Test-time Ego-Exo Adaptation for Action Anticipation via Multi-Label Prototype Growing and Dual-Clue Consistency

Supplementary Material

A. Details of State-of-the-Art Methods

In this section, we introduce some technical and re-implementation details of the related state-of-the-art comparison methods.

A.1. Tent

Tent [9] is the pioneering work that utilizes target-domain data to adjust the trained model during test time via an entropy minimization strategy. It systematically defines the settings of fully test-time adaptation (TTA), i.e., fully TTA is independent of the training data and training loss, and adapts the model by optimizing an unsupervised loss during testing based on the unlabeled test data stream. Technically, it adjusts the affine parameters in batchnorm layers by minimizing the calculated entropy of the predicted logits. In this paper, we re-implement Tent for the TE^2A^3 task under the same settings as with our method. Due to the action anticipation model [3] not containing batchnorm layers, we practically select the parameters of the bias layers in the anticipation head, and the learning rate is set to 1e-4 for *EgoExoLearn* and 1e-3 for *EgoMe-anti* for optimization.

A.2. TPT

TPT [7] is the first work to perform prompt tuning based on the CLIP model. TPT achieves test-time adaptation by augmenting a single test image to diverse views and minimizing entropy to optimize the learnable prompts. In this paper, we conduct re-implementation according to the "TPT+CoOp" setting in this work. Since our TE^2A^3 task focuses on the adaptation of video rather than image, we adjust the original spatial cropping augmentation to the temporal re-sampling strategy to suit our task. In addition, we initialize the learnable prompts with "a photo of a", whose length is 4 as in our work. For optimization, the learning rate is set to 1e-5 on the *EgoExoLearn* and 1e-4 for the *EgoMe-anti* benchmark.

A.3. VITTA

VITTA [5] is the first TTA work for video-level action recognition. To address unanticipated distribution shifts of different-domain videos, VITTA first adopts a feature distribution alignment technique to align the estimated online statistics of the training and testing sets. Additionally, it constrains the predictions of different temporally augmented views to construct consistency. We re-implement this work according to its original settings. In detail, we first save the mean and variance for the features of the training set under all settings. Then, we employ momentum updating to obtain the mean and variance of the test set. Moreover, we adopt L1 losses to align the training and testing mean and variance statistics of each layer, and establish consistency between temporal augmented views. The learning rates are set to 1e-5 and 1e-4 for *EgoExoLearn* and *EgoMe-anti*, respectively. The quantitative results show that, even though VITTA accesses the statistics of the training set, our method still outperforms it by a large margin.

A.4. TDA

TDA [4] is a training-free method for effective and efficient test-time adaptation. To tackle the distribution shifts between the source and target domains, TDA introduces a lightweight key-value cache and progressively refines the saved pseudo-labels for efficient adaptation to test data. Moreover, TDA also incorporates a negative cache branch to mitigate the impact of noise and achieves high performance for image-level TTA. In this paper, we adjust the hand-crafted prompt to be the same as our method (i.e., "a photo of a") for a fair comparison. Furthermore, other hyperparameters such as threshold values for negative pseudo-labeling and testing feature selection are set following the official settings for re-implementation.

A.5. ZERO

ZERO [2] is a simple yet effective method by setting the softmax temperature to zero. In detail, to address the model's poor generalization capability when presented with challenging examples, ZERO conducts thorough theoretical and empirical analysis. Technically, ZERO first augments the input image into multiple views, then feeds the images into the model to obtain anticipation logits. Next, it remains the confident predictions, and it finally sets the softmax temperature to zero. Following this paradigm, we conduct the temporal augmentation instead of the spatial augmentation specific to the video-level task. Then, we re-implement the ZERO method under its official settings for our TE^2A^3 task on the *EgoExoLearn* and *EgoMe-anti* benchmarks.

A.6. TCA

TCA [10] stands for Token Condensation Adaptation, which is an efficient training-free adaptation method to progressively refine the token selection process during test time. In detail, TCA first introduces a token reservoir to track and store class tokens sensitive to the domain shifts. Then, TCA dynamically adjusts multi-head attention maps based on the current data and accumulated knowledge. Fi-

nally, predictions are inferred not only from image-text similarity but also from correlations between visual features and tokens, which facilitates leveraging historical information for effective and efficient adaptation. In this paper, we re-implement this excellent work for the proposed TE^{2}A^3 task on the *EgoExoLearn* and *EgoMe-anti* benchmarks according to the official setting of TCA.

