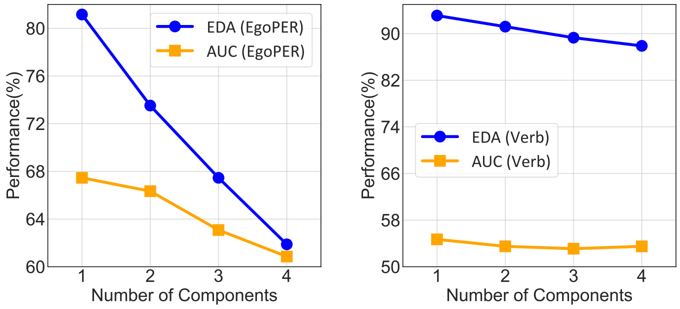


Fig. 3. Error Detection metrics for different numbers of components in the GMM on EgoPER (left) and HoloAssist (right), using verb as the class label.

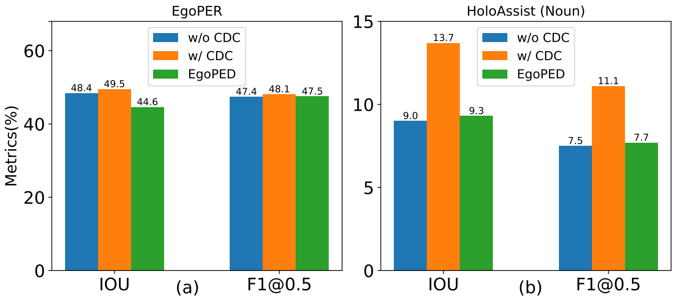


Fig. 4. Analysis of the CDC module on TAS metrics (IOU and F1@0.5) for EgoPER (left) and HoloAssist (right) using ActionFormer.

Analysis on Number of Components in GMM. As shown in Fig. 3, the performance of the action error detection method on both the EgoPER and HoloAssist datasets varies with the number of components in the GMM. In general, using fewer components leads to better results, while increasing the components causes a decline in metrics. This suggests that modeling the data distribution with fewer, more concentrated components helps capture the essential patterns more effectively, leading to improved error detection performance.

Analysis on CDC Module for TAS. Here, we compare the TAS performance of ActionFormer [15] under three configurations: (1) the original ActionFormer without the CDC module, (2) ActionFormer enhanced with the CDC module, and (3) the EgoPED method, which integrates Contrastive Prototype Learning (CSPL) to enhance ActionFormer. As shown in Fig. 4, the results demonstrate that ActionFormer with the CDC module consistently outperforms both the original ActionFormer (without CDC) and EgoPED, highlighting the robustness and effectiveness of the CDC module in enhancing temporal action segmentation. While EgoPED generally outperforms the original ActionFormer, it occasionally underperforms even the original version, indicating the instability of CSPL compared to the CDC module.

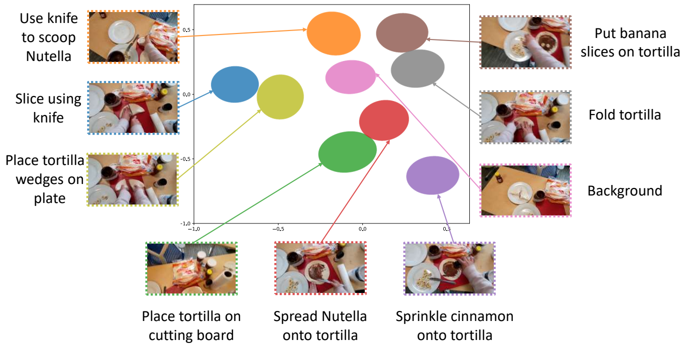


Fig. 5. Visualization of the 2D projections of the GMM distributions learned for each action in the quesadilla task from the EgoPED dataset. Each elliptical region represents the GMM distribution of a specific action, with clear separations between actions, facilitating accurate error detection.

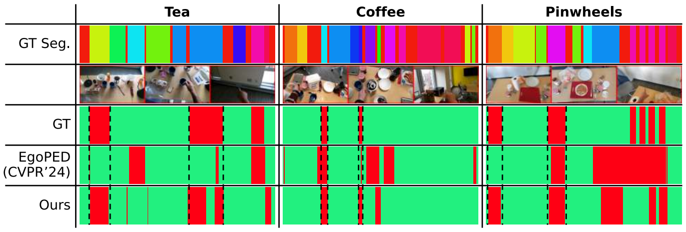


Fig. 6. Comparison of error detection on tea, coffee, and pinwheels tasks in EgoPER. The first row represents the ground truth (GT) action segmentation. The second row visualizes several frames from the sequences. In the subsequent three rows, red segments indicating incorrect frames and green segments indicating correct frames. These three rows depict erroneous segments identified in GT, EgoPED, and our proposed method, respectively.

Qualitative Analysis. As shown in Fig. 5, the projection of the learned GMM distribution for each action in the 2D space demonstrates that each action corresponds to a distinct region. These well-separated regions facilitate effective error detection. The GMMs effectively capture the underlying data characteristics, addressing the issue of large intra-class variance and small inter-class variance observed in the original EgoPED prototypes. By modeling the data distribution more precisely, our method significantly improves the ability to distinguish between normal and erroneous frames.

As illustrated in Fig. 6, our proposed method demonstrates superior error detection performance across the tea, coffee, and pinwheels tasks in the EgoPER dataset. Compared to EgoPED, our approach identifies erroneous segments with higher accuracy, significantly reducing false positives where normal frames are incorrectly detected as errors. This improvement highlights the robustness of our method in distinguishing correct and incorrect frames.

V. Conclusion

In this paper, we proposed a probabilistic framework, ***P**robabilistic **E**mbeddings with **C**ausal **C**onstraint (**PECC**)*, combining causal mechanism and Gaussian-Based Probabilistic Model to improve error detection in egocentric procedural videos. By leveraging a causal dilated convolution module for temporal dependencies and learning GMMs for the probabilistic distributions of normal actions, our method consistently outperformed other methods across key metrics on the EgoPER and HoloAssist datasets. These results highlight the framework's effectiveness and potential for accurate and robust error detection in egocentric procedural tasks.

References

[1] Y. Ahmine, A. Dey, and A. I. Comport, "Pnerf: Probabilistic neural scene representations for uncertain 3d visual mapping," *CoRR*, vol. abs/2209.11677, 2022.

[2] S. Chun, S. J. Oh, R. S. de Rezende, Y. Kalantidis, and D. Larlus, "Probabilistic embeddings for cross-modal retrieval," in *CVPR*, 2021, pp. 8415–8424.

[3] W. Ding, J. Li, W. Mao, Z. Meng, and Z. Shen, "Rolling bearing remaining useful life prediction based on dilated causal convolutional densenet and an exponential model," *Reliab. Eng. Syst. Saf.*, vol. 232, p. 109072, 2023.

[4] A. Flaborea, G. M. D. di Melendugno, L. Plini, L. Scofano, E. D. Matteis, A. Furnari, G. M. Farinella, and F. Galasso, "PREGO: online mistake detection in procedural egocentric videos," in *CVPR*, 2024, pp. 18 483–18 492.

[5] R. A. Hamad, M. Kimura, L. Yang, W. L. Woo, and B. Wei, "Dilated causal convolution with multi-head self attention for sensor human activity recognition," *Neural Comput. Appl.*, vol. 33, no. 20, pp. 13 705–13 722, 2021.

[6] Y. Ji, J. Wang, Y. Gong, L. Zhang, Y. Zhu, H. Wang, J. Zhang, T. Sakai, and Y. Yang, "MAP: multimodal uncertainty-aware vision-language pretraining model," in *CVPR*, 2023, pp. 23 262–23 271.

[7] S. Lee, Z. Lu, Z. Zhang, M. Hoai, and E. Elhamifar, "Error detection in egocentric procedural task videos," in *CVPR*, 2024, pp. 18 655–18 666.

[8] S. Li, Y. A. Farha, Y. Liu, M. Cheng, and J. Gall, "MS-TCN++: multi-stage temporal convolutional network for action segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 6647–6658, 2023.

[9] Z. Liu, Y. Nie, C. Long, and Q. Zhang, "A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction," in *ICCV*. IEEE, 2021, pp. 13 568–13 577.

[10] A. Mehta and W. Yang, "NAC-TCN: temporal convolutional networks with causal dilated neighborhood attention for emotion understanding," in *ICVIP*, 2023, pp. 9–16.

[11] N. Ristea, N. Madan, R. T. Ionescu, and K. Nasrollahi, "Self-supervised predictive convolutional attentive block for anomaly detection," in *CVPR*, 2022, pp. 13 566–13 576.

[12] F. Sener, D. Chatterjee, D. Shelepov, and K. He, "Assembly101: A large-scale multi-view video dataset for understanding procedural activities," in *CVPR*, 2022, pp. 21 064–21 074.

[13] X. Wang, T. Kwon, M. Rad, B. Pan, I. Chakraborty, S. Andrist, D. Bohus, A. Feniello, B. Tekin, F. V. Frujeri, N. Joshi, and M. Pollefeys, "Holoassist: an egocentric human interaction dataset for interactive AI assistants in the real world," in *ICCV*, 2023, pp. 20 213–20 224.

[14] J. Wu, H. Hsieh, D. Chen, C. Fuh, and T. Liu, "Self-supervised sparse representation for video anomaly detection," in *ECCV*, ser. Lecture Notes in Computer Science, vol. 13673, 2022, pp. 729–745.

[15] C. Zhang, J. Wu, and Y. Li, "Actionformer: Localizing moments of actions with transformers," in *ECCV*, ser. Lecture Notes in Computer Science, S. Avidan, G. J. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., vol. 13664, 2022, pp. 492–510.