Summarize the Past to Predict the Future: Natural Language Descriptions of Context Boost Multimodal Object Interaction Anticipation
[Supplementary Material]

Contents

1. Past summarization

1.1. Method details

Extracting frame-wise action context. A visualization of the action context extraction for a single frame is provided in Figure 8.

To obtain action descriptions consisting of verb-noun pairs, we first generate multiple image captions (e.g. *"a person cutting wood"*) by forwarding diverse prompts to the task-agnostic and modality-agnostic OFA model [15]. The exact prompts used are *"what does the image describe?"*, *"what is the person in this picture doing?"*, and *"what is happening in this picture?"*.

We then perform part-of-speech tagging on the natural-language captions using Flair [1], followed by a lemmatization using NLTK [3], to extract candidate verb-noun pairs intended to represent the frame's action description (e.g. *"cut wood"*). We obtain at most one verb-noun pair per processed frame by selecting the most frequently found pair. In the case of ties, we select the pair that was detected first.

The extraction of frame-wise salient objects $\mathcal{N}_s^F$ is described in Section 3.2 of the main paper.

To extract frame-wise held objects $\mathcal{N}_h^F$, we first obtain *labelless* bounding boxes of active objects from EPIC-KITCHENS VISOR [6] together with labeled object bounding boxes (of not necessarily active objects) from [16], as further visualized in Figure 8. To obtain *labeled* bounding boxes of active objects, a pair of bounding boxes detected by UniDet and VISOR is considered to show the same object if they exhibit an intersection over union (IoU) greater than a threshold θ_{IoU}, where we set $\theta_{IoU} = 0.25$. The labels corresponding to these bounding boxes together form the set $\mathcal{N}_h^F$ of held objects for this frame.

As we use UniDet pre-trained on COCO [9] for object detection in an off-the-shelf manner without further training, the domain of the object detector is not aligned with that of the Ego4D nouns. We thus perform some label merging to simplify the UniDet detection domain, e.g. merging "home appliance" and "pressure cooker" into "machine".

Cross-frame aggregation. See Figure 9 for an illustration of the cross-frame aggregation scheme, restricted to $\mathcal{A}$ for simplicity. An equal scheme is used to obtain $\mathcal{N}_s$ and $\mathcal{N}_h$.

For a given frame on which to predict, we mark the 150 previous frames to be processed by the context extraction models using a stride of 3 frames. The videos in the Ego4D dataset use 30 FPS. We thus process the preceding 5 seconds for each prediction frame. Yet, it is possible for a prediction frame to make use of action context obtained from more than 5 seconds in the past through the inclusion of action context computed for previous prediction frames.

After processing the individual frames, we separately post-process each action context category $c \in \{\mathcal{A}, \mathcal{N}_h, \mathcal{N}_s\}$ (noun-verb pairs, held objects, and salient objects) via a cross-frame aggregation scheme to merge consecutive