A.7. ML-TTA

ML-TTA [11] is a multi-label TTA method using the bound entropy minimization strategy. In detail, ML-TTA first describes the input images to determine the number of positive classes. Then, it infers a weak and a strong label set. Next, ML-TTA performs the label-binding strategy to bind the potential positive classes to mitigate the over-confident problem of plain entropy-based optimization, and conducts prompt learning for the two branches, respectively. The inferred logits of the two branches are added to obtain the final result. While effective, ML-TTA is for image-level TTA rather than video-level Ego-Exo adaptation and anticipation. In this paper, we directly utilize fine-level annotations from the original datasets [3, 6] as descriptions for the input target-view videos. The learning rates are set to 1e-4 and 5e-4 for the *EgoExoLearn* and *EgoMe-anti* benchmarks for optimization. Despite our method employing the predicted descriptions while ML-TTA utilizes manually labeled caption annotations, our approach still surpasses ML-TTA by a large margin under all settings, which demonstrates the effectiveness of the proposed method.

B. Details of Lightweight Narrator

In this section, we introduce more architectural and implementation details of the lightweight narrator $\mathcal{N}(\cdot)$ in the Dual-Clue Consistency Module (DCCM) in Section 3.3.

The architecture of the $\mathcal{N}(\cdot)$ follows "S2VTAttModel" in the official video-caption.pytorch library, which is a well-known open-source library for video captioning. The narrator is composed of an encoder and a decoder based on linear layers, GRU units, and attention layers. Assuming the input video frame features are denoted as F^T, we first convert F^T into sequential features, as follows:

$$X^T = \{x_1^T, x_2^T, \cdots, x_L^T\} = \mathcal{F}_V(F^T) \tag{1}$$

where $\mathcal{F}_V(\cdot)$ is an MLP comprising three linear layers, $X^T = \{x_1^T, x_2^T, \cdots, x_L^T\} \in \mathbb{R}^{L\times C}$ means the converted input features, L denotes the number of input video frames. Then, we utilize GRU units to model the temporal dependencies between the video frames. Taking the t-th timestep as an example, the calculation process is as follows:

$$r_t = s(W_{ir}x_t^T + b_{ir} + W_{hr}\hat{h}_{t-1} + b_{hr}) \tag{2}$$

$$z_t = s(W_{iz}x_t^T + b_{iz} + W_{hz}\hat{h}_{t-1} + b_{hz}) \tag{3}$$

$$n_t = \tanh(W_{in}x_t^T + b_{in} + r_t * (W_{hn}\hat{h}_{t-1} + b_{hn})) \tag{4}$$

$$\hat{h}_t = (1 - z_t) * n_t + z_t * \hat{h}_{t-1} \tag{5}$$

where $\hat{h}_t$ is the hidden state at the timestamp t, $s(\cdot)$ is the sigmoid function, $*$ is the Hadamard product, and W, b are learnable weights and biases, respectively. The output of the encoder can be represented as $\hat{H} = \{\hat{h}_1, \hat{h}_2, \cdots \hat{h}_L\} \in \mathbb{R}^{L\times C_h}$, where C_h is the dimension of the hidden state.

In the decoder, an attention mechanism is first applied:

$$O_1 = \tanh(W_1[\hat{H}; t(\hat{h}_L)] + b_1) \tag{6}$$

$$Att = \{\alpha_1, \alpha_2, \cdots \alpha_L\} = \sigma(W_2O_1 + b_2) \tag{7}$$

$$c_{tx} = \sum_{i=1}^{L} \alpha_i \cdot \hat{h}_i \tag{8}$$

where $t(\cdot)$ is the tile operation, σ is softmax function, $c_{tx} \in \mathbb{R}^{C_h}$ denotes the context feature. Then, we concatenate the context feature c_{tx} with embeddings of the decoded words (each sentence begins with the <sos> token), and initialize the hidden state of the decoder with the encoder's final hidden state $\hat{h}_L$. Finally, we adopt a GRU-based layer and a 2-layer MLP to decode the final sentence autoregressively.

