ACTION ANTICIPATION WITH GOAL CONSISTENCY

Olga Zatsarynna and Juergen Gall

University of Bonn & Lamarr Institute for Machine Learning and Artificial Intelligence

ABSTRACT

In this paper, we address the problem of short-term action anticipation, i.e., we want to predict an upcoming action one second before it happens. We propose to harness high-level intent information to anticipate actions that will take place in the future. To this end, we incorporate an additional goal prediction branch into our model and propose a consistency loss function that encourages the anticipated actions to conform to the high-level goal pursued in the video. In our experiments, we show the effectiveness of the proposed approach and demonstrate that our method achieves state-of-the-art results on two large-scale datasets: Assembly101 and COIN. The code is available at `https://github.com/olga-zats/goal_consistency`.

***Index Terms*—** Action Anticipation, Action Forecasting, Activity Understanding, Video Understanding

1. INTRODUCTION

Anticipation of human actions is a task that we naturally solve in various day-to-day situations. We anticipate movements of cars while crossing the road, predict the plot elements of a new movie and picture how someone will react to our own actions. For all of these scenarios, we are able to imagine the future and adjust our beliefs and behavior accordingly. Due to how ubiquitous the situations that require the ability for action anticipation are, it is crucial that the intelligent agents designed to operate among human beings get hold of this task.

In this work we consider the setting of short-term anticipation, i.e., we want to anticipate a single action one second before it happens. Short-term anticipation has been addressed in different works [1–12], which showed impressive performance on this task in question. Most of these approaches, however, directly predict future actions without taking into account what has driven humans to undertake these actions in the first place. Yet, we observe that understanding the intent behind actions can simplify the task of anticipation. This is because different high-level goals are associated with different subsets of lower-level actions required to complete them. For example, as shown in Figure 1, if we know that the person's goal is to *attach a bumper* to a toy vehicle, we can conclude that the upcoming action could be *pick up bumper* or *position bumper* depending on the progress of the goal, but not

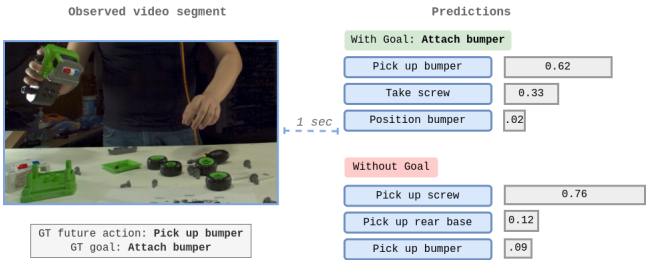

Fig. 1. To anticipate future actions, we propose to incorporate high-level intent prediction as part of our model. By relying on this information, the prediction model can filter out fine-grained actions that are not in correspondence with the pursued goal.

pick up rear base or *pick up cabin*. In this way, goal awareness reduces the number of valid options for future actions and thereby makes the task of anticipation simpler to solve. Motivated by this observation, we introduce in addition to the fine-grained action anticipation branch a separate goal prediction branch into our model. Nevertheless, simply forecasting both actions and their high-level goals independently does not explicitly ensure consistency between these two predictions. Therefore, we formulate an additional consistency loss that forces the fine-grained branch to predict actions that afford the completion of the pursued goal. Overall, the contributions of this work can be summarized as follows:

- We propose to harness high-level goal information to facilitate anticipation of future actions. We incorporate an additional goal prediction branch and propose a consistency loss to encourage alignment between predicted future actions and the underlying high-level intent.
- We demonstrate that our proposed approach achieves state-of-the-art results for the task of action anticipation on two large-scale procedural datasets: Assembly101 and COIN.

2. RELATED WORK

Action Anticipation in Videos. Works on action anticipation generally deal with one of the two established directions: long-term or short-term anticipation. Long-term anticipation works are focused on predicting multiple actions into the future with a forecasting horizon of several minutes. Short-term anticipation methods, on the other hand, focus on predicting only the next action a few seconds in advance. Both research

directions have received increased attention in recent years due to the availability of new large-scale datasets [13–15].