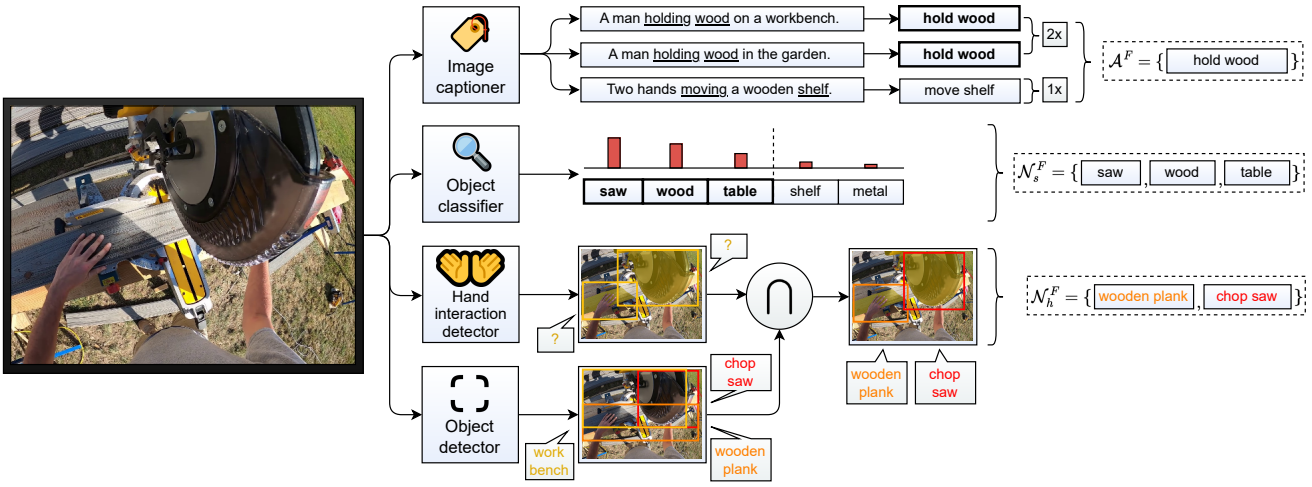

Figure 8. **Illustration of the frame-wise context extraction.** Frame-wise context is extracted using off-the-shelf models: an image captioner for obtaining verb-noun pairs for $\mathcal{A}$, an object classifier to keep the highest-ranking k objects (here $k = 3$) for $\mathcal{N}_s$, and a hand-object interaction detector used jointly with an object detector to obtain and subsequently label active object bounding boxes for $\mathcal{N}_h$. Details can be found in Section 3.2 of the main paper and Section 1.1 of this supplement.

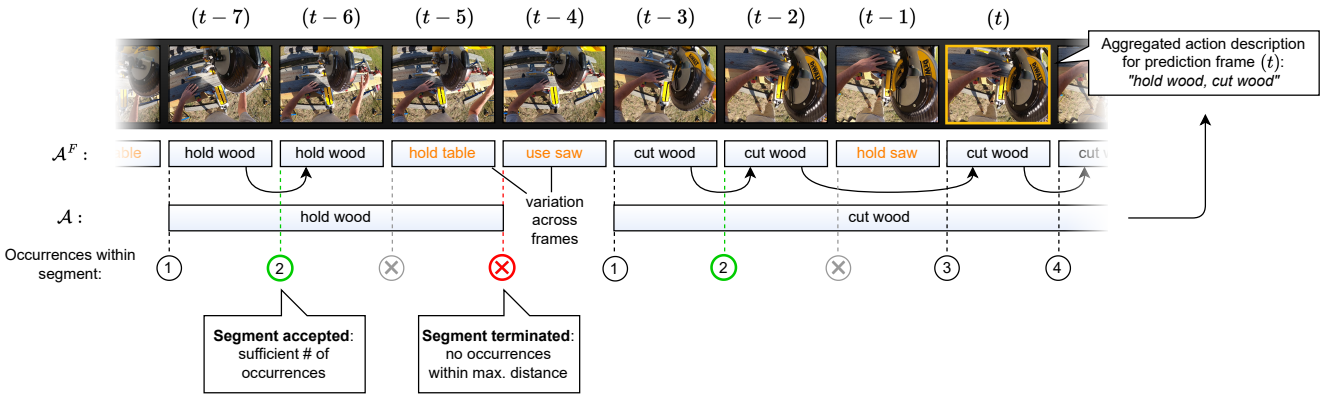

Figure 9. **Illustration of the *cross-frame aggregation* scheme, as used to construct action descriptions $\mathcal{A}$ from $\mathcal{A}^F$.** Identical schemes are used to construct $\mathcal{N}_s$ and $\mathcal{N}_h$. A sequence of identical frame-wise action descriptions (see Figure 8) with the term v forms a segment. As the aggregation traverses the frames, the segment is accepted into $\mathcal{A}$ once a number $P_{o,\mathcal{A}}$ of occurrences of v have been found with each at most $P_{\ell,\mathcal{A}}$ frames apart from the last, and terminated once no occurrences have been found within $P_{\ell,\mathcal{A}}$ frames. The preceding L_c segments up to a frame (t) in $\mathcal{A}$ form the action description for that frame.

frames with identical terms into segments. Note how working with language summaries allows us to opt for this simple duplicate elimination scheme, whereas embedding-based input is often repetitive and nontrivial to deduplicate.

More specifically, let $\mathcal{V}_c$ represent the vocabulary of category c, as detailed in the next subsection. For instance, $V_{\mathcal{N}_s}$ is the domain of nouns in the Ego4D short-term object interaction anticipation dataset.