For practical implementation, we collect view-agnostic video-text pairs $\{V_i, T_i\}_{i=1}^N$ for training the narrator. In detail, we first remove all videos and corresponding annotations relevant to the data for test-time adaptation in the proposed TE^{2}A^3 task under all settings from the original EgoMe [6] and EgoExoLearn [3] datasets. Then, we integrate the Ego and Exo video clips and the raw textual descriptions within the original dataset to construct the video captioning dataset $\{V_i, T_i\}_{i=1}^N$. However, the description labeling rules are different for Ego and Exo videos in the EgoMe dataset (i.e., descriptions of Exo videos include attributes of the demonstrator). Therefore, we uniformly preprocess the texts in the video captioning dataset to concentrate on procedural activities and exclude potential view-specific information. Finally, we utilize the above video captioning dataset to train the lightweight narrator $\mathcal{N}(\cdot)$ to generate view-agnostic textual clues, facilitating the adaptation between Ego and Exo views during test time. Specifically, we set the feature dimension C_h to 512. The maximum sentence length is set to 28, and each sentence begins with the <sos> token and ends with the <eos> token. Additionally, the sampling strategy of video frame features is identical to that in the action anticipation network (i.e., uniformly sample 5 frames as input). We train the narrator network for 1000 epochs with a batch size of 1024. The dropout rate is 0.5 and 0.2 for the linear and GRU-based layers, respectively, which helps alleviate overfitting. Moreover, we set the initial learning rate to 5e-4 and apply a learning rate decay strategy, in which the learning rate is reduced to 80% of its original value every 200 epochs.

Table A1. The number of samples under each view of the *train*/*val*/*test* set of the *EgoMe-anti* and *EgoExoLearn* benchmarks.

Subset	*EgoMe-anti*				*EgoExoLearn*			
	Exo		Ego		Exo		Ego	
	Noun	Verb	Noun	Verb	Noun	Verb	Noun	Verb
Train set	8336	8336	8939	8939	11762	17516	78055	33496
Validation set	1817	1817	1970	1970	8235	1960	84726	91454
Test set	3584	3584	3889	3889	12395	5371	15231	50471

C. Details of the Benchmarks

In this section, we show more details of our newly proposed *EgoMe-anti* and the existing *EgoExoLearn* benchmarks to support the TE^{2}A^3 task.

C.1. EgoMe-anti

The *EgoMe-anti* benchmark is constructed based on the recent EgoMe [6] dataset, which follows the imitation learning process in real-world scenarios. In detail, it contains 82.8 hours of Exo observation and Ego following videos and covers up to 41 real-world scenarios such as the library, classroom, gym, and so on. The video durations range widely from 3 seconds to over 60 seconds, and most of the videos last from 10 to 25 seconds. Moreover, each video contains 2 to above 15 fine-level consecutive and non-overlapped atomic actions with timestamps and descriptions. We exclude the incorrectly following videos and leverage the fine-level timestamps and description annotations to construct the *EgoMe-anti* benchmark, whose sample numbers of the *train*/*val*/*test* sets are shown in the left panel of Table A1. Because we extract nouns and verbs from the raw sentences simultaneously, the number of samples remains identical in noun or verb anticipation tasks under the same perspective. Finally, we summarize 35 noun and 29 verb categories for the action anticipation task.

C.2. EgoExoLearn

The *EgoExoLearn* benchmark in our work directly follows the official cross-view action anticipation benchmark proposed in EgoExoLearn [3], which is an excellent dataset containing large-scale asynchronous Ego and Exo videos of procedural activities in daily and professional scenarios. Specifically, it comprises up to 745 long videos with a total duration of around 120 hours. The video recording is oriented towards 5 types of daily cooking tasks and 3 types of chemical laboratory tasks, which are recorded in 4 kinds of different kitchens and 3 kinds of laboratories to ensure data diversity. This dataset also contains detailed coarse-grained and fine-grained timestamps, descriptions, and noun/verb categories. Therefore, the *EgoExoLearn* benchmark for the action anticipation task is constructed based on the fine-grained annotations with massive samples, and the number of samples of each subset in the Ego or Exo perspectives is represented in the right panel in Table A1. In addition, *EgoExoLearn* has 31 noun and 19 verb categories for the action anticipation task.

D. Results in Top-1 Recall Evaluation

In addition to the common Top-5 recall evaluation metric [1, 3], we also evaluate our DCPGN and comparison methods under the Top-1 recall metric for more comprehensive evaluation. As shown in Table A2, our method still outperforms other comparison methods by a large margin. In detail, it surpasses the most recent ML-TTA [11] by 4.52%, 10.11%, 5.43%, 7.69% on the *EgoMe-anti* benchmark, and by 6.75%, 4.15%, 6.29%, 8.85% on the *EgoExoLearn* benchmark, which further demonstrates the effectiveness and superiority of the proposed DCPGN.

