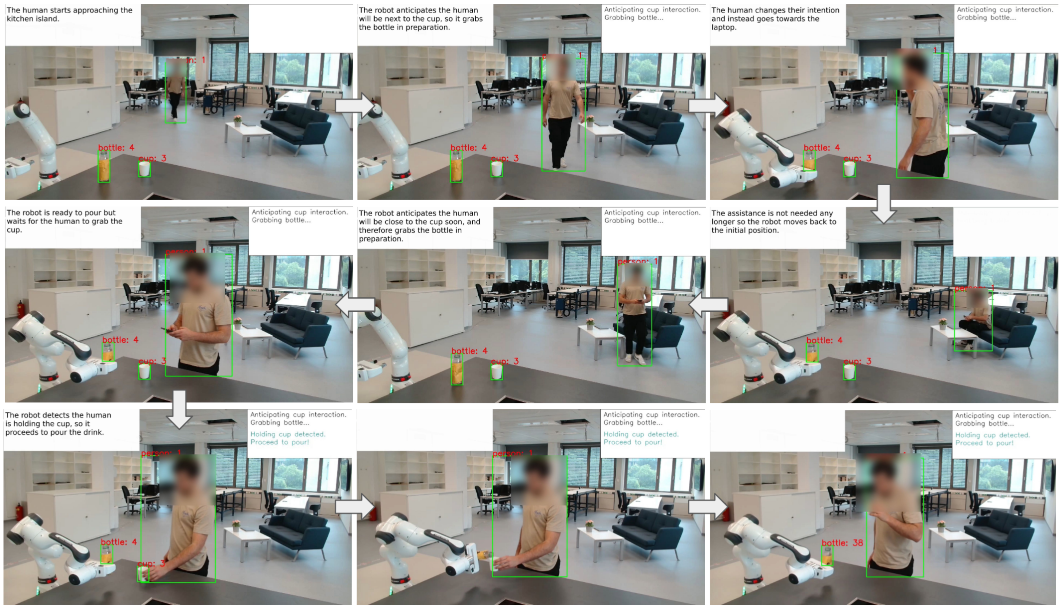
HOI4ABOT: Human-Object Interaction Anticipation for Human Intention Reading Assistive roBOTs
Supplementary Materials

Anonymous Author(s)
Affiliation
Address
`email`


The human starts approaching the kitchen island.

The robot anticipates the human will be next to the cup, so it grabs the bottle in preparation.

Anticipating cup interaction.
Grabbing bottle...

The human changes their intention and instead goes towards the laptop.

Anticipating cup interaction.
Grabbing bottle...

The robot is ready to pour but waits for the human to grab the cup.

Anticipating cup interaction.
Grabbing bottle...

The robot anticipates the human will be close to the cup soon, and therefore grabs the bottle in preparation.

Anticipating cup interaction.
Grabbing bottle...

The assistance is not needed any longer so the robot moves back to the initial position.

The robot detects the human is holding the cup, so it proceeds to pour the drink.

Anticipating cup interaction.
Grabbing bottle...
Holding cup detected.
Proceed to pour!

Anticipating cup interaction.
Grabbing bottle...
Holding cup detected.
Proceed to pour!

Anticipating cup interaction.
Grabbing bottle...
Holding cup detected.
Proceed to pour!

Figure 1: Real-world experiments scenario

The accompanying Supplementary Materials include the code to facilitate the reproduction of the results as well as an additional video to show the qualitative results of our HOI4ABOT model in real-time and working together with a robot to enhance its human intention reading capabilities.

1 Experimental Scenario

Our HOI4ABOT framework enhances the human intention reading through HOI anticipation. We conduct a real-world experiment with a Franka Emika Panda robot to support our proposed approach. Fig. 1 provides a step-by-step overview of the considered bartender scenario. First, the robot detects a human in the scene and anticipates the human intention to approach a kitchen island. When the robot anticipates with confidence that the human will be close to the cup, it executes a movement to grab the bottle, thus preparing for pouring. If the intention of the human changes, the robot adapts its behavior and moves back to the initial position after placing the bottle on the table. on the other hand, if the human proceeds to grab the cup, the robot pours the drink and goes back to its initial position. This preparatory behavior reduces the serving time while improving the overall experience for the human.