The aggregation progresses through the video in a temporal manner, maintaining vectors of active and past segments. Unless a segment with the given term $v \in \mathcal{V}_c$ is already active, a series of occurrences of v in a sequence of frames that is at least $P_{o,c}$ long, with consecutive occurrences at most $P_{\ell,c}$ frames apart, leads to a segment being *accepted* into the list of active segments. Note that the segment is considered to *start* with the first occurrence that contributed to its acceptance, and is *terminated* once v has not occurred in the last $P_{\ell,c}$ frames. For a given context length L_c to be used when constructing the action context for a prediction frame, we construct a context of at most L_c non-overlapping active and/or past segments. Segments containing more occurrences of their term eliminate overlapping segments of different terms with fewer occurrences.

To construct the action context for a given prediction frame and action context category c, we distinguish between the *current* context, and the *past* context, with a context length of L_c resulting in 1 current and $L_c - 1$ past segments for $\mathcal{A}$ and $\mathcal{N}_h$. For $\mathcal{N}_s$, we do not consider past segments.

Model	Filtering	NO ↑	N ↑	N-V ↑
$\mathcal{A} + \mathcal{N}_s$	✓	**33.20**	**19.63**	**7.36**
$\mathcal{A} + \mathcal{N}_s$	✗	32.38	19.27	7.20
$\mathcal{A}$	✓	**32.41**	**18.67**	**7.16**
$\mathcal{A}$	✗	30.82	18.05	6.61

Table 7. **Evaluation of filtering of extracted verb-noun action descriptions.** We evaluate the performance obtained on the Ego4D validation set when training using filtered verb-noun pairs with nouns restricted to the Ego4D noun domain, and when using unfiltered verb-noun pairs. The results show that the filtered versions achieve better performance scores than their unfiltered counterparts.

	$\mathcal{A}$	$\mathcal{N}_s$	$\mathcal{N}_h$	SlowFast
t (ms)	340	200	280	200
M (GB)	5.5	3.0	5.4	9.5

Table 8. **Feature generation costs for a context unit**. We compare the time and the GPU memory requirements for generating each of the language inputs and the SlowFast features. Note that the generation of A, $\mathcal{N}_s$ and $\mathcal{N}_h$ can be parallelized, and we only utilize $A + \mathcal{N}_s$ for our final model.

Instead, we operate only using currently active segments, as salient objects change quickly throughout video frames and we are interested in summarizing the *recent* environment of the actor for this action context category.

We set $P_{o,\mathcal{A}} = 1$, $P_{o,\mathcal{N}_h} = 7$, $P_{o,\mathcal{N}_s} = 10$. We further use $P_{\ell,\mathcal{A}} = P_{\ell,\mathcal{N}_h} = P_{\ell,\mathcal{N}_s} = 7$. For the experiments in Table 1, $L_{\mathcal{A}} = 3$. For those in Table 2, Table 12 and Table 13, $L_{\mathcal{A}} = 4$. In all cases, $L_{\mathcal{N}_h} = L_{\mathcal{N}_s} = 3$, and version 1 of the dataset is used unless indicated otherwise.

Action context vocabularies.

For the verb-noun action description pairs $\mathcal{A}$, we denote the vocabulary $V_{\mathcal{A}} = V_{\mathcal{A},verb} \times V_{\mathcal{A},noun}$. We restrict $V_{\mathcal{A},noun}$ and $V_{\mathcal{N}_s}$ to the domain of the 87 noun classes used in the Ego4D dataset by eliminating all verb-noun pairs with nouns outside this domain during the cross-frame aggregation. To increase the number of frames for which action descriptions can be found, $V_{\mathcal{A},noun}$ additionally contains a small set of generic words such as "something" and "object", as we observed these to occur frequently in the captions generated by OFA.

As seen in Table 7, restricting $V_{\mathcal{A},noun}$ in this manner yields better performance for both $\mathcal{A}$ and $\mathcal{A} + \mathcal{N}_s$ models. We hypothesize that using a broad vocabulary might inhibit the model's ability to learn regularities in the language input, given the limited number of training samples available. $V_{\mathcal{N}_h}$ is the domain of UniDet object classes, while $V_{\mathcal{A},verb}$ consists of the lemmatized versions of all verbs in the output domain of OFA.

1.2. Computational cost and hyperparameters

Computational cost. The computational cost of generating language and video features is reported in Table 8. To obtain the language features, our final model configuration using $\mathcal{A} + \mathcal{N}_s$ needs only about 140ms more time per frame than when using SlowFast features, while requiring less GPU memory: 8.5 instead of 9.5 GB. In this calculation, we assume the systems to generate the action context language features to be running in parallel. We consider the two methods to have similar costs.

