Appendix
– Ego-VPA: Egocentric Video Understanding with Parameter-efficient Adaptation

Tz-Ying Wu[1,2] Kyle Min[1] Subarna Tripathi[1] Nuno Vasconcelos[2]
[1]Intel Labs [2]UC San Diego
{tz-ying.wu,kyle.min,subarna.tripathi}@intel.com nvasconcelos@ucsd.edu

Appendix

The appendix is organized as follows. Section A summarizes the implementation details for the model architecture and training. Section B provides the parameter analysis of the proposed prompt synthesis compared to the CMM [2] as mentioned in section 5.1. Section C and section D provide more ablations to complement the ablations presented in the main paper, and section E presents the basis prompt visualization.

A. Implementation Details

Model architecture. The transformer architectures are inherited from LaViLa [5], the Ego-VFM we adopted. The text encoder ϕ_{txt} is a 12-layer ($L = 12$) Transformer with $d_{txt} = 512$ and context length as 77. We prompt the text encoder with $M_v = 8$ for all methods. The video encoder ϕ_{vid} is a 12-layer ($L = 12$) TimeSformer [1], where each video frame is decomposed into 14×14 patches ($N_p = 196$) and $d_{vid} = 768$. All the models are trained with 16 frames per video ($T = 16$) unless explicitly noted (e.g. Table 4). We prompt our method and the baselines with the same amount of prompts to ensure fair comparisons. For VPT and VoP, since the visual prompts are not frame-specific, we prompt the video encoder with $M_v = 128$ prompts. For VoP^{F+C} and Ego-VPA, we prompt each frame with $M_v = 8$ frame-specific prompts so that the total number of prompts is $M_v T = 8 \times 16 = 128$. Note that we only prompt for the spatial attention blocks in the TimeSformer, as mentioned in section 4, and the dimension of all the prompts is equal to the feature dimension of the encoder. For the prompt basis $\mathcal{F}$, we set $d_f = 512$ and $B = 10$ as ablated in Figure 8 and Figure 5a. These prompts are randomly initialized. We set the intra/inter-frame attention boundary (i.e. K) as 8 for both VoP^{F+C} and Ego-VPA as it produces the best results (See Figure 5c).

Training. We implement all the codes with PyTorch atop the codebase of LaViLa [5]. As mentioned in section 6.1, all the experiments are trained with 8 NVIDIA Titan Xp GPUs with a batch size of 4 per GPU. According to our preliminary experiments in section D, we train all the prompt-tuning baselines and Ego-VPA with a learning rate of 0.01 for 10 epochs with AdamW optimizer and the cosine learning rate scheduler implemented in [5]. For fine-tuning, such a learning rate will be too high and make the training noisy. Thus, we also optimize the learning rate for fine-tuning and set learning rates as 1e-5/1e-4 for Charades-Ego and EGTEA, respectively. For Ego-VPA, we set λ as 0.05 in all experiments, and adopt the γ schedule in Figure 7.

Table 5. Ablations on hyper-parameter λ.

λ	0.01	0.05	0.1	1
mAP	32.9	**33.8**	33.2	33.5

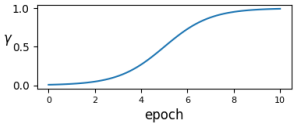


Figure 7. Schedule of γ.

B. Parameter Analysis

We compare the number of learnable parameters for video prompt tuning using CMM [2] and the proposed prompt synthesis (PS) (detailed in section 5.1) in the *intra-frame* attention layers respectively. Note that we ignore the common parts of these methods in this comparison as the number of trainable parameters in these parts is the same for both methods.

The CMM module [2] uses a 1-layer bi-directional LSTM (Bi-LSTM) to model context information across frames, and generates M_v frame-specific prompts per frame by mapping the LSTM output with a linear projection layer. The Bi-LSTM with hidden size equal to input size (i.e. d_{vid}) requires $16{d_{vid}}^2$ model parameters, and the linear projection layer requires $2d_{vid} \times M_v T d_{vid}$ parameters, resulting in $param(CMM) = (16 + 2M_v T) d_{vid}^2$ parameters in total.