2 Implementation Details

In this section, we offer a comprehensive summary of the implementation details to aid in the reproduction of the experiments and the replication of the results. All experiments were conducted using a single NVIDIA RTX A4000 graphics card with 16GB of memory and an Intel i7-12000K CPU.

Hyperparameters. All trained models are conducted using the same strategy as [1]. We use the official code from https://github.com/nizhf/hoi-prediction-gaze-transformer and implement our HOI4ABOT model into their framework. All training settings are summarized in Table 1. We adopt Cross Binary Focal Loss [2] with $\gamma = 0.5$ and $\beta = 0.9999$, which improves training in extremely imbalanced datasets, such as VidHOI [3]. We train our models using the AdamW optimizer [4]. We define a scheduler for the learning rate, with an initial value of 1×10^{-8} that increases to a peak value of 1×10^{-4} in 3 warm-up epochs. The learning rate then decreases with an exponential decay with a factor 0.1. We run the training for 40 epochs.

Model configuration. All trained models use a similar configuration, but some variants such as 'stacked' or 'single' are adapted to ensure having a similar number of trainable parameters in the architecture (57.04M). All models reported in our paper use the DINOv2 [5] as the image feature extractor, using the smallest variant available ViT-B/14 that only contains 22.06M parameters; and CLIP [6] for the semantic extractor, with the largest available variant ViT-L/14 that contains 85.05M parameters. However, due to the fact that the number of objects in the dataset is limited, we pre-extracted the features for all possible objects. For our baseline HOI4ABOT model, we consider two transformer models with cross-attention layers, each of them with depth 4 and MLP expansions of ratio 4.0. Each transformer uses the multi-head attention variant with 8 heads to better extract the relationships within a sequence of features. Moreover, we consider sinusoidal positional embedding to facilitate learning the temporal information of a sequence. Finally, we consider the embedding size of each extracted feature, bounding box, or image feature, as 384. The embedding size for the prepended class token is also 384, as this is the embedding dimensions of the features extracted using DINOv2. For the semantics, CLIP obtains a feature of dimensionality 764.

Table 1: Training settings.

Optimizer	AdamW
Weight Decay	1.0e-2
Scheduler	ExponentialDecay
Warmup Epochs	3
Initial LR	1e-8
Peak LR	1e-4
Exponential Decay	0.1
Epochs	40
Random Seed	1551
Augmentation	Horizontal Flip
Flip Ratio	0.5
Batch Size	16
Dropout	0.1

Table 2: Model settings.

Transformer Depth	4
Number of Heads	8
Feature Extractor	DINOv2: ViT-B/14 [5]
Semantic Extractor	CLIP: ViT-L/14 [6]
Embedding Dimension	384
Positional Embedding	Sinusoidal
Exponential Decay	0.1
Mainbranch	humans
MLP ratio	4.0

3 Inference time

Our model is able to run in real-time thanks to the efficient design and reduced dimensionality.

Inference time versus the number of human-object pairs. Due to the nature of HOIs, each interaction needs to be computed for each human-object pair existing in the scene at a given time step. Therefore, to speed up the results and parallelize the forward pass for a given video, we stack all found human-object pairs in the batch dimension. Still, we consider it necessary to observe how different models' inference speed is affected by the number of pairs in a given video. Therefore, we run 1000 executions of our model processing a given video with I interactions. We implement all