Starting with long-term action anticipation, Abu Farha *et al.* [16] introduced two anticipation approaches based on RNN and CNN networks. While the CNN network performed anticipation in one shot relying on a matrix representation of actions, the RNN model predicted actions and their length autoregressively and achieved superior results. To avoid the accumulation of errors due to the autoregressive prediction, Ke *et al.* [17] introduced a temporal convolutional time-conditioned network that anticipated all upcoming actions in one shot. Recently, Gong *et al.* [18] and Nawhal *et al.* [4] proposed two transformer-based [19] architectures for long-term anticipation.

The second line of work focuses on anticipation of the next action several seconds before its onset. Vondrick *et al.* [8] proposed to solve this task by regressing the representation of a single future frame and classifying it to get the future action prediction. Extending upon [8], Gao *et al.* [7] used an encoder-decoder network to process several observed frames and anticipate multiple future representations instead of just one. Another sequence-to-sequence approach was introduced by Furnari *et al.* [1] - an RU-LSTM model consisting of two LSTM networks for past summarization and future action prediction respectively. Zatsarynna *et al.* [2] proposed a temporal convolutional network to tackle the inefficiency of the approaches relying on the recurrent layers. In [20], Sener *et al.* presented a TempAgg model that used non-local-block [21] attention to encode the observed video snippets at different temporal scales creating recent and spanning features used for both short-term and long-term anticipation. Several recent works [5, 6, 22] proposed transformer-based architectures to allow for information flow between distant parts of the observed video segments, as well as enable spatial attention within individual video frames. In contrast to these approaches, our work focuses on harnessing information about action goals to predict future actions more accurately.

Intention-driven Forecasting. So far only very few works have addressed intention or goal-driven forecasting [23–25]. These works define intention in a different way or address other tasks. Debaditya *et al.* [25] defines the goal 'as the visual representation after performing the final action based on the procedure planning paradigm'. The approach thus aims to forecast visual features that are closer to the expected visual representation at the end of the sequence. Mascaro *et al.* [24] addresses long-term action anticipation and conditions a VAE on the high-level activity. Tanke *et al.* [23] forecast the future actions ahead of time to generate smooth and plausible human motion sequences.

3. METHOD

We start by formally defining the task of short-term action anticipation in Section 3.1. We then describe our proposed approach in Section 3.2.

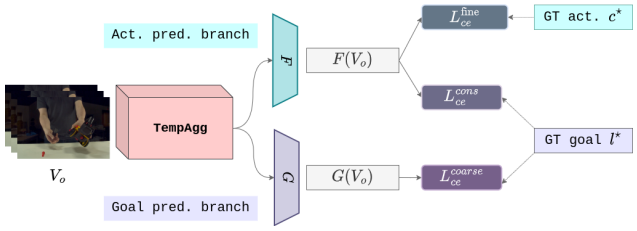

Fig. 2. Overview of our proposed approach. Our model contains two branches: an action branch that predicts future actions and a goal branch that predicts future goals. The goal prediction branch is trained using cross-entropy loss L_{ce}^{coarse}, while the action branch is trained with a combination of a cross-entropy L_{ce}^{fine} and a consistency loss L_{ce}^{cons}.

3.1. Task

Following [1, 20], we define the task of action anticipation as follows: given the observed video segment V_o that precedes the action of interest, we want to predict its label T_a seconds before the onset. In our work, we consider the anticipation time of one second, i.e. $T_a = 1$.