On the contrary, for the proposed PS, the learnable parameters for the adaptation are the basis prompts $\mathcal{F} = \{\mathbf{f}_1, ..., \mathbf{f}_B\}$ and projection layers $h_{vid}(;\mathbf{W}^h_{vid})$, $g_{vid}(;\mathbf{W}^g_{vid})$, where $\mathbf{f}_i \in \mathbb{R}^{d_f} \forall i$, $\mathbf{W}^h_{vid} \in \mathbb{R}^{d_f \times d_{vid}}$, and $\mathbf{W}^g_{vid} \in \mathbb{R}^{d_{vid} \times d_f}$. Hence, the total number of trainable parameters in *intra-frame* attention layers is $param(PS) = d_f(B + 2d_{vid})$.

Let $d_f = cd_{vid}$, $B = ud_{vid}$ and c, u be scalars in $(0, 1]$, then we have $param(PS) = (ud_{vid} + 2d_{vid})cd_{vid} =$

Table 6. Ablations on the loss function and prompt query method. Orthogonality Constraint: 2^{nd} term in Eq. (10).

Prompt Query	Orthogonality Constraint	mAP
random		33.2
top-k		32.8
sampling from π_m		33.3
random	✓	32.9
top-k	✓	33.5
sampling from π_m	✓	**33.8**

$(u+2)cd_{vid}^2 < param(CMM) = (16+2M_vT)d_{vid}^2$ if $(u+2)c < 16+2M_vT$, which is easily satisfied as $(u+2)c \leq 3 < 16+2M_vT$. Note that $M_vT = 128$ in our settings, where we prompt all the methods with the same number of prompts per frame to ensure fair comparison (Check section A for more details). The proposed PS appears to be a more parameter-efficient method for adaptation.

C. Additional Ablations

In this section, we provide more ablations to complement the ablations presented in the main paper. Experiments are conducted on Charades-Ego.

Hyper-parameter λ. λ is a hyper-parameter to weight between the contrastive learning loss of Eq. (2) and the cross-modal prompt synthesis loss of Eq. (11). We experimented with different λ values as shown in Table 5, and set λ as 0.05 in all the experiments as it achieves the best performance.

Schedule of γ. As introduced in section 5.3, we gradually increase γ, the weight of π_{sim}, from 0 to 1 during training, following the schedule in Figure 7. This is to prevent the case that some features are never selected to learn. In the beginning, the mixture distribution is dominated by π_{invf}, while gradually shifting to π_{sim} as γ increases to 1. We compare the adopted schedule with a baseline schedule that linearly increases the γ from 0 to 1. Results show that this baseline underperforms the proposed one (33.0 vs 33.8).

Loss function and prompt query strategy. In Table 3, we present the comparison of the proposed sampling strategy to the top-k method. Table 6 provides the complete table including comparisons to naive random sampling and top-k query along with the effect of the orthognality constraint (i.e. second loss term in Eq. (10)). Sampling with π_m surpasses the other feature query methods either with or without the orthognality constraint, while adding the constraint generally improves the performance since it encourages the sparsity of features in the prompt basis. Overall, the proposed loss and prompt query strategy performs the best over other variants.

Latent feature dimension d_f. Figure 8 ablates the dimension of the latent feature space $\mathcal{H}$ the text/video frame features mapped to. Note that the largest d_f is $min(d_{txt},d_{vid}) = 512$, i.e. $log_2 d_f = 9$. As d_f increases, the compression ratio for the frame feature $\mathbf{z}_f \in \mathbb{R}^{d_{vid}}$ or the text feature $\mathbf{z}_t \in \mathbb{R}^{d_{txt}}$ is lower, and thus it can be better represented by the basis prompts and reach better performance.

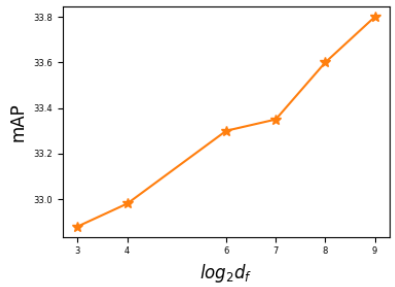


Figure 8. Ablations on latent feature dimension d_f (shown in log-scale).

