FICTION: 4D Future Interaction Prediction from Video

Supplementary Material

A. List of supplementary materials

We attach a supplementary video containing an overview of the paper, including dataset and result visualization. We will also release the interaction dataset and code.

B. Future interaction dataset

This section contains additional details about the future interaction dataset, discussed in Sec. 3.3.

3D object bounding boxes. We use Detic [118], along with the object taxonomy from LVIS [39], to find the mapping between the pixels in the video frames and the object labels. Since the inference from this method is fast, we perform segmentation at the original frame rate, i.e., 30 frames per second. Note that each pixel on the SLAM camera has an associated 3D location, which we use to map the object labels to a point in the 3D space. We perform object segmentation on SLAM frames directly because they have a direct mapping from 2D to 3D.

The DBSCAN [26] algorithm mentioned in Sec. 3.3 is useful in tightening the bounding boxes. For example, if there are two chairs in the scene, attempting to create a bounding box directly results in the box containing everything between the two chairs. Thus, we use DBSCAN to find the approximate bounding boxes. We set $50cm$ as the threshold for distinct clusters, and require 100 points at least to register as a unique object. This choice can correctly capture most of the objects seen in the chosen scenarios.

Extracting body poses. As mentioned in Sec. 3.3, the dataset has only one actor per video. However, there are other people present in some views. They are either bystanders or data collection volunteers. The dataset does not provide a full coverage of the annotation of the main actor in the videos. Thus, we use our heuristics to use the multi-view and disambiguate the main actor. Furthermore, the dataset contains multi-view videos showing the same person. We use the following two observations to extract the human pose. Firstly, the camera rig in the capture setup ensures the actor will have the largest area in all the video frames. Secondly, the similarity of the poses of the same person from all views will be higher than with people in the background. We use these observations to find the actor and then choose the *best* view—having the maximum joint visibility—to obtain the extracted body pose. An alternative is to focus on maximum hand visibility. However, we do not over-emphaisze on the hands. Furthermore, comparing to manually annotated 3D poses in Ego-Exo4D (available for only a subset of the data), the MPJPE error is 82mm when we use max joints and 115mm when we use maximum hand visibility.

Finding interaction instances. We use the following prompt for Llama 3.1-8B [21]:

> **System:** You are a helpful AI assistant. Match the narrations with the object labels that is provided.
> **User:** You are given narrations labeled by human annotators for a video. You are also given a set of object labels as per an object detection vocabulary. Find all instances of object interaction where the person would touch an object and map it to all the synonyms or similar words in the vocabulary. Sentences like 'C looks at the fridge' has no object interaction. Objects like cup, glass can be grouped together. Here are the object labels that you have to use: {labels}. Answer in this format:
> 1. {rewrite first narration} - answer: (object1, object2)
> 2. {rewrite second narration} - answer: NO INTERACTION
> 3. {rewrite third narration} - answer: NO MATCHING OBJECTS. Use 'NO INTERACTION' and 'NO MATCHING OBJECTS' in cases with no interaction and matching objects, respectively. Here are the numbered narrations:
> {narrations}

C. Details of baseline implementation

We introduce the baselines in Sec. 4. None of the baselines are directly applicable for 4D interaction prediction. Thus, we appropriately modify related models to create strong baselines for comparison. The model and task-specific adaptations are listed below:

- **HierVL [5]** is a recent method in Ego4D [37] long-term anticipation (LTA) benchmark with publicly available codebase. This LTA version generates future action labels (nouns and verbs). We use the output noun and locate the same in the 3D space, and mark all voxels for the predicted object as future interaction locations. Since HierVL is initially pretrained on Ego4D, we do not need to finetune the dataset since the egocentric videos are from a similar distribution. We do, however, finetune the last layer to match the output class dimension to the objects detected in Ego-Exo4D scenes.
- **OCT [63]** is a recent work in joint hand motion and interaction hotspot prediction from EPIC-Kitchens-100 [18]. We use this method to predict future interaction hotspot

for the next 3 minutes. We then use the camera parameters in Ego-Exo4D to map the 2D interaction points into the 3D environment. Since, this model is also trained on egocentric videos, and just requires images as input, we do not retrain this method on Ego-Exo4D.

- **OccFormer [114]** and **VoxFormer [55]** are methods originally designed for occupancy map prediction. We replace the image encoder in these networks with the video encoder f_V, used in our method.
- **4D-Humans [35]** and **T2M-GPT [111]** are recent works with autoregressive pose prediction capabilities. 4D-Humans extracts body pose from images and videos. We use the pose prediction module that predicts the next pose given the current body pose. We use this transformer autoregressively to generate multiple possible poses in the future. Similarly, T2M-GPT converts the body pose into a VQ-VAE based tokens and then predicts the pose tokens. The model is originally designed to generate pose based on the text condition; we modify the model to input prior pose tokens. Since our focus is not on *when* a pose is happening but rather *where*, we generously choose the prediction as the closest pose to the ground truth interaction location, out of all the generations. Both the methods are trained on large-scale pose datasets [45, 66]. Regardless, we finetune T2M-GPT (called **T2M-GPT-FT**) on our dataset to investigate the role of the training data. We choose to finetune the latter model due to a better performance and the stable nature of VQ-VAE codebooks for pose token generation.
- **Video-to-pose** CVAE [93] model takes as input the video of the person and generates a future pose distribution. We use the same video encoder f_V but do not provide any additional 3D context and expect the model to learn the 3D semantics implicitly. We train this method on our dataset. At inference, we choose the pose closest to the ground truth location.

