PLOT-TAL - Prompt Learning with Optimal Transport for Few-Shot Temporal Action Localization - Supplementary Material

Edward Fish, Jon Weinbren, and Andrew Gilbert

University of Surrey, Guildford, UK

1 Introduction

We introduce several additional ablation experiments, provide implementation details, and provide qualitative results that could not be included in the paper due to space limitations.

2 Implementation Details

2.1 Feature Extraction

Features are extracted from a pre-trained I3D network [6] trained on the Kinetics-600 dataset [2,**?**] in a supervised setting. We extract the optical flow and RGB output embeddings, which are then concatenated to form a $2048 \times T$ embedding, where T is the total number of video segments. Each video segment refers to 16 frames sampled at 30 FPS with a stride of 4 frames. This is the standard feature extraction pipeline used in all previous TAL works [4,**?**]. To deal with variable frame lengths T, we pad all samples to $T = 2048$, which accounts for the length of all videos. During training, we include a mask to represent the zero-padded regions and apply the mask after each operation.

2.2 Training

We train each model for 100 epochs, except for when we increase the number of shots above 15, in which case we train for 200. We randomly initialize the *ctx* embedding vectors and append them to the start of the prompt. All models are trained with a batch size of 2 on a single NVIDIA RTX 3090 24GB GPU. The memory required for training the model on THUMOS'14 with a batch size of 2 and when $N = 4$ is 5GB. We include a summary of the method in alg:fstal.

Algorithm 1 Overview of TAL-PLOT method

Input: Untrimmed input video $\mathcal{V}$
Output: Action instances $\mathcal{Y} = \{y_1, y_2, \dots, y_N\}$
Feature Extraction and Representation:
for $t = 1$ to T **do**
  Extract feature vector $x_t = f_{\text{CNN}}(v_t)$ using a 3D CNN
  Refine features $x'_t = f_{\text{conv}}(x_t)$ with a 1D convolutional layer
end for
Adaptive Prompt Learning:
for each action category k **do**
  Generate N prompts $\mathcal{P}_k = \{P_{k1}, P_{k2}, \dots, P_{kN}\}$ using f_{CLIP}
end for
Optimal Transport with Sinkhorn Algorithm:
for each action category k **do**
  Align features $\{x'_1, \dots, x'_T\}$ with prompts $\mathcal{P}_k$ using OT
end for
Temporal Pyramid and Feature Integration:
Construct temporal feature pyramid X'_l with max-pooling
Multi-Resolution Temporal Alignment:
for $l = 1$ to L **do**
  Align features at level l of the pyramid with $\mathcal{P}_k$
end for
Decoder Architecture:
Use aligned features to predict action labels Ψ and boundaries O_l
Learning Objective:
Minimize total loss L_{total} with Focal Loss and DIoU Loss
return $\mathcal{Y}$

2.3 Optimal Transport

As discussed in the main paper. The optimal transport is optimized in a two-stage process as proposed in [1] where we find the transport cost between the video features and prompts in the inner loop. After converging the Sinkhorn algorithm, we use the backward pass to update the learnable prompts. For the parameters, we follow the setup in [1] where $\delta = 0.01$, $\lambda = 0.1$, and we perform 100 iterations within the inner loop. We generate results over 4 random seeds and report the average. Further details are provided in alg:detailedOTsinkhorn.

3 Ablation Experiments

3.1 Number of Learnable Context Tokens

We state in the paper that each prompt has several learnable context tokens as described in [5] and [3]. These context tokens are randomly initialized so that for the class 'Basketball Dunk' with 4 *ctx* tokens, the full prompt will be