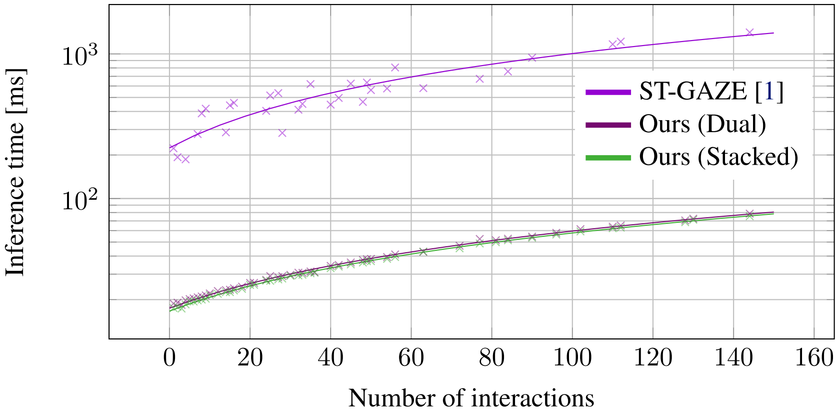


Figure 2: Model performance depends on the number of interactions for different architectures. Our variants ('Dual' and 'Stacked') have similar inference times (curves overlap) while outperforming by large margins the ST-GAZE model [1]

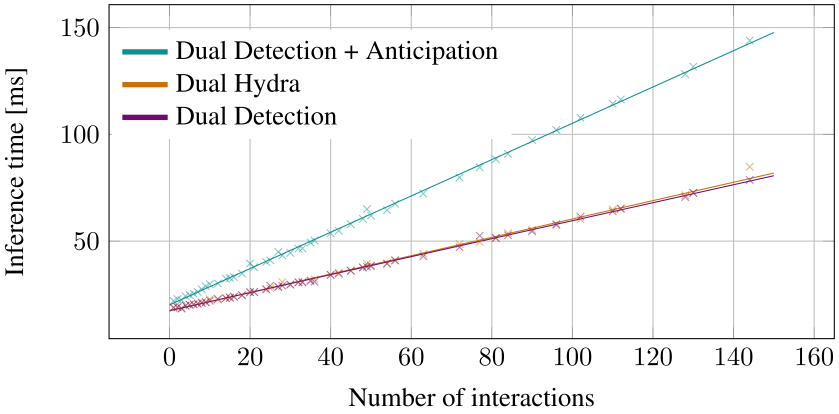


Figure 3: Model performance depends on the number of interactions for different model variants. The proposed multi-head approach allows us to detect and anticipate HOIs at multiple time horizons while maintaining a similar inference speed as the 'Dual' version (purple and dark orange curves overlap). We observe the benefit of the Hydra compared to running a specific 'Dual' transformer per detection and per anticipation.

models reported in Fig. 2 and 3 in the same batch strategy and observe a similar tendency in the increase of the inference time for a higher number of interactions.

Efficiency comparison with current state-of-the-art [1]. Both HOI4ABOT and [1] adopt a transformer-based architecture to comprehend the temporal relationships between the humans and objects in the scene. However, our model is designed to be efficient and to run in real-time despite having a large number of interactions, contrary to [1]. The comparison of the efficiency of both models is depicted in Fig. 2, which shows that our HOI4ABOT outperforms [1] by large margins in terms of speed. Next, we list the major differences in the model design that cause our improvement. First, we do not use any additional modality to predict HOIs, compared to [1] that leverages pre-extracted gaze features to capture the human's attention. Predicting these gaze features is costly as it requires detecting and tracking each human's head in the scene, predicting the corresponding gaze per human, and matching it to the corresponding body. Thus the speed decreases considerably depending on the number of humans in the scene. Moreover, [1] also considers an initial spatial transformer that leverages all humans and objects per frame, thus [1] speed is more affected by the number of frames considered.

Table 3: Anticipation mAP in Oracle mode.