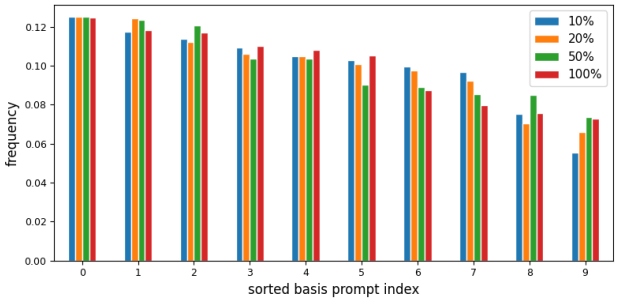


Figure 9. Basis prompt selection frequency with different amounts of training data.

Basis prompt selection frequency. Figure 9 compares the frequency that each basis prompt in $\mathcal{F}$ is selected when different amounts of data are used for training. We notice that compared to low-data regimes (e.g. using only 10% data), using all the training data tends to make the frequency distribution more balanced and lead to lower reconstruction errors (e.g. 10%/100%: 1.36/0.62), which means the basis prompts are better estimated. This is shown to achieve higher classification performance as in Figure 5d.

D. Preliminary Experiments on a Charades-Ego Subset

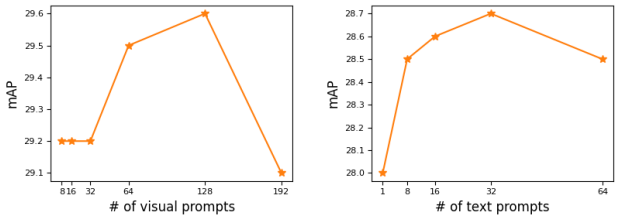
Since there is no prior work on prompt-tuning for TimeSformer-based video foundation models (VFMs), we first created a smaller subset of Charades-Ego [4] that contains 25-shot instances per class to quickly iterate over different design choices and understand the behavior of different components. We provide these preliminary experiments to supplement our main results and as a reference for future research.

D.1. Ablation Studies

Prompt-tuning with different attention blocks. The video encoder (i.e. TimeSformer [1]) contains two types of

Table 7. Ablations on prompt tuning with different attention blocks in TimeSformer [1].

Attention block	mAP
Spatial-only	**29.6**
Temporal-only	24.0
Both	28.8



(a) (b)

Figure 10. Ablations on the number of prompts for (a) VPT and (b) TPT, respectively.

Table 8. Ablations on prompt tuning learning rates.

Learning rate	0.001	0.01	0.1
mAP	29.9	**30.3**	29.9

Table 9. Comparisons to state-of-the-art prompt-tuning methods on 25-shot Charades-Ego subset.

Method	Trainable Params (%)	mAP
Zero-shot	0%	26.8
Full fine-tuning	100%	30.6
TPT [6]	0.002%	28.5
VPT [3]	0.66%	29.6
VoP [2]	0.67%	30.3
VoP^{F+C} [2]	10.86%	30.8
Ego-VPA (ours)	0.84%	**31.4**

attention blocks, spatial attention and temporal attention, as introduced in section 3.1. We started with the vanilla video prompt tuning (VPT) approach as described in section 4 with these two attention blocks. Table 7 compares the results of prompt-tuning with different blocks, and we can see that prompt-tuning only with the spatial attention block leads to the best performance. As a result, we only prompt-tune for the spatial attention block in the rest experiments.

Number of prompts. After deciding on the prompting strategy for the video part. We explored the effect of the number of prompts on the model performance. Figure 10a and Figure 10b summarize the results of this ablation on VPT and TPT respectively. We see similar behaviors when increasing the number of prompts in both modalities, where the performance increases with the number of prompts in the beginning, and then it starts to saturate. By comparing the results of VPT and TPT, we can also observe that the domain gap between the pretrained dataset and the downstream dataset is larger in the visual part than in the text part, as VPT leads to better performance. However, both of them only prompt-tune a single modality and can only reach limited gains due to the limited learning capacity, which justifies the need of prompt-tuning for both modalities.

Learning rates. We further ablate the learning rates with VoP that prompt-tunes for both encoders (i.e. VPT+TPT). Since prompts are learnable parameters added in the input space, it requires a larger learning rate to propagate the gradients to these parameters. We experimented with learning rates in $\{0.001, 0.01, 0.1\}$ as shown in Table 8, and chose the best-performing one (i.e. 0.01) for the rest experiments.