3.2. Model

Network branches. To address the above-defined task, we propose to harness information about the goals behind the upcoming actions. Intuitively, knowledge of the pursued goal simplifies the task of anticipation by constraining the number of valid future action choices. To make use of the goal information, as shown in Figure 2, the network consists of two prediction branches: a fine-grained action anticipation branch F and a goal prediction branch G. While the fine-grained action anticipation branch is trained for the final task of future action anticipation, the goal branch learns to predict the goals behind these actions. These two branches share the same backbone, namely TempAgg [20].

Formally, the fine-grained action branch F receives the observed video segment V_o as input and outputs the probability distribution of the next action $F(V_o) \in \mathbb{R}^{|C|}$, where C is the set of all fine-grained actions. Similarly, the goal branch G takes V_o as input and outputs probability distribution of the goal $G(V_o) \in \mathbb{R}^{|L|}$, where L is the set of all possible goal classes. Both branches are optimized using the cross-entropy loss:

$$L_{ce}^{fine} = -\sum_{n}\sum_{c\in C} \mathbb{1}(c = c_n^*)\log(F(V_o^n)_c), \qquad (1)$$

$$L_{ce}^{goal} = -\sum_{n}\sum_{l\in L} \mathbb{1}(l = l_n^*)\log(G(V_o^n)_l), \qquad (2)$$

where n is the batch index, $F(V_o^n)_c$ and $G(V_o^n)_l$ are c^{th} and l^{th} element of the corresponding probability distribution, and c_n^* and l_n^* are ground-truth action and goal labels, respectively.

Consistency loss. Simply incorporating a separate goal prediction branch into the model does not explicitly ensure that its predictions and the predictions of the action branch are

aligned with each other. Here, by alignment, we mean that a fine-grained action will lead to a progress or completion of a given goal. For example, a fine-grained action *screw chassis* is aligned with the goal *attach chassis*, while an action *screw water tank* is not. First, to understand which actions align with which goals, we compute how often individual goals and actions occur together in the training set. Then, to enforce the alignment on the model's fine-grained action predictions, we make use of the previously obtained co-occurrence statistics to formulate the consistency loss. More specifically, we first map fine-grained action predictions to probability distributions over the goal classes. To this end, we use the joint probability marginalization formula, where we substitute the action probabilities by the predictions of the fine-grained action branch:

$$\hat{G}(V_o^n)_l = \sum_{c \in C} P(l|c) F(V_o^n)_c. \tag{3}$$

To estimate the conditional distribution $P(l|c)$, we collect the action-goal co-occurrence matrix $M \in \mathbb{R}^{|L| \times |C|}$, where entry $M(l, c)$ stores the number of times goal l and fine-grained action c occurred together in the training set:

$$M(l, c) = |\{n \in \{1, \dots, N\} | c_n^* = c \wedge l_n^* = l\}|, \quad \forall c \in C, \forall l \in L. \tag{4}$$

Here, N is the total number of training examples in the dataset. Having acquired the co-occurrence matrix, we first approximate the joint probability distribution $P(l, c)$ and then obtain $P(l|c)$:

$$P(l, c) \approx \frac{M(c, l)}{\sum_{c', l'} M(c', l')}, \tag{5}$$

$$P(l|c) = \frac{P(l, c)}{\sum_{l'} P(l', c)}. \tag{6}$$

Finally, we compute our consistency loss as the cross-entropy loss between the obtained remapped goal distributions (3) and the true goal labels. Formally:

$$L_{ce}^{cons} = -\sum_n \sum_{l \in L} \mathbb{1}(l = l_n^*) \log(\hat{G}(V_o^n)_l). \tag{7}$$

This loss ensures that the predicted actions $F(V_o^n)$ are aligned with the underlying goals l_n^* according to the predefined action-goal hierarchy given by the conditional probability.

Final loss. To conclude, we train our network with the linear combination of the previously discussed branch-wise and consistency losses:

$$L = L_{ce}^{fine} + L_{ce}^{goal} + \lambda_{cons} L_{ce}^{cons}, \tag{8}$$

where λ_{cons} weights the consistency loss.

4. EXPERIMENTS

4.1. Datasets and Evaluation

Since our approach relies on a predefined action-goal hierarchy, we use two procedural activity datasets that contain hierarchical action annotations: Assembly101 [14] and COIN [26].

Assembly101 is a large-scale dataset that contains 362 recordings of 15 toy-vehicle assembly and disassembly sequences shot from 12 different viewpoints. The videos are annotated with 1M fine-grained and 100K coarse action segments, that we use as fine-grained actions and goals, respectively. Fine-grained segments span 1380 action classes composed of 90 objects and 24 verbs, while coarse actions span 202 action classes formed by 11 verbs and 69 objects. Assembly101 is divided into training, validation, and test splits. At the time of writing, the test set was not available, thus in our work we report results on the validation split. Following [14], we additionally provide results on two subsets of validation examples - *Tail* and *Unseen* - that contain video segments with tail action classes and toys unseen during training time, respectively.

COIN consists of 11827 videos that were collected from Youtube. The videos depict 180 high-level tasks and are annotated with 46354 action segments from 778 lower-level action classes. In our experiments, we regard video-level task annotations as goal actions, while segment-level action annotations as fine-grained actions. Training and testing splits contain subsets of 9030 and 2797 videos, respectively.

For performance evaluation, we used Class-Mean Top-5 Recall following [14] to account for the uncertainty of future predictions. For Assembly101, we report action, noun, and verb recall, while for COIN only action recall.

4.2. Implementation Details

For our experiments, we adopted the TempAgg model from [14] as the baseline and made use of RGB features provided by [14, 26] for the corresponding datasets. In addition to the already existing branches, we incorporated a separate goal prediction branch that operates on the spanning features, similar to [20]. For training, we used batch size 64 instead of 32. It improves the results as shown in the first two rows of Table 1[1].

4.3. Results

We present the results of our method on Assembly101 and COIN in Table 1 and 2, respectively. On both datasets, our method achieves improvements over the previously proposed TempAgg [14] approach. We note that the main focus is on the performance of action anticipation, while verb and noun predictions are secondary. On Assembly101 our model outperforms TempAgg [14] and our baseline (No goal) on the

[1]We follow the official protocol `https://github.com/assembly-101/assembly101-action-anticipation/tree/main/tempAgg-action-anticipation`, which differs from [14].

Model	M. Top-5 Rec.%												Params
	P.V. ACT.	P.V. NOUN	P.V. VERB	M.V. ACT.			M.V. NOUN			M.V. VERB			
	Overall	Overall	Overall	Overall	Unseen	Tail	Overall	Unseen	Tail	Overall	Unseen	Tail	
TempAgg [14]	8.19	25.59	54.61	8.53	8.34	3.94	26.27	23.00	25.93	59.11	58.77	53.10	207M
No goal	8.74	26.89	**55.85**	9.53	8.77	5.00	26.94	23.40	26.14	59.87	**59.73**	53.41	207M
Ours (1 goal)	10.39	27.50	54.59	11.29	9.69	6.71	27.66	23.32	26.84	58.40	58.17	52.59	+330.0K
Ours (2 goals)	**10.64**	**27.63**	55.82	**12.07**	**10.81**	**7.68**	**28.38**	**23.64**	**27.78**	**60.04**	59.63	**53.87**	+61.47K

Table 1. Action anticipation results on the Assembly101 validation set. *P.V.* and *M.V.* stand for per-view and multi-view evaluation, respectively. In the first case, different views of the same video sequence are considered as separate examples, while in the second case, only one prediction per video sequence is made by averaging results over all the views associated with it.

overall set of actions by 1.65% and 1.76% in the per-view and multi-view settings accordingly, while on COIN our approach achieves 0.54% improvement. On Assembly101, we further experimented with extending the model with one more branch (Ours (2 goals)) that predicts an even higher-level video sequence goal: assembly/disassembly of a particular toy type *(i.e. assembly truck, disassembly SUV)*. The consistency loss for this goal type is computed analogously to L_{ce}^{cons}. This extension brought further 0.25% and 0.78% improvements in the overall action recall for the per-view and multi-view settings, respectively. The increase of the number of model parameters by the goal prediction branches is very small as shown in Table 1 and 2. Since the goal branches are discarded after training, the inference cost remains the same.

Method	ACT. M. Top-5 Rec	Params
No goal	13.39	61.072M
Ours	**13.93**	+ 369.0K

Table 2. Action anticipation results on COIN validation set.

4.4. Ablation

In this section, we present the results of the ablation studies for our method. We inspect the impact of each loss term, ablate the formulation of the consistency loss, and analyze the effect of the loss weight λ_{cons}.

	ACT. M. Top-5 Rec.	
Method	COIN	Assembly101 (P.V.)
L_{ce}^{fine}	13.39	8.74
$L_{ce}^{fine} + L_{ce}^{goal}$	13.57	9.09
$L_{ce}^{fine} + L_{ce}^{goal} + L_{ce}^{cons}$	**13.93**	**10.39**

Table 3. Ablation of the final loss components.

Loss components. The loss function for our model consists of three terms: L_{ce}^{fine}, L_{ce}^{goal} and L_{ce}^{cons}. To analyze how individual terms impact the final performance, we train separate models with different combinations of these loss terms. The results of this experiment are presented in Table 3. Using only the fine-grained action anticipation loss L_{ce}^{fine} is always required and is equivalent to just applying the TempAgg [14] model. Adding the goal prediction loss improves the performance by 0.18% and 0.35% for COIN and Assembly101, respectively. Further extending the final loss function with the consistency loss L_{ce}^{cons} results in additional improvement of 0.36% and 1.30% accordingly. This shows the benefit of incorporating the additional goal prediction and consistency loss into the training loss function.

Consistency loss formulation. For computing the consistency loss, we make use of the ground-truth goal labels during training. Another possibility is to harness the predictions made by the goal branch instead. For comparison, we replace the cross-entropy loss by the KL divergence between the predictions of the goal branch and the remapped fine-grained action predictions. We show the results of this experiment in Table 4. We observe that the KL loss improves the performance compared to not using a consistency loss, but it is inferior to training with ground-truth goal labels. This result is intuitive since predicted goal distributions can be noisy as opposed to the ground-truth labels.

	ACT. M. Top-5 Rec.	
Consistency Loss	COIN	Assembly (P.V.)
Predicted	13.65	9.89
Ground-truth (Ours)	**13.93**	**10.39**

Table 4. Ablation of the consistency loss formulation.

Consistency loss weight. The consistency term in the final loss function is weighted by a factor λ_{cons}. We evaluate the impact of this hyper-parameter in Table 5. We observe that lower values of λ_{cons} perform better for the COIN dataset, while higher values perform better for Assembly101. The reason is that Assembly101 has a higher ratio of fine-grained actions to goals than COIN, which is compensated by a higher λ_{cons}.

Dataset	λ_{cons} / ACT. M. Top-5 Rec.			
COIN	0.1	0.5	1.0	2.5
	13.63	**13.93**	13.91	12.85
Assembly101	1.0	2.5	5.0	10.0
	9.65	10.12	**10.39**	10.24

Table 5. Ablation of the consistency loss weight.

5. CONCLUSION

In our work, we proposed to harness intent information to perform action anticipation by extending the model with a goal prediction branch and computing a goal-action consistency loss. We demonstrated that our proposed approach achieves stat-of-the-art results on two large-scale procedural activity datasets on the task of action anticipation.

6. ACKNOWLEDGEMENTS

The work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) GA 1927/4-2 (FOR 2535 Anticipating Human Behavior) and the ERC Consolidator Grant FORHUE (101044724).

7. REFERENCES

[1] A. Furnari and G. M. Farinella, "Rolling-unrolling lstms for action anticipation from first-person video," *TPAMI 2020*.

[2] O. Zatsarynna, Y. Farha, and J. Gall, "Multi-modal temporal convolutional network for anticipating actions in egocentric videos," in *CVPRW 2021*.

[3] O. Zatsarynna, Y. Farha, and J. Gall, "Self-supervised learning for unintentional action prediction," in *DAGM GCPR 2022*.

[4] M. Nawhal, A. A. Jyothi, and G. Mori, "Rethinking learning approaches for long-term action anticipation," in *ECCV 2022*.

[5] C. Wu, Y. Li, K. Mangalam, H. Fan, B. Xiong, J. Malik, and C. Feichtenhofer, "MeMViT: Memory-Augmented Multiscale Vision Transformer for Efficient Long-Term Video Recognition," in *CVPR 2022*.

[6] R. Girdhar and K. Grauman, "Anticipative Video Transformer," in *ICCV 2021*.

[7] G. Jiyang, Y. Zhenheng, and N. Ram, "Red: Reinforced encoder-decoder networks for action anticipation," in *BMVC 2017*.

[8] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating visual representations from unlabeled video," in *CVPR 2016*.

[9] M. Liu, S. Tang, Y. Li, and J. M. Rehg, "Forecasting human-object interaction: Joint prediction of motor attention and actions in first person video," in *ECCV 2020*.

[10] E. Dessalene, C. Devaraj, M. Maynord, C. Fermuller, and Y. Aloimonos, "Forecasting action through contact representations from first person video," *TPAMI 2021*.

[11] A. Miech, I. Laptev, J. Sivic, H. Wang, L. Torresani, and D. Tran, "Leveraging the present to anticipate the future in videos," in *CVPRW 2019*.

[12] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, "Recurrent neural networks for driver activity anticipation via sensory-fusion architecture," in *ICRA 2016*.

[13] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "The epic-kitchens dataset: Collection, challenges and baselines," *TPAMI 2021*.

[14] F. Sener, D. Chatterjee, D. Shelepov, K. He, D. Singhania, R. Wang, and A. Yao, "Assembly101: A large-scale multi-view video dataset for understanding procedural activities," *CVPR 2022*.

[15] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *ECCV 2018*.

[16] Y. Abu Farha, A. Richard, and J. Gall, "When will you do what?-Anticipating temporal occurrences of activities," in *CVPR 2018*.

[17] Q. Ke, M. Fritz, and B. Schiele, "Time-conditioned action anticipation in one shot," in *CVPR 2019*.

[18] D. Gong, J. Lee, M. Kim, S.J. Ha, and M. Cho, "Future transformer for long-term action anticipation," in *CVPR 2022*.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS 2017*.

[20] Fadime Sener, Dipika Singhania, and Angela Yao, "Temporal aggregate representations for long-range video understanding," in *ECCV 2020*.

[21] X. Wang, Ross B. Girshick, Abhinav Kumar Gupta, and Kaiming He, "Non-local neural networks," *CVPR 2018*.

[22] Y. Zhao and P. Krähenbühl, "Real-time online video detection with temporal smoothing transformers," in *ECCV 2022*.

[23] J. Tanke, C. Zaveri, and J. Gall, "Intention-based long-term human motion anticipation," 3DV 2021.

[24] E. V. Mascaro, H. Ahn, and D. Lee, "Intention-conditioned long-term human egocentric action anticipation," in *WACV 2023*.

[25] R. Debaditya and F. Basura, "Action anticipation using latent goal learning," in *WACV 2022*.

[26] T. Yansong, D. Dajun, R. Yongming, Z. Yu, Z. Danyang, Z. Lili, L. Jiwen, and Z. Jie, "Coin: A large-scale dataset for comprehensive instructional video analysis," *CVPR 2019*.