Hyperparameter sensitivity. We additionally present the effect that different context generation hyperparameters have on the quality of the generated action context. For $\mathcal{N}_s$, Table 9 shows the effect of k, the number of candidate salient objects that are kept per frame, using 3 metrics. Precision in Table 9 denotes the fraction of all inferred salient objects which are the respective frames' ground-truth NAO noun, while recall denotes how often the ground-truth noun appears in its frames' inferred salient objects. Similarly, Table 10 illustrates the effect of d, the maximum distance between verbs and nouns when extracting candidate verb-noun pairs from the natural-language image captions during $\mathcal{A}$ context construction. The number of exact hits in Table 10 represents how often the generated noun/verb matches the ground-truth Ego4D NAO noun/verb. The average GloVe [10] similarity in both tables is computed by averaging and then normalizing the 300-dimensional GloVe vector representation of the salient objects (for $\mathcal{N}_s$, Table 9) resp. verbs/nouns (for $\mathcal{A}$, Table 10) in the context description, and computing their dot product with the normalized GloVe embedding of the ground-truth noun/verb. It represents a less strict matching evaluation to account for the possibility of synonyms to the ground-truth: the closer the generated descriptions are, the higher the final average similarity is. Frame coverage shows how many frames we retain at least one salient object/verb-noun pair for after the cross-frame aggregation: larger numbers are better here since we reduce the risk of skipping important action steps. The average Glove similarity remains virtually the same for $k \geq 3$.

Sensitivity of $\mathcal{N}_h^F$ to noise from hand-object interaction and object detectors. To obtain a descriptive $\mathcal{N}_h^F$ for a given frame, both involved models, the hand-object interaction (HOI) detector and the object detector, must produce satisfactory results which can additionally be matched to each other. Specifically, the HOI detector must avoid omissions, false positives, undersegmentations and oversegmentations of active objects. The bounding boxes of active objects are obtained by taking the outer limits of the segmentation. The object detector must detect the active object and additionally assign it a correct label. Lastly, the bounding boxes produced by both models must sufficiently overlap so that the label inferred by the object detector can be assigned

k	Precision	Recall	Ø GloVe sim.	Frame coverage
1	**0.2582**	0.2566	**0.4221**	57.69 %
2	0.1892	0.2715	0.3656	82.13 %
3	0.1550	0.3106	0.3371	89.82%
4	0.1370	0.3374	0.3198	92.80%
5	0.1275	**0.3477**	0.3113	**93.93** %

Table 9. **Influence of parameter k on generated $\mathcal{N}_s$ action context**. Increasing k leads to better frame coverage (fraction of frames for which $\mathcal{A}^F$ is non-empty) and increased recall of the generated $\mathcal{N}_s$ on the Ego4D validation set, but reduces their precision and GloVe similarity to the ground-truth next active object nouns. As we consider recall and frame coverage to be more important than precision, and noisy detections are likely to be eliminated by the subsequent cross-frame aggregation, we choose $k = 5$.

d	Hits (N)	Hits (V)	Ø GloVe sim. *	Frame coverage
1	10.61%	**3.42%**	0.3348	75.18 %
2	13.12%	3.29%	0.3567	78.85 %
3	15.69%	3.05%	0.3752	79.41 %
4	**16.99%**	3.15%	**0.3850**	79.64 %
5	9.67%	2.65%	0.3426	**81.11 %**

Table 10. **Influence of parameter d on generated $\mathcal{A}$ action context**. We observe a sudden drop in the fraction of ground-truth correspondences for both nouns and verbs of $\mathcal{A}$ on the Ego4D validation set when transitioning from $d = 4$ to $d = 5$, likely caused by the introduction of spurious verb-noun pair detections. We hence choose $d = 4$. *Average between cosine similarity of $\mathcal{A}$-noun to ground-truth noun and $\mathcal{A}$-verb to ground-truth verb, measured using GloVe embeddings.

to the object segmented by the HOI detector. Examples of $\mathcal{N}_h^F$ we deem useful to the prediction task are visualized in Figure 17. We further showcase some failure cases of the HOI detector in Figure 18 and of the object detector in Figure 19. These failures lead to missing or incorrect $\mathcal{N}_h^F$.