D.2. Main results

Table 9 presents the comparison of Ego-VPA to state-of-the-art prompt tuning methods on this 25-shot subset. Cross-referencing to the results in Table 2, we can see that the model performance of these methods holds similar trends. VoP improves over single-modal prompting methods, VPT and TPT. By introducing the context modeling module and frame-aware attention layers, VoP^{F+C} reaches further gain. However, they all underperform the proposed Ego-VPA, which only uses 0.84% trainable parameters. This shows that Ego-VPA is a more efficient and effective solution.

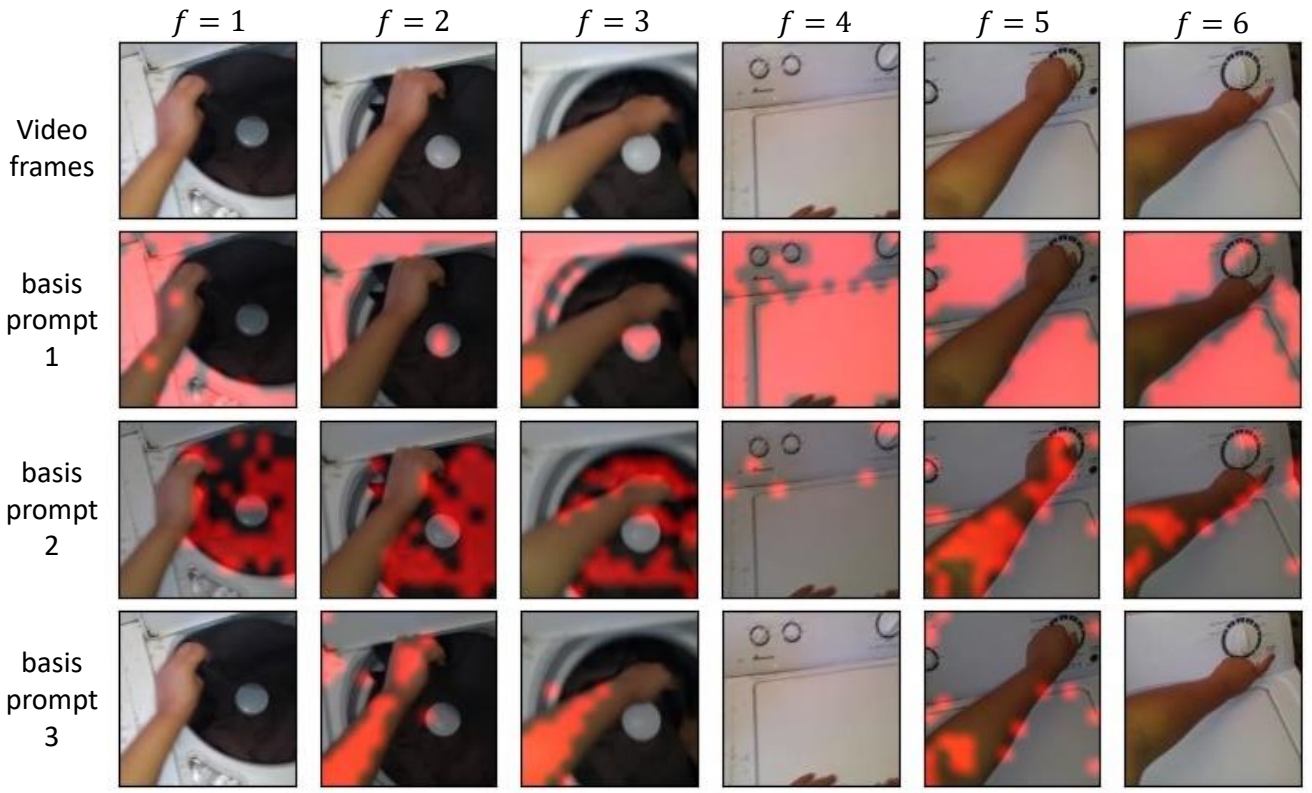
E. Visual Examples

To get more intuition of the prompts in the prompt basis $\mathcal{F}$, we further sample some basis prompts $\mathbf{f}_i$ from $\mathcal{F}$ and visualize the attention maps between these basis prompts and the video frames. Figure 11 presents two examples in Charades-Ego [4]. The red parts represent the areas with high attention scores. Note that some video frames do not contain red areas since the basis prompt is not queried by those frames. We can see that the attention maps of different basis prompts do not have many overlaps. For example, in the laundry example, basis prompt 1 focuses on the washing machine, while basis prompt 2 attends to the clothes and hands controlling the button of the washing machine. In the cooking example, basis prompt 1 and basis prompt 2 focus on the background scene and the cooking behavior respectively. This shows that the basis prompts are diverse and contain different meanings.