Qualitative comparison with baselines. Fig. 5 compares our output with baselines. We see that our method is able to predict the interaction location and pose better than both the baselines. Autoregressive methods cannot predict long-term change in location and pose, while video-to-3D additionally misses the correct environment context.

D. Additional ablations

We also experiment with different choices of hyperparameters. We only report numbers on training performed on the cooking scenario. The numbers are reported for the validation split, distinct from the testing split, mentioned in Sec. 4. We only report PR-AUC and MPJPE for location prediction and pose prediction, respectively.

Effect of the observation time τ_o. Table 2 shows the results. We see that past video observation is crucial for providing the activity context. Thus, not providing any location gives the worst performance. The performance with 30 seconds of past observation is at par with other past observation durations. Therefore, we choose $\tau_o = 30$ so that the model has enough context for interaction prediction.

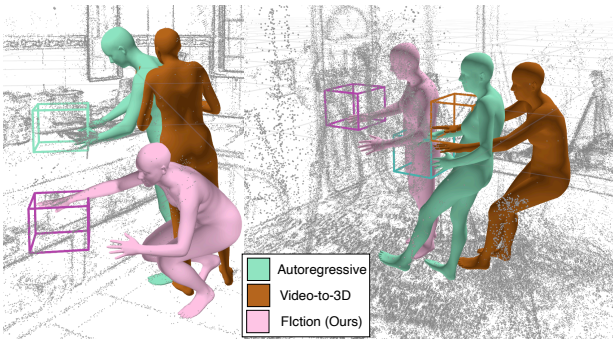

Figure 5. Comparison of our method with baselines and a cooking (left) and a bike-repair (right) take.

Location prediction				Pose prediction			
$0s$	$30s$	$60s$	$120s$	$0s$	$30s$	$60s$	$120s$
16.0	**21.2**	21.0	**21.2**	225	215	213	**212**

Table 2. Effect of τ_o on the performance.

Effect of the future time τ_f. Table 3 shows the results. We see an expected trend that the task becomes more difficult as τ_f increases. However, at a very high τ_f, the interaction location prediction becomes an easier task since the person has navigated to a large part of the environment, thus making majority of the locations as ground truth. Thus, we choose $\tau_f = 180s$ as a challenging version of the future interaction location prediction.

Location prediction				Pose prediction			
$60s$	$120s$	$180s$	$600s$	$60s$	$120s$	$180s$	$600s$
22.6	21.4	21.2	21.4	**207**	212	215	220

Table 3. Effect of τ_f on the performance.

Effect of the learning rate. Table 4 shows the results. We see that the model performs the best with a learning rate of 5×10^{-5} for interaction location prediction, and 5×10^{-6} for future pose prediction. This same parameter is chosen for all testing, as mentioned in Sec. 3.4.

Location prediction			Pose prediction		
5.10^{-6}	5.10^{-5}	5.10^{-4}	5.10^{-6}	5.10^{-5}	5.10^{-4}
20.6	**21.2**	19.6	**215**	220	226

Table 4. Effect of learning rate on the performance.

Effect of the encoder model size. We use a simple transformer encoder $\mathcal{L}$ for encoding the environment context (Sec. 3.2). We experiment with varying number of transformer layers. We experiment with $2, 4$ and 6 layers.

Table 5 shows the results. We observe that the number increases with the number of layers. This suggests that the performance can be further improved, with a larger transformer size. We do not experiment beyond 6 due to hardware constraints.

Location prediction			Pose prediction		
2	4	6	2	4	6
20.2	20.9	**21.2**	228	222	**215**

Table 5. Effect of the model size on the performance. We vary the number of transformer layers.

E. Limitations

As discussed in Sec. 3, our current method assumed one actor per video. The model design cannot explicitly handle multi-person scenarios. We will handle multi-person scenarios in the future. Nevertheless, the single-actor problem is still challenging with scope for improvement. We also assume a static point cloud when creating the dataset, while in practice, the object location can change with time. It is possible to use 3D information only from the last time segment for improving the spatial input to the model, we do not consider this case for the ease of the I/O. Note that this simplification does not affect the curated dataset quality, since we use narrations from Ego-Exo4D [38] as an additional signal. Finally, we use state-of-the-art methods Detic [118], WHAM [92] and Llama 3.1 [21] for creating the dataset, which are prone to errors. Any future improvement in these domains will further strengthen our dataset quality and the resulting trained model.