2. The TransFusion model

The fusion module is based on the query-key-value (QKV) attention mechanism popularized by the Transformer [14] architecture. Such an attention aggregation scheme can loosely be interpreted as computing a weighted average of the value vectors v for each of the query vectors, where the weight is given by the compatibility between the query and key vectors: q and k. The final compatibility score is obtained after applying softmax on the pairwise dot products as described in Equation 1.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

This attention mechanism is applied multiple times in parallel through a set of attention heads, each one with a distinct set of parameters, such that the attention mechanism is allowed to focus on different input subspaces. The output is finally concatenated and projected to the initial token dimension. The multihead functionality is laid out in Equations 2 and 3.

$$MultiHead(Z) = Concat(head_1, ..., head_h)W^O \quad (2)$$

$$head_i = Attention(QW_i^Z, KW_i^Z, VW_i^Z) \quad (3)$$

In the following, we state the equations for the visual and language features' tokenization and projection, embedding addition, and concatenation operations prior to feeding the result to the TransFusion module for a single scale level.

$$Vf = patchify(Vf_i) \in \mathbb{R}^{N \times P^2 \cdot c} \quad (4)$$

$$Vf = VfW_p; W_p \in \mathbb{R}^{P^2 \cdot c \times D} \quad (5)$$

$$Lf = LM(X); lf \in \mathbb{R}^{L_A \times D} \quad (6)$$

$$Vf \mathrel{+}= Vf_{emb} + Pos_{emb} \quad (7)$$

$$Lf = Lf + Lf_{emb} \quad (8)$$

$$Lf = Dropout(Lf) \quad (9)$$

$$Vf = Dropout(Vf) \quad (10)$$

$$Z = Concat(Vf, Lf) \quad (11)$$

where $Vf_{emb} \in \mathbb{R}^D, Pos_{emb} \in \mathbb{R}^{N \times D}$, Lf represent the tokenized language features and Vf represent the tokenized visual features.

As described in Section 3, the TransFusion model consists of multiple transformer encoder layers applied in succession. This mechanism is replicated on multiple input scales to enhance the corresponding visual features. A single transformer encoder layer employs Layer Normalization [2], MLP blocks, multihead QKV self-attention, a Dropout module [12], and the GELU [8] non-linearity. The functioning is described in equation 12, where we drop the scale level indices for simplicity.

$$\begin{aligned} Z' &= LN(MultiHead(Dropout(Z)) + Z) \\ Z' &= MLP(Dropout(GELU(Z'))) + Z' \\ Z' &= LN(Z') \end{aligned} \quad (12)$$

3. Implementation details

For the majority of our runs, we use a learning rate of 1e-4. For training the backbone encoders, we additionally divide the learning rate by 5 to better synchronize with the fusion module dynamics that starts from a random initialization.

We augment the data by altering both height and width resolutions while ensuring their downsampled shapes are divisible by the patch sizes. The following height-width

pairs are used for most of the experiments: 480-596, 544-640, 640-768, 704-896, 768-896, 800-1200. Before rescaling the images, they are cropped randomly in a relative range of 0.9 for both height and width. This way, we preserve about 80% of the original visual area and reduce the chances of evicting ground-truth object boxes. The images are flipped horizontally with a probability of 50%. We apply a moderate amount of color jittering: we alter the brightness in a relative range of $[-0.15, 0.15]$, the contrast in $[-0.1, 0.1]$, and the hue in $[-0.05, 0.05]$. For reference, the Ego4D baselines resize the image height to 800 pixels while limiting the width to 1333 pixels. By the choice of height and width ratios, both approaches also provide a weak form of aspect ratio augmentation.

We use a 1D sinusoidal positional embedding for the visual tokens. The patch dimensions used per level are the following: high-resolution ResNet-50 runs use patch projection sizes of $4, 4, 2, 1$ for the FPN stages. Smaller patches tend to give better performance, but increase the computational cost of the self-attention mechanism that scales with the square of the number of tokens. We also apply language token dropout with $p \in \{0.1, 0.2\}$ (depending on the language model size) and visual token dropout with $p = 0.1$ before the transformer fusion layers, which slightly improves validation performance. Each of our models can be trained on a single NVIDIA A100 GPU with 80GB of VRAM when working with version 1 of the dataset. For experiments on version 2 of the dataset, we use 3 NVIDIA A100 GPUs per run.