E. Analysis of Hyperparameters

In this section, we conduct comprehensive experiments to analyze several hyperparameters in the proposed method. Moreover, we report the results under Exo2Ego and Ego2Exo settings on the *EgoMe-anti* and *EgoExoLearn* benchmarks.

E.1. Analysis of memory bank capacity N

In Table A3, we conduct experiments under different settings of the hyperparameter N, which denotes the maximum capacity of the memory bank for each class. Specifically, we set the value of N to 100, 200, 300, 400, and 500 and report the number of parameters (including the saved data in the memory banks) and quantitative results under all settings on the *EgoMe-anti* and *EgoExoLearn* benchmarks. The experimental results show that the performances fluctuate within a relatively small range, and the number of parameters is tolerable under all settings of N. It demonstrates the effectiveness of the proposed memory banks in the ML-PGM, which can achieve high performance even though the memory bank capacity is limited. Moreover, the model reaches the highest performance when N is 500. Considering its superior performance and tolerable number of parameters, we finally set N to 500 in our DCPGN.

Table A2. Quantitative results on the *EgoMe-anti* and *EgoExoLearn* under the Exo2Ego and Ego2Exo settings in Top-1 recall evaluation.

Methods	*EgoMe-anti*				*EgoExoLearn*			
	Exo2Ego		Ego2Exo		Exo2Ego		Ego2Exo	
	Noun	Verb	Noun	Verb	Noun	Verb	Noun	Verb
Ours without Adaptation	27.30	6.24	22.72	6.45	6.07	5.43	6.90	5.35
Tent [9]	31.72	8.09	26.92	8.98	7.72	6.57	7.98	5.40
TPT [7]	31.97	8.05	27.14	8.89	8.11	6.35	8.13	5.43
VITTA [5]	32.21	8.75	26.87	9.14	7.49	6.13	8.27	5.56
TDA [4]	33.42	10.60	27.18	10.46	8.26	6.59	8.41	5.48
ZERO [2]	32.50	10.02	25.82	10.26	9.48	8.73	10.27	7.42
TCA [10]	32.35	8.45	27.36	9.07	7.85	6.22	8.05	5.55
ML-TTA [11]	33.48	10.37	27.80	10.06	10.39	7.40	10.01	5.38
DCPGN (Ours)	**38.00**	**20.48**	**33.23**	**17.75**	**17.14**	**11.55**	**16.30**	**14.23**

Table A3. Analysis of the maximum capacity N of the memory bank. Note that the saved data in the memory banks is also included in the statistics of the number of parameters.

N	*EgoMe-anti*				*EgoExoLearn*				#Params (M)
	Exo2Ego		Ego2Exo		Exo2Ego		Ego2Exo		
	Noun	Verb	Noun	Verb	Noun	Verb	Noun	Verb	
100	78.99	43.64	71.97	39.62	45.07	42.60	47.48	45.58	1.71
200	**79.24**	42.47	71.97	39.71	45.36	42.72	47.87	45.61	3.42
300	78.95	42.95	71.81	39.82	45.50	**43.00**	48.18	45.99	5.13
400	78.86	43.17	71.57	38.95	45.78	42.74	48.06	46.10	6.84
500	79.03	**43.84**	**72.01**	**40.10**	**46.26**	42.98	**48.48**	**46.51**	8.54

E.2. Analysis of hyperparameter α

In Table A4, we conduct quantitative experiments under all settings on the two benchmarks to analyze the hyperparameter α, which is used to balance the distinct logits from ML-PGM and DCCM. Keeping other hyperparameters and settings unchanged, we set the balance coefficient α to 0.25, 0.50, 0.75, and 1.00, and report their respective results. The results in Table A4 show that the model achieves the best performance when α is set to 0.5. Furthermore, the model is not sensitive to the settings of α within the range from 0.25 to 1.0, which indicates the robustness of the model to different settings of the balancing hyperparameter, demonstrating the effectiveness of the proposed method.

E.3. Analysis of hyperparameter μ_1

In Table A5, we quantitatively analyze the hyperparameter μ_1, which is used to scale the visual logits inferred based on the visual clue in the proposed DCCM. In detail, we conduct experiments and report the performances under Exo2Ego and Ego2Exo settings on the two benchmarks, where the hyperparameter μ_1 is set to 0.25, 0.5, 0.75, and 1.0, respectively, and other hyperparameters remain unchanged. The results show that the model exhibits robustness to the change of μ_1, i.e., the metrics fluctuate within 1.5%. Finally, we set μ_1 to 1.0, in which case the model yields optimal performance under both view adaptation settings.