$$P = \{X, X, X, X, \text{Basketball Dunk}\} \tag{1}$$

Algorithm 2 Optimal Transport Sinkhorn Algorithm for Few-Shot TAL

Input: Untrimmed input video $\mathcal{V}$, pretrained model features f_{CNN}, number of prompts N, entropy parameter λ, maximum number of iterations T_{in}, T_{out}
Output: Optimized prompt parameters $\{\omega_n\}_{n=1}^N$

```
Initialize prompt parameters {ω_n}_{n=1}^N
for t_out = 1 to T_out do
    Obtain a visual feature set F ∈ ℝ^{M×C} with the visual encoder f_CNN(x_t)
    Generate prompt feature set G_k ∈ ℝ^{N×C} for each class with textual encoder
    g(label_k, ctx_{k1}, ..., ctx_{kn_ctx})
    Calculate the cost matrix C_k = 1 − F^⊤ G_k for each class
    Calculate the OT distance with an inner loop:
    Initialize v^{(0)} = 1, δ = 0.1, Δv = ∞
    for t_in = 1 to T_in do
        Update u^{(t_in)} = u/(exp(−C/λ) v^{(t_in−1)})
        Update v^{(t_in)} = v/(exp(−C/λ)^⊤ u^{(t_in)})
        Update Δv = ∑ |v^{(t_in)} − v^{(t_in−1)}|/N
        if Δv < δ then
            Break
        end if
    end for
    Obtain optimal transport plan T_k^* = diag(u^{(t)}) exp(−C_k/λ) diag(v^{(t)})
    Calculate the OT distance d_OT(k) = ⟨T_k^*, C_k⟩
    Calculate the classification probability p_OT(y = k|x) with the OT distance
    Update the parameters of prompts {ω_n}_{n=1}^N with cross-entropy loss L_CE
end for
return Optimized prompt parameters {ω_n}_{n=1}^N
```

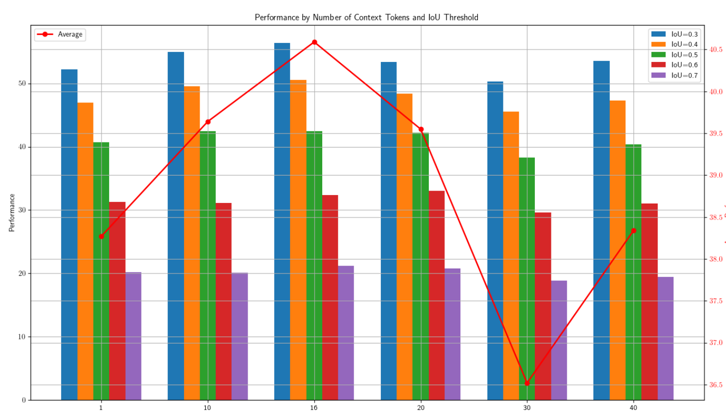

Fig. 1. mAP over various IoU thresholds for the THUMOS' 14 dataset with variable number of additional context tokens appended to each N prompt.

In fig:iou and tab:performance$_{s}cores, weshowtheeffectofvaryingthenumberoflearnablectxtokensappende$
tokens are randomly initialized. The figure shows that the optimum number of tokens is between 10 and 20. As per the existing literature [5,**?**], we select 16 tokens for all methods unless otherwise stated and train and test using the 5-shot, 20-way setup as described in the paper.

Table 1. Ablation experiment on the number of context tokens on the THUMOS'14 Dataset.

Ctx Tokens	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**avg**
1	52.25	46.94	40.73	31.26	20.17	38.27
10	54.94	49.55	**42.49**	31.14	20.08	39.64
16	**56.42**	**50.54**	42.48	32.35	**21.17**	**40.59**
20	53.39	48.38	42.19	**33.00**	20.78	39.55
30	50.27	45.54	38.30	29.64	18.83	36.52
40	53.55	47.30	40.35	31.06	19.46	38.34

3.2 Visual Feature Embeddings

To evaluate the effectiveness of adding motion information via optical flow, we also performed additional experiments using only the RGB embeddings, the optical flow embeddings, and RGB CLIP embeddings from a ViT-B-16 encoder, with results shown in tab:embedding$_{r}esults.TheresultsshowthattheCLIPembeddingsperformbetterthantheRGBem$
2.67. This is because of the implicit alignment between the image and text encoder embeddings before temporal convolution. However, when combined with optical flow, the performance is improved by a large margin of $\uparrow$ 7.56, demonstrating the improved classification ability of the network when we add additional temporal information via optical flow.

Table 2. Comparison of mAP scores for various visual input embeddings on the THUMOS'14 dataset.

Embeddings	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**avg (mAP)**
CLIP	46.99	42.09	34.26	25.34	15.82	32.90
RGB	43.13	38.76	31.71	23.15	14.46	30.24
Optical Flow	26.03	23.10	19.54	14.07	8.93	18.33
RGB + Flow	**55.88**	**50.21**	**43.06**	**31.97**	**21.16**	**40.46**

4 Visualisation Results

In fig:t$_{p}lot, weshowthenormalizedtransportcostforeachframeand$N$embeddingfortheclasslabel`CricketShot'$
or Prompt 1 learns global information across all frames. This shows how, in a