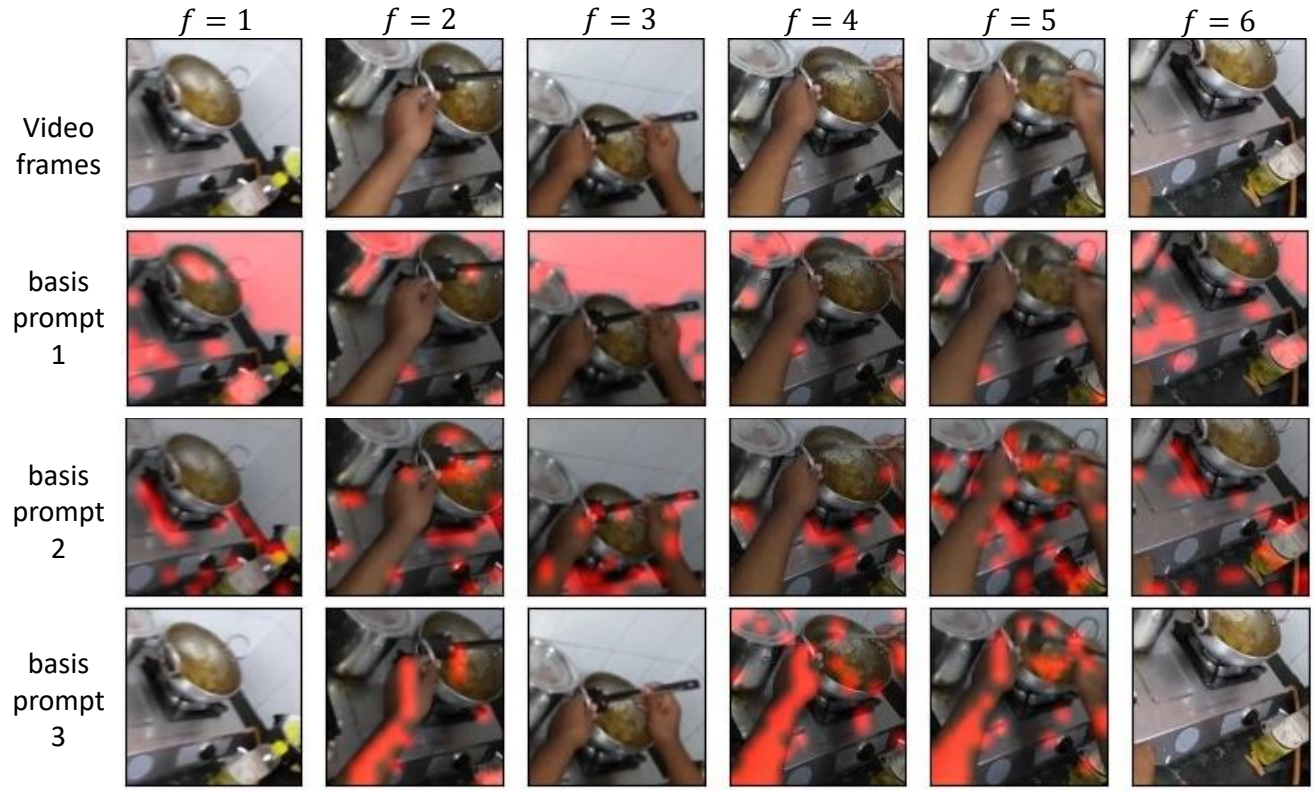
References

[1] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021. 1, 2, 3

[2] Siteng Huang, Biao Gong, Yulin Pan, Jianwen Jiang, Yiliang Lv, Yuyuan Li, and Donglin Wang. Vop: Text-video co-operative prompt tuning for cross-modal retrieval. In *CVPR*, 2023. 1, 3



(a) Washing some clothes.



(b) Cooking.

Figure 11. Visualization of prompt attention maps.

[3] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022. 3

[4] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Charades-ego: A large-scale dataset of paired third and first person videos. *arXiv preprint arXiv:1804.09626*, 2018. 2, 3

[5] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6586–6597, 2023. 1

[6] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022. 3