E.4. Analysis of hyperparameter μ_2

In Table A6, we set the hyperparameter μ_2 for scaling the textual logits to 0.25, 0.5, 0.75, and 1.0, and conduct experiments to analyze its effects comprehensively. The experimental results show that when the hyperparameter μ_2 is set to 0.5, the model achieves the best performance under all settings on *EgoMe-anti*. For the *EgoExoLearn* benchmark, the model yields the best metrics under the Ego2Exo setting and second-place results under the Exo2Ego setting. Therefore, we finally assign the hyperparameter μ_2 to 0.5 in our model. In addition, the final performance remains stable when the hyperparameter μ_2 is changed from 0.25 to 1.0, which demonstrates the robustness of the proposed DCPGN to different μ_2 values.

E.5. Analysis of the number of test samples N_{test}

The number of test samples N_{test} is crucial for the memory banks in the Multi-Label Prototype Growing Module (ML-PGM) to create reliable and meaningful prototypes. Therefore, it is necessary to ablate the number of samples during the test-time adaptation process. In Table A7, we conduct

Table A4. Analysis of the hyperparameter α for balancing distinct logits inferred by ML-PGM and DCCM.

α	*EgoMe-anti*				*EgoExoLearn*			
	Exo2Ego		Ego2Exo		Exo2Ego		Ego2Exo	
	Noun	Verb	Noun	Verb	Noun	Verb	Noun	Verb
0.25	78.61	42.64	71.85	39.21	44.44	42.89	47.70	46.22
0.50	79.03	**43.84**	**72.01**	**40.10**	**46.26**	**42.98**	**48.48**	**46.51**
0.75	**79.19**	41.62	71.09	37.81	45.39	42.62	48.43	46.31
1.00	78.17	40.89	71.37	38.31	45.37	41.98	48.13	46.11

Table A5. Analysis of the hyperparameter μ_1 for scaling the visual logits in the DCCM.

μ_1	*EgoMe-anti*				*EgoExoLearn*			
	Exo2Ego		Ego2Exo		Exo2Ego		Ego2Exo	
	Noun	Verb	Noun	Verb	Noun	Verb	Noun	Verb
0.25	77.52	43.16	71.41	38.90	45.36	42.82	48.10	46.26
0.50	78.57	42.62	71.67	39.62	45.53	42.97	48.08	45.89
0.75	78.47	42.37	71.82	39.40	45.59	42.93	48.38	45.66
1.00	**79.03**	**43.84**	**72.01**	**40.10**	**46.26**	**42.98**	**48.48**	**46.51**

Table A6. Analysis of the hyperparameter μ_2 for scaling the textual logits in the DCCM.

μ_2	*EgoMe-anti*				*EgoExoLearn*			
	Exo2Ego		Ego2Exo		Exo2Ego		Ego2Exo	
	Noun	Verb	Noun	Verb	Noun	Verb	Noun	Verb
0.25	78.48	42.98	71.90	39.37	43.99	42.62	47.90	46.14
0.50	**79.03**	**43.84**	**72.01**	**40.10**	46.26	42.98	**48.48**	**46.51**
0.75	78.86	42.76	71.87	39.05	**46.27**	42.78	48.34	46.14
1.00	77.99	42.39	70.60	39.21	44.33	**43.01**	48.39	45.68

ablation experiments of the number of test samples N_{test} on the *EgoMe-anti* benchmark with a fixed random seed, which show that reliable prototypes can be created when using only 25% samples of the full test set (972 Ego or 896 Exo samples), with only a slight performance drop compared to using the full test set. It further demonstrates the effectiveness of the proposed ML-PGM, which can achieve promising performance even with a small test set.

Table A7. Ablation on the number of test samples on *EgoMe-anti* benchmark (with 3889 Ego and 3584 Exo test samples in total).

N_{test}	Exo2Ego		Ego2Exo	
	Noun	Verb	Noun	Verb
Full	**79.03**	**43.84**	**72.01**	**40.10**
50%	78.59	42.76	71.38	39.28
25%	78.41	42.38	71.01	38.76
10%	75.42	38.32	64.84	35.75
5%	72.84	36.08	63.40	33.51

F. More Visualization Results

In this section, we show more visualization results, such as the t-SNE visualization results and the noun/verb anticipation examples to verify the effectiveness of our method.