During development, we observed that the TransFusion architecture reaches high confidence in predicting foreground objects with a corresponding local minimum of the loss before learning to effectively fuse the visual and language modalities. This diminished the final classification performance while improving the box localization performance. Hence, we reduced the region proposal network's sampling batch size and the detection network's image batch size from 256 and 512 to 64 and 128 respectively, penalizing the model less for foreground-background mismatches. This reduces the dependence on visual features, which are already adapted for object detection tasks. We also perform multiscale augmentation by resizing the shortest edge to measure between 480 and 800 pixels, random-relative cropping, color jittering, and image horizontal flipping to enable a longer learning stage and achieve a more effective feature fusion.

3.1. Impact of prediction postprocessing

After obtaining the boxes and associated noun, verb and TTC estimates from the model, we apply non-maximum suppression to the boxes and alter unlikely verb-noun pairs to eliminate duplicate or faulty predictions.

Specifically, for each possible verb-noun pair $(v, n) \in \mathcal{D}_v \times \mathcal{D}_n$, where $\mathcal{D}_v$ and $\mathcal{D}_n$ are the verb and noun domains of our dataset, we count the number of occurrences of (v, n) in the ground-truth labels of our training set. If no occurrences are found and a box with (v, n) is predicted by our model, the verb v is exchanged with the verb v' such that (v', n) occurs most frequently in the dataset for all $v' \in \mathcal{D}_v$.

Furthermore, for each prediction frame, we remove all boxes with associated verb-noun pairs (v, n) that have an intersection-over-union $i > \theta_{IoU,pp}$ with another box associated with (v, n), where we set $\theta_{IoU,pp} = 0.3$.

We report our model's performance on both versions of the Ego4D short-term object interaction anticipation dataset's validation set before and after the described postprocessing scheme in Table 11. We further include the performance of other state-of-the-art works on the validation set for comparison. Here, we emphasize that the better validation results of [4] do not reflect in their method's performance dataset's test set. Unlike the validation set's publicly available ground-truth data, the test set's ground-truth data is hidden, and the number of submissions to the Ego4D test set evaluation server is limited to 1 per day. This suggests a degree of overfitting to the validation set on part of [4].

Note that only the version of our model in Table 1 and the asterisk-denoted version of our model in Table 11 use the described postprocessing scheme. All other experiments reported in this work do not make use of this postprocessing to prevent it from influencing the ablation results.

Ver.	Model	N ↑	N-V ↑	N-T ↑	A ↑
v1	FRCNN+Rnd. [7]	17.55	1.56	3.21	0.34
	FRCNN+SF [7]	17.55	5.19	5.37	2.07
	InternVideo [4]	19.45	**8.00**	**6.97**	**3.25**
	StillFast [11]	16.20	7.47	4.94	2.48
	TF (ours)	<u>20.12</u>	7.48	6.13	2.60
	TF (ours) *	**20.47**	<u>7.74</u>	<u>6.56</u>	<u>2.78</u>
v2	FRCNN+SF.	21.0	7.45	7.04	2.98
	StillFast	20.26	10.37	7.16	3.96
	GANO$_{v2}$ [13]	20.52	10.42	7.28	<u>3.99</u>
	TF (ours)	<u>24.11</u>	<u>10.62</u>	<u>7.84</u>	3.70
	TF (ours) *	**24.57**	**11.06**	**8.37**	**4.02**

Table 11. **Performance on the Ego4D validation set.** We report the performance of our model and other state-of-the-art approaches on both versions of the Ego4D short-term object interaction anticipation dataset's validation set. Note that while we are outperformed by [4] on version 1 of the dataset's validation set with publicly available ground-truth data, we outperform their method on the test set (see Table 1) with hidden ground-truth data, suggesting overfitting to the validation set on their part. * Denotes our model with post-processing applied to the predictions.

4. Generalizability

To evaluate the generalizability and domain shift robustness of our model and context extraction approach, we randomly select a set of images from the EPIC-Kitchens 100 (EK100;

Figure 10. **Uncurated qualitative examples of cross-dataset predictions on EPIC-Kitchens 100.** We evaluate our model, trained using the Ego4D dataset, on randomly selected frames from the EPIC-Kitchens 100 [5] action recognition dataset, and show the top 2 highest-scoring detections on those frames without selecting for quality. Despite having been trained on a different dataset, TransFusion shows strong generalizability by predicting reasonable next object interactions on images from an unseen dataset. The aggregated action context is visualized in the bottom left corner of each image, with salient objects in the first row and action descriptions in the second.

see [5]) action recognition dataset and process them using a TransFusion model trained on the Ego4D short-term object interