MixANT: Observation-dependent Memory Propagation for Stochastic Dense Action Anticipation

Supplementary Material

7. Introduction

This supplementary material provides comprehensive details regarding the implementation, ablation studies, and qualitative analysis of our proposed MixANT model. The document is structured as follows: Section 8 describes the implementation details of our model architecture; Section 9 presents additional ablation studies to analyze the contribution of different components; Section 10 provides some additional results on other anticipation tasks; and Section 11 offers further qualitative comparisons between MixANT and existing approaches in the literature.

8. Implementation Details

The overall task for the long-term dense anticipation is shown in Fig. 9. For the Breakfast and 50Salads datasets, we extract I3D features from previous works [2] and [11], while for Assembly101 we use TSM-features [21] provided by [29]. The features are then concatenated with zero padding to represent future feature frames. A Gaussian noise vector is sampled and added to the input tensor, which is then processed by the MixANT model to generate per-frame labels. Importantly, each noise vector produces a single sample, and multiple outputs are generated by repeating the process with different noise vectors while maintaining the same input sample.

The list of hyperparameters for reproducibility purposes is provided in Tab. 2. We used a single A100 GPU (80 GB) for training all of our MixANT models.

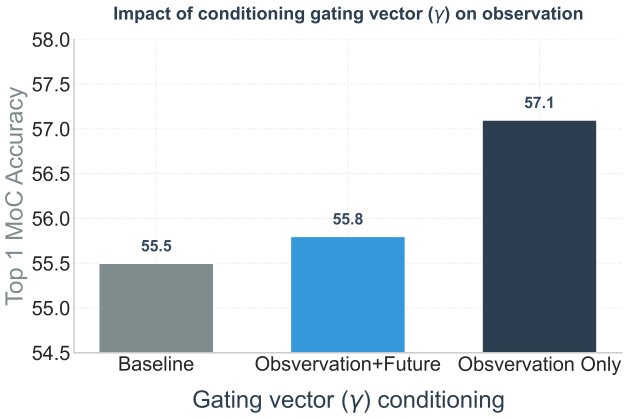

Figure 7. Impact of conditioning gating vector γ on present and future.

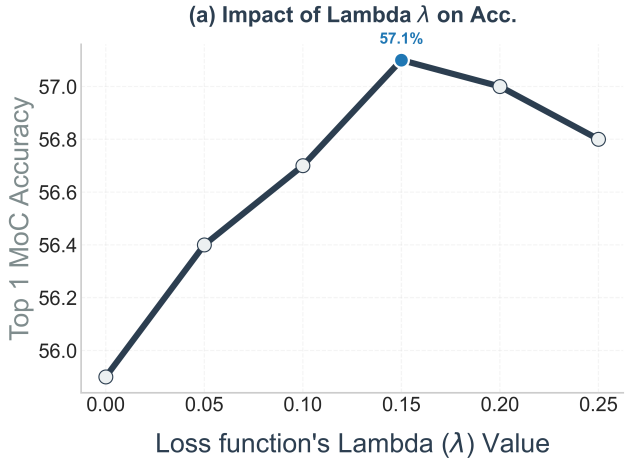

Figure 8. Impact of λ_{lb} for the load balancing loss ($\mathcal{L}_{lb}$).

9. Additional Ablation Studies

9.1. Impact of Conditioning Gating Vector on Present and Future

We conducted an ablation study examining how performance is affected when the gating vector γ is conditioned on either the observed part of the input alone or the complete input containing both observed and future parts.

For our ablation, we condition the gating vector on the observed part of the input $F^{k-1}_{t,1:P}$ and the complete input $F^{k-1}_{t,1:P+F}$ as in Eq. (17). As shown in Fig. 7, conditioning γ exclusively on observed frames yields superior performance compared to conditioning on both observation and future components. This finding is consistent with theoretical expectations, as the latter approach incorporates zero-padded future values that provide no meaningful information for the decision-making process of selecting appropriate $\mathbf{A}$ matrices. Consequently, restricting the conditioning of the gating vector to only the observation component—which contains the complete contextual information relevant to the selection process—proves to be more effective while appropriately disregarding uninformative future padding.

9.2. Impact of Contribution of Load Balancing Loss

We analyze the impact of incorporating a load-balancing loss component into the overall training loss function. This addition of load balancing loss is controlled by a coefficient λ_{lb} that determines the relative contribution of the load balancing loss. To analyze its impact, we systematically varied

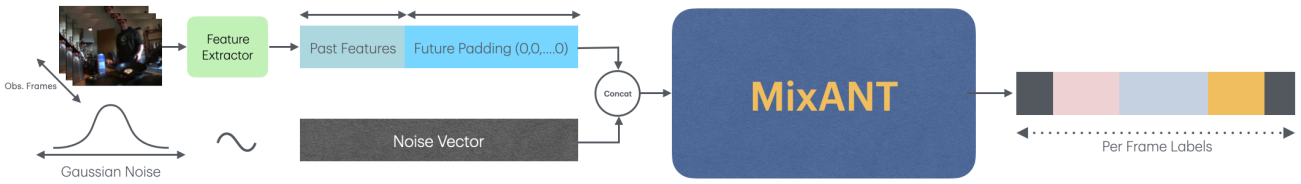

Figure 9. Overall task of stochastic long-term dense action anticipation with its inputs and outputs. For the Breakfast and 50Salads datasets, we extract I3D features from previous works [2] and [11], while for Assembly101 we use TSM-features [21] provided by [29]. The features are then concatenated with zero padding to represent future frames. A Gaussian noise vector is sampled and added to the input tensor, which is then processed by the MixANT model to generate per-frame labels. Importantly, each noise vector produces a single sample, and multiple outputs are generated by repeating the process with different noise vectors while maintaining the same input sample.

MixANT Training Recipe			
Dataset	Breakfast	50Salads	Assembly101
Epochs	90	90	90
Num of MixANT Blocks	15	15	15
Optimizer	AdamW	AdamW	AdamW
Optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.999$	$\beta_1 = 0.9, \beta_2 = 0.999$	$\beta_1 = 0.9, \beta_2 = 0.999$
Learning rate	0.0005	0.001	0.0005
Diffusion Steps (Training)	1000	1000	1000
DDIM steps	50	10	50

Table 2. Hyperparameters for MixANT.

λ_{lb} from 0 to 0.25 in increments of 0.05 in Fig. 8.

When $\lambda_{lb} = 0$, effectively training without load balancing loss, the network performs marginally better than the baseline Mamba network. However, the introduction of load balancing loss substantially improves performance. The optimal results are achieved at $\lambda_{lb} = 0.15$, beyond which performance degrades as the optimization objective becomes overly focused on load balancing at the expense of overall performance metrics.

9.3. Impact of Number of Experts on Inference Speed

We evaluate the mean time required for generating 25 samples using our model on the Breakfast dataset with α=0.3 and β=0.5, for different numbers of experts. The results are presented in Fig. 10. We observe that there is a slight increase in inference time when the first mixture is used (2 experts). After that increase, the number of experts poses a minimal overhead. We do not include GTDA in the plot because its inference speed is much higher (71.8 seconds).

Figure 10. Mean inference time for generating 25 samples using our model on the Breakfast dataset with α=0.3 and β=0.5 across varying numbers of experts. Dashed horizontal lines represent inference speed of other methods.

9.4. Computational Complexity and Performance

Tab. 3 compares parameters, memory, and inference time of various methods. We also compare our approach to alternatives to obtain input-dependent A matrices (large MLP and query-key). MixANT is much more efficient than a large MLP ($>$ 80x faster) and query-key ($>$ 14x faster), and it

Method	Mamba	$A(x)$	Param. (M)	Mem. (GB)	Inf. Time (s)	Top-1 MoC
GTDA			3.9	19.2	71.8	48.9
Transf. (15)			1.2	11.3	15.6	48.8
Transf. (18)			1.4	13.2	18.2	50.3
MANTA	✓		1.4	10.2	1.1	52.7
Large MLP	✓	✓	8.0	38.7	137.4	47.4
Query-Key	✓	✓	2.3	22.7	24.4	52.3
MixANT ($E = 5$)	✓	✓	1.6	10.9	1.7	54.1

Table 3. Comparison of computational cost and Top-1 MoC for different methods. The last three rows report results for three variants of making A input-dependent ($A(x)$). Top-1 MoC is averaged over all α and β values on Breakfast. Inference time is per video for $\alpha = 0.3$, $\beta = 0.5$, and generating 25 samples.

K_0	1	2	3	4	5	6
Params. (M)	1.70	1.67	1.64	1.62	1.60	1.58
Mem. (GB)	11.0	10.9	10.9	10.8	10.8	10.7
Inf. Time (sec)	1.8	1.8	1.7	1.7	1.6	1.6
Top-1	53.0	53.2	53.5	53.3	52.9	52.4

Table 4. Computational cost and mean inference time for generating 25 samples using our model on the Breakfast dataset with α=0.3 and β=0.5 for varying numbers of initial static blocks K_0.

achieves higher Top-1 MoC.

We also report the impact of the number of initial static blocks K_0 on parameters, memory, and inference time in Tab. 4.

10. Additional Quantitative Results

Tab. 5 presents the action anticipation results on the EK-100 dataset, comparing our proposed MixMamba approach with traditional attention and Mamba baselines. Our MixMamba method demonstrates consistent improvements across all evaluation metrics and scenarios. Specifically, MixMamba achieves 29.7% verb accuracy and 17.1% action accuracy on the overall split, representing improvements of 4.6% and 3.0% over the attention baseline, and 1.8% and 1.9% over the Mamba baseline, respectively. The improvements are particularly notable in the challenging tail scenarios, where MixMamba reaches 22.7% verb accuracy and 14.1% action accuracy, outperforming both baselines by substantial margins. These results demonstrate that our mixture approach is effective on diverse anticipation scenarios.

11. Additional Qualitative Results

We present some qualitative results for MixANT in comparison to the baseline MANTA for different action videos on the Breakfast dataset in Fig. 11 (Making Pancake), Fig. 12 (Making Sandwich), and Fig. 13 (Making Coffee). All the qualitative results are for the setting α=0.2 and β=0.5, and we show two samples for each approach. The results show that our proposed approach is consistently better across all three videos, with greater alignment with the ground truth.

Method	Block	Overall			Unseen			Tail		
		Verb	Noun	Act	Verb	Noun	Act	Verb	Noun	Act
Testra [40]	short Attn [5]	25.1	30.8	14.1	24.3	24.5	10.7	17.4	23.0	10.9
Testra [40]	short Mamba [5]	27.9	34.1	15.2	28.1	24.2	12.0	20.5	27.8	12.3
Testra [40]	short MixMamba	29.7	35.6	17.1	30.4	24.8	13.5	22.7	30.2	14.1

Table 5. Results of action anticipation on EK-100 [6]. Accuracy measured by class-mean recall@5(%) following the standard protocol. "short" denotes using short-term memory.

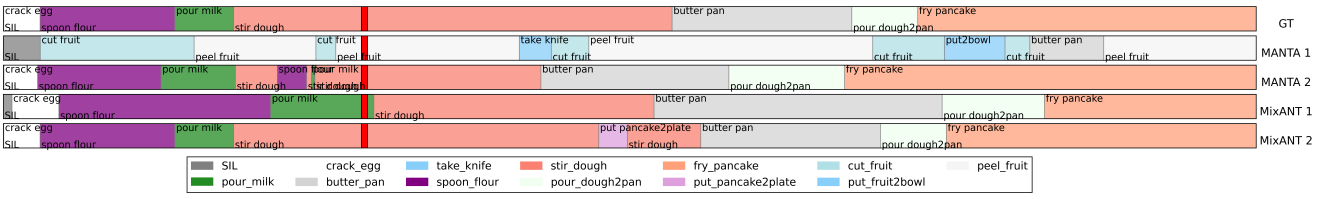

Figure 11. Qualitative figure for anticipation result on the Breakfast dataset for the video P42 making pancake.

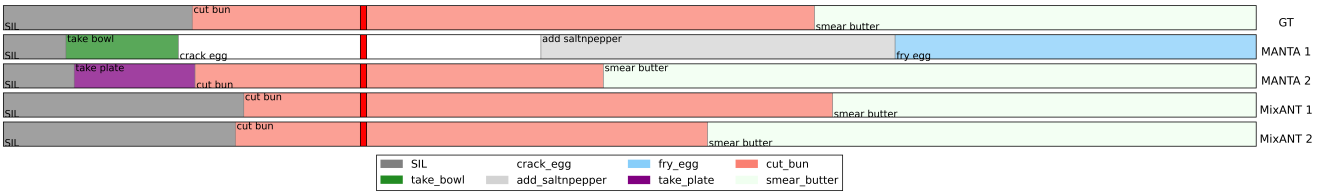

Figure 12. Qualitative figure for anticipation result on the Breakfast dataset for the video P47 making a sandwich.

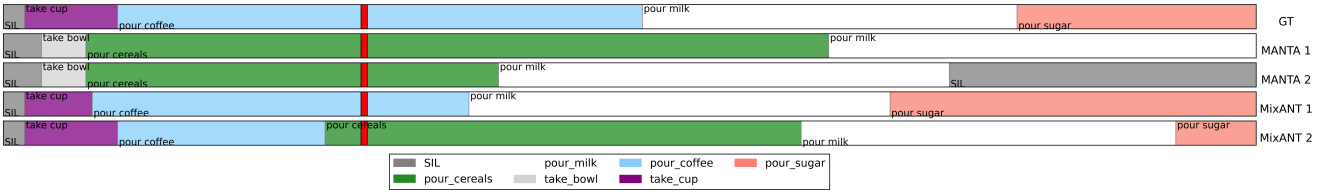

Figure 13. Qualitative figure for anticipation result on the Breakfast dataset for the video P53 making coffee.