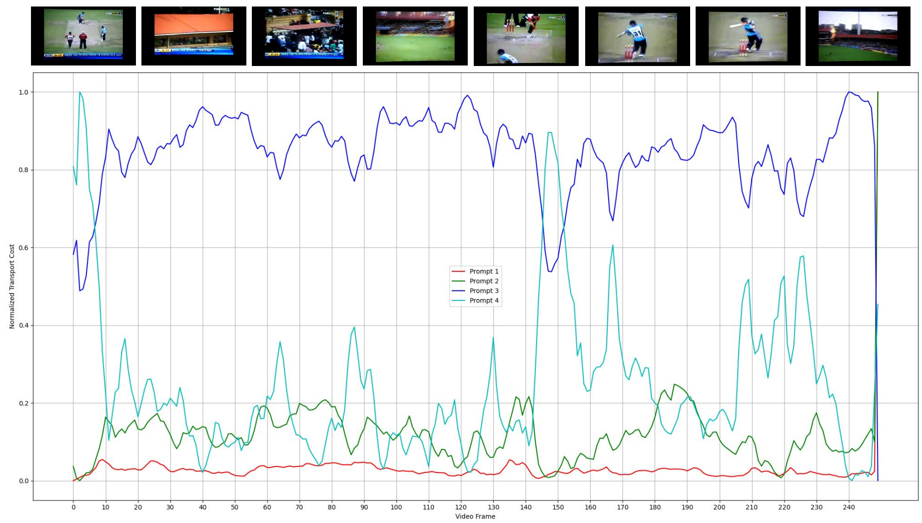

Fig. 2. The normalized transport cost of each N prompt for the class ‘Cricket Shot’ after training. Prompt one aligns with global information, while the other prompts learn additional, complementary views. In the transport cost algorithm, a lower value indicates closer alignment.

single prompt framework, we may distribute alignment across all frames and lose disciminative ability, since it learns global information over the whole video. In the figure, we can note that Prompt 4 appears to learn background information and is more closely aligned to frames where we can see the stadium stands. Prompts 2 and 3, however, indicate a closer alignment with objects related to the class of ‘cricket shot,’ including when the cricket strip is in the shot and there are people on the field.

5 Prompt Engineering

We demonstrate how including crafted prompts can help to boost performance. In tab:sports$_a ctions, we show the prompts generated by GPT 3.5 with the prompt -$
‘Generate prompts for a temporal action localization task for the following class IDs. The prompts should incl

References

1. Chen, G., Yao, W., Song, X., Li, X., Rao, Y., Zhang, K.: Prompt learning with optimal transport for vision-language models. arXiv preprint arXiv:2210.01253 (2022)
2. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
3. Weng, Y., Pan, Z., Han, M., Chang, X., Zhuang, B.: An efficient spatio-temporal pyramid transformer for action detection. In: ECCV (2022)

Table 3. GPT generated descriptions for PLOT-TAL Verbose on THUMOS'14 Dataset.

ID	Description
7	The precise moment a baseball player winds up and releases the ball towards the batter
9	The instant a basketball player leaps into the air to forcefully slam the ball through the hoop
12	The exact moment the cue stick strikes the cue ball, initiating the billiards shot
21	The moment a weightlifter hoists the barbell from the ground to overhead in one fluid motion
22	The split second a diver leaps off the cliff edge, beginning their descent into the water below
23	The moment a cricket bowler releases the ball towards the batsman with a swift arm motion
24	The precise moment the batsman swings the bat to strike the cricket ball
26	The instant a diver jumps off the board, tucking and twisting before plunging into the pool
31	The moment a frisbee is caught by a leaping player, securing it firmly in their hands
33	The exact moment a golfer swings the club, making contact with the ball to send it flying
36	The moment an athlete spins and releases the hammer, propelling it into the air
40	The split second an athlete takes off over the high jump bar, attempting to clear it without touching
45	The precise moment the javelin is thrown, with the athlete's arm extending forward in a powerful motion
51	The instant an athlete sprints and leaps into the air to cover the maximum distance before landing in the sand pit
68	The moment an athlete plants the pole in the box and vaults over the bar, pushing themselves upwards
79	The exact moment the shot is put from the neck, using one hand, in a pushing motion through the air
85	The moment a soccer player strikes the ball with their foot aiming to score a penalty kick
92	The precise moment a tennis player swings their racket to strike the incoming ball
93	The instant an athlete spins and releases the discus, hurling it into the designated sector
97	The moment a volleyball player jumps and forcefully spikes the ball over the net towards the opponent's court

4. Zhang, C.L., Wu, J., Li, Y.: Actionformer: Localizing moments of actions with transformers. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV. pp. 492–510. Springer (2022)
5. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
6. Zisserman, A., Carreira, J., Simonyan, K., Kay, W., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)