F.1. t-SNE visualizations

We conduct more experiments by means of t-SNE [8] to visualize representations in the memory banks and the corresponding prototypes under all settings on the *EgoExoLearn* and *EgoMe-anti* benchmarks in Fig. A1. Practically, we visualize five dominant classes with the most samples following the settings in Section 4.4.3 for ease of analysis.

We visualize representations and prototypes of the model that only assigns a single class, the model without confidence-based reweighting, and the final DCPGN model. In the first column, the visualization results show that some classes are assigned only a small number of representations,

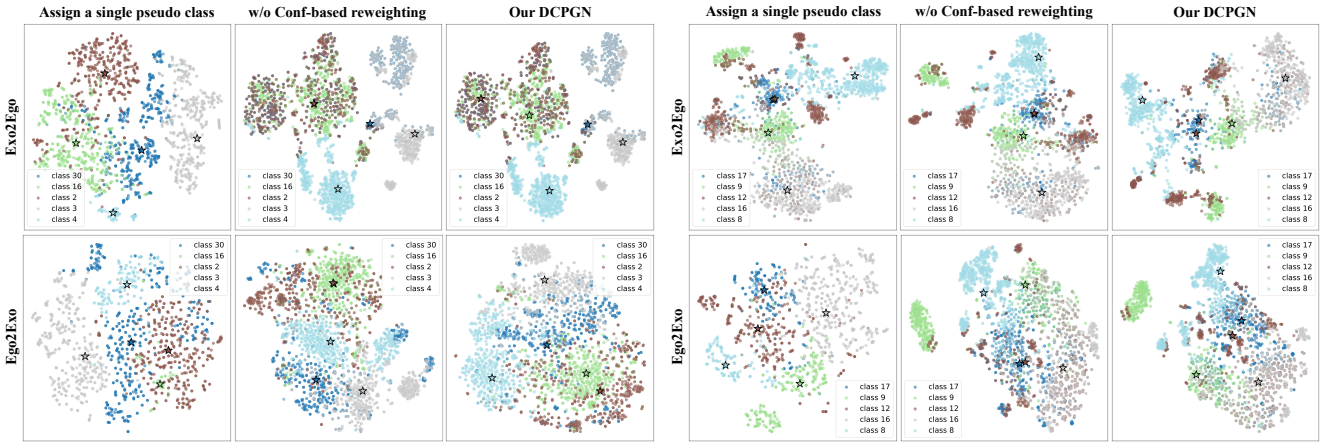

Figure A1. t-SNE visualization of representations in the memory banks and the corresponding prototypes of five dominant classes under Exo2Ego and Ego2Exo settings. The left panel shows the results on the *EgoExoLearn* benchmark and the right panel shows the results on the *EgoMe-anti* benchmark. Dots denote class-wise representations and stars denote prototypes (best viewed in color).

and prototypes of different classes are highly overlapping in some cases. It further verifies that the single-label assignment strategy may lead to remarkable inter-class imbalance and unreliable prototypes, which cause performance degradation. The results of the model without confidence-based reweighting are presented in the second column, which show that distinct classes form balanced representations and can be well distinguished. Note that it is normal for overlapping representation clusters of different classes, because each selected representation sample is assigned to multiple pseudo labels. However, due to the interference of the potential negative samples and highly overlapping representations across many classes, some prototypes are extremely close, which impairs the model's class-wise discriminative capability. Finally, in the third column, we visualize the representations and prototypes of our DCPGN. The results show that the representations are balanced and prototypes can be explicitly distinguished, which demonstrates the effectiveness of the multi-label assignment and confidence-based reweighting strategies in the ML-PGM of the proposed DCPGN.

F.2. Noun/verb anticipation visualizations

In this section, we conduct more experiments to visualize the noun and verb anticipation results of the model with or without DCCM in Fig. A2 and Fig. A3, respectively. We visualize the *Top-5* noun/verb anticipation candidates and the corresponding visual and textual clues under the Exo2Ego and Ego2Exo settings on *EgoExoLearn* and *EgoMe-anti*.