Method	t	mAP	Preson-wise top-5			
			Rec	Prec	Acc	F1
STTran	1	29.09	**74.76**	41.36	36.61	50.48
	3	27.59	**74.79**	40.86	36.42	50.16
	5	27.32	**75.65**	41.18	36.92	50.66
Zhifan	1	37.59	72.17	59.98	51.65	62.78
	3	33.14	71.88	60.44	52.08	62.87
	5	32.75	71.25	59.09	51.14	61.92
Dual (scratch)	1	**38.46**	73.32	63.78	55.37	65.59
	3	34.58	73.61	61.7	54	64.48
	5	33.79	72.33	63.96	55.28	65.21
Dual (Hydra)	1	37.77	74.07	64.9	**56.38**	**66.53**
	3	34.75	74.37	64.52	56.22	**66.4**
	5	34.07	73.67	65.1	56.31	**66.4**
Stacked (scratch)	1	36.14	70.03	64.61	53.99	64.34
	3	34.65	73.85	62.13	54.15	64.77
	5	34.27	72.29	61.81	53.65	64.03
Stacked (Hydra)	1	37.8	72.05	**65.58**	56.23	66.09
	3	**34.9**	72.96	**65.05**	**56.3**	66.2
	5	**35**	72.86	**65.18**	**56.36**	66.2

Efficiency comparison of the Hydra HOI4ABOT. Human intention reading requires understanding both current and future HOIs. Therefore, we develop a multi-head HOI4ABOT, called Hydra, that allows us to predict HOIs at different time horizons in the future through a single forward step. While Table 3 shows the benefit of our Hydra variant compared to training from scratch, in this subsection we focus on the benefit of efficiency. Fig. 3 shows the inference time in milliseconds depending on the number of human-object pairs across different variants. We consider the 'Dual detection' as the baseline of our HOI4ABOT model when only predicting the HOI in the present. 'Dual Detection + Anticipation' is an optimized model that uses two dual transformer blocks that benefit from the same image backbone, one for HOI detection and the other for HOI anticipation in a single future $\tau = 3$. Finally, our 'Dual Hydra' performs HOI detection and anticipation for $\tau = [0, 1, 3, 5]$ in a single step by using our multi-head strategy. We observe the benefit of our Hydra variant compared to the model ensemble, as it has a comparable speed to the single head while anticipating HOIs in three additional future horizons.

4 Extensive comparison with variants

Our HOI4ABOT model outperforms the current state-of-the-art across all tasks and metrics in the VidHOI dataset, as shown in Tabel 3. In this section, we extend the comparison from the manuscript for the HOI anticipation for our 'Dual' and 'Stacked' variants, both when being trained by scratch or through the multi-head Hydra mode. Our results show that the 'Stacked' variant obtains slightly better performance in the mAP for longer futures. We consider this marginal improvement to be motivated because of the width difference in the transformer blocks, as well as the bigger representation space from which we project when classifying the HOIs. The 'Stacked' variant is based on a single self-attention block that operates on the human windows and object windows stacked in time. Therefore, the 'Stacked' transformer has double the width compared to the 'Dual' variant. Given that the output of a transformer model has the same shape as its input, the obtained tokens are also wider in the 'Stacked' variant. Having a bigger embedding dimension in the projected token allows the encoding of more information, which could result in better performance. However, Table 3 shows that the 'Stacked' variant has a lower recall and therefore lower F1-Score. These findings might indicate that the 'Stacked' variant struggles when anticipating HOIs in the videos where the

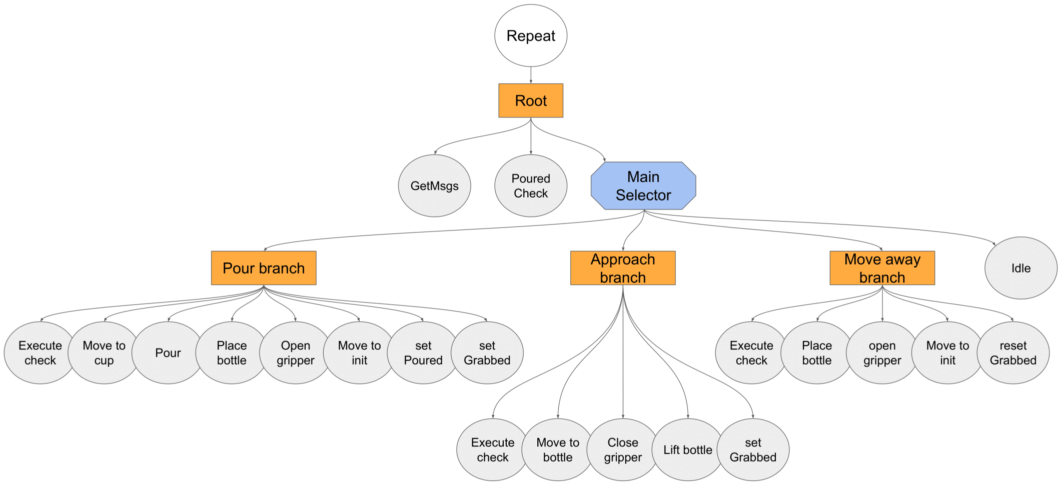


Figure 4: Schematic of the Behaviour Tree for our HOI4ABOT framework.

interaction changes in the anticipation horizon, being more conservative in its predictions. Therefore, we consider the 'Dual' variant to be optimal as it balances both precision and recall metrics across all tasks, as shown by outperforming all other models in the F1-score for the Hydra version.

5 Behavior Tree

In this section, we describe the structure of the Behavior Tree [7] used in our real-world experiments, which is shown in Fig. 4. The primary focus of this work is to enhance the assistive ability of robots through human intention reading using HOI anticipation. We conduct a simple real-world experiment with a Franka Emika Panda robot to showcase the benefits of our approach. This paper does not intend to provide a general development of BT for HOI tasks. However, the same methodology employed can be extended to more complex scenarios thanks to the modularity of BT.

The entire tree is built from three sub-trees: the 'Pour branch', the 'Approach branch', and the 'Move Away branch'. First, the 'Pour branch' is responsible for pouring the liquid into the cup. It is executed once the bottle is grabbed, and the 'hold' interaction between the human and the cup is detected. To achieve this conditional execution we add the 'Execute check' behavior at the beginning of the branch. Then, we reset the 'Grabbed flag' and set the 'Poured flag' to prevent any potential duplication of pouring into the cup. Secondly, the goal of the 'Approach branch' is to grab the bottle. This sub-tree is executed when the bottle is not currently grabbed and the robot anticipates the 'next to' interaction with a confidence greater than a pre-defined threshold. Once the bottle is grabbed, the 'Grabbed flag' is set. Thirdly, the 'Move Away branch' is responsible for releasing the bottle and moving back to its initial position. This branch is executed when the bottle is grasped by the robot and the robot anticipates the interaction 'next to' with a confidence lower than a predefined threshold. After executing the movements the 'Grabbed flag' is reset.

The appropriate sub-branch is selected by using the 'Main Selector' composite node. This node attempts to execute each sub-tree starting from left to right. The selector node executes the next branch in the sequence when the check in the preceding branch is not satisfied. Finally, the last behavior in the sequence is an 'Idle' behavior where the robot waits for a short period of time.

The root of the tree is a sequential node, which first collects all messages from the appropriate ROS topics, next checks if the beverage has been already poured, and finally executes the 'Main Selector'. To achieve continuous operation, the 'Root' node is decorated by a 'Repeat' modifier, which executes the root node indefinitely.

References

[1] Z. Ni, E. Valls Mascaró, H. Ahn, and D. Lee. Human-object interaction prediction in videos through gaze following. *Computer Vision and Image Understanding*, page 103741, 2023. ISSN 1077-3142. doi:https://doi.org/10.1016/j.cviu.2023.103741. URL `https://www.sciencedirect.com/science/article/pii/S1077314223001212`.

[2] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

[3] M.-J. Chiou, C.-Y. Liao, L.-W. Wang, R. Zimmermann, and J. Feng. St-hoi: A spatial-temporal baseline for human-object interaction detection in videos. In *Proceedings of the 2021 Workshop on Intelligent Cross-Data Analysis and Retrieval*, ICDAR '21, page 9–17, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385299. doi:10.1145/3463944.3469097. URL `https://doi.org/10.1145/3463944.3469097`.

[4] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, (6-9):2019, 2019.

[5] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[6] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[7] M. Colledanchise and P. Ogren. *Behavior Trees in Robotics and AI: An Introduction*. 07 2018. ISBN 9781138593732. doi:10.1201/9780429489105.