F.2.1. Noun anticipation results

In Fig. A2, we comprehensively visualize the results of the anticipated noun candidates and the corresponding visual and textual clues in the proposed DCCM. The experimental results show that visual and textual clues can effectively convey information about nouns in the upcoming actions under both Exo2Ego and Ego2Exo settings on the two benchmarks. In the example under the Exo2Ego setting on the *EgoExoLearn* benchmark (left panel), the model without DCCM cannot anticipate the active noun "pot" in the future. However, the visual clue in DCCM clearly contains the "pot", which facilitates the subsequent noun anticipation. For the Ego2Exo example on *EgoExoLearn*, due to the visual clue clearly presenting "meat" in the pot and the textual clue comprising the word "meat", the model with DCCM can precisely anticipate the "meat/beef" class, which is active in the following stir-frying process. For the examples on the *EgoMe-anti* benchmark (right panel), in the upper Exo2Exo example, the visual and textual clues of DCCM both present the "book" semantic, and successfully anticipate the active "book/notebook" class. Similarly, in the lower Ego2Exo example, the model without DCCM fails to anticipate the "hanger" class, which is an important object in the cloth-picking process. Nevertheless, the DCCM provides clues of "hanger" in both visual and textual modalities and accurately anticipates the corresponding noun class. The above visualization examples demonstrate that the proposed DCCM can explicitly mitigate the spatial gap between the Ego and Exo perspectives, thereby facilitating accurate noun anticipation.

F.2.2. Verb anticipation results

In Fig. A3, we conduct more experiments to visualize the anticipated verbs as well as the visual and textual clues under both Exo2Ego and Ego2Exo settings on the *EgoExoLearn* (left panel) and *EgoMe-anti* (right panel) benchmarks. Because it is difficult for single-frame visual clues to reflect the temporal activity progressions, textual clues become more essential in the verb anticipation task. In the example under the Exo2Ego setting on *EgoExoLearn*,

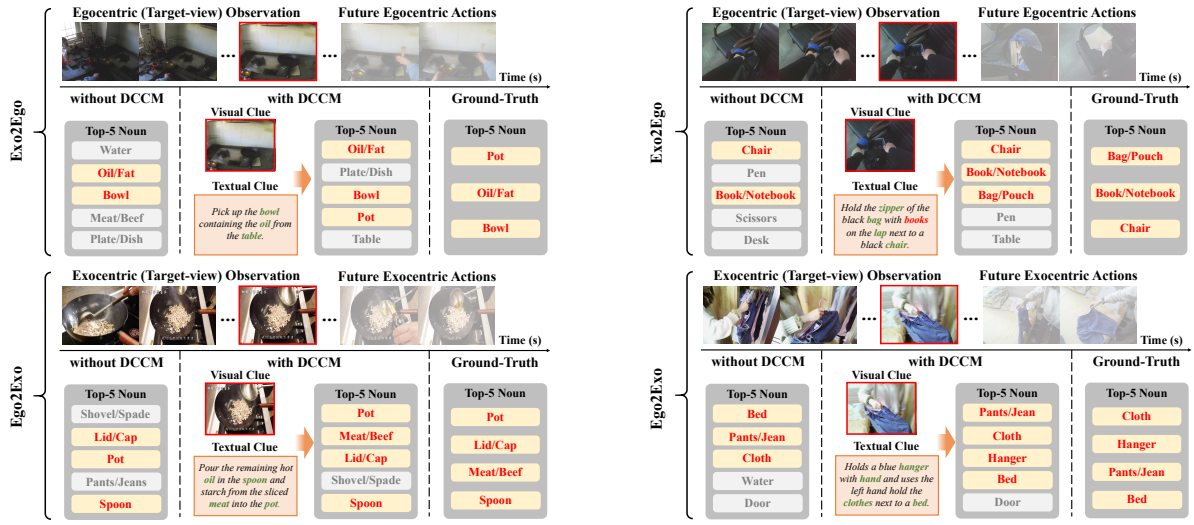

Figure A2. Visualization of the *Top-5* candidates for **noun anticipation** with or without the proposed DCCM under the Exo2Ego and Ego2Exo setting. The left panel shows the results on the *EgoExoLearn* benchmark, while the right panel shows the results on *EgoMe-anti*.

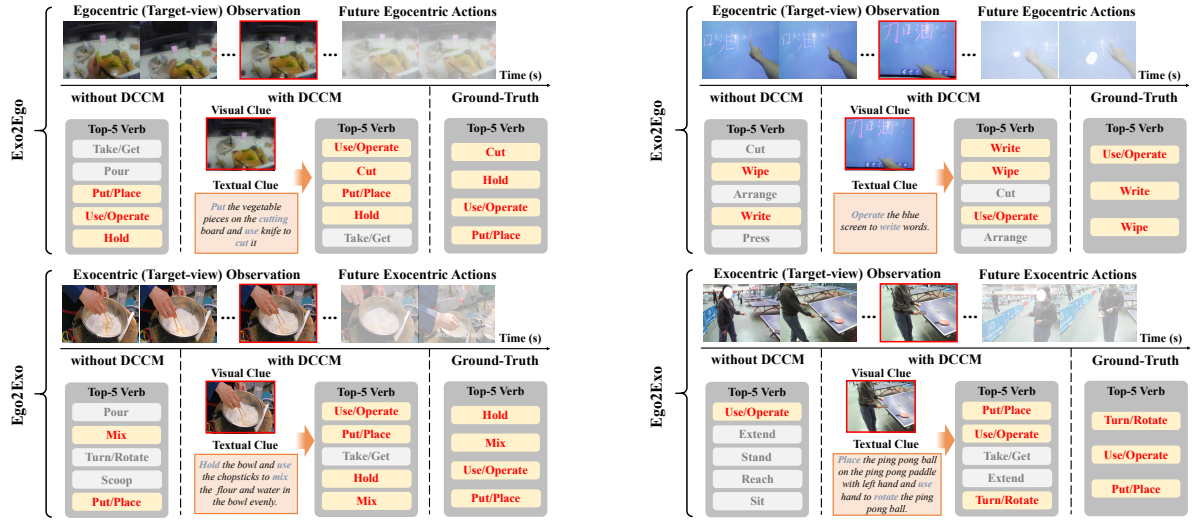

Figure A3. Visualization of the *Top-5* candidates for **verb anticipation** with or without the proposed DCCM under the Exo2Ego and Ego2Exo setting. The left panel shows the results on the *EgoExoLearn* benchmark, while the right panel shows the results on *EgoMe-anti*.

the model without DCCM fails to anticipate the verb "cut", whereas the textual clues in the DCCM clearly describe the process of cutting the vegetable, which leads to accurate anticipation of this verb class. In the Ego2Exo example on *EgoExoLearn*, the textual clue includes the verbs "hold" and "use", which facilitate more accurate anticipated verb candidates compared to the model without DCCM. The right panel presents examples of the *EgoMe-anti* benchmark. Specifically, the upper Exo2Ego example shows the subject operating the screen to write and wipe words, and the textual clue contains words such as "operate" and "write", which are consistent with classes of the upcoming actions. Finally, the lower Ego2Exo example shows that a girl will place the ball in her hand and then use the paddle to turn the ball. The results demonstrate that our model with DCCM yields more accurate verb anticipation candidates than those output by the model without DCCM. The above cases further demonstrate the effectiveness of DCCM in alleviating the temporal gap between distinct views, leading to more accurate anticipation for verb classes of the upcoming fine-level actions.

References

[1] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130(1):33–55, 2022. 3

[2] Matteo Farina, Gianni Franchi, Giovanni Iacca, Massimiliano Mancini, and Elisa Ricci. Frustratingly easy test-time adaptation of vision-language models. *Advances in Neural Information Processing Systems*, 37: 129062–129093, 2024. 1, 4

[3] Yifei Huang, Guo Chen, Jilan Xu, Mingfang Zhang, Lijin Yang, Baoqi Pei, Hongjie Zhang, Lu Dong, Yali Wang, Limin Wang, et al. Egoexolearn: A dataset for bridging asynchronous ego-and exo-centric view of procedural activities in real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22072–22086, 2024. 1, 2, 3

[4] Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14162–14171, 2024. 1, 4

[5] Wei Lin, Muhammad Jehanzeb Mirza, Mateusz Kozinski, Horst Possegger, Hilde Kuehne, and Horst Bischof. Video test-time adaptation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22952–22961, 2023. 1, 4

[6] Heqian Qiu, Zhaofeng Shi, Lanxiao Wang, Huiyu Xiong, Xiang Li, and Hongliang Li. Egome: Follow me via egocentric view in real world. *arXiv preprint arXiv:2501.19061*, 2025. 2, 3

[7] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35:14274–14289, 2022. 1, 4

[8] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 5

[9] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020. 1, 4

[10] Zixin Wang, Dong Gong, Sen Wang, Zi Huang, and Yadan Luo. Is less more? exploring token condensation as training-free test-time adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 144–154, 2025. 1, 4

[11] Xiangyu Wu, Feng Yu, Qing-Guo Chen, Yang Yang, and Jianfeng Lu. Multi-label test-time adaptation with bound entropy minimization. *arXiv preprint arXiv:2502.03777*, 2025. 2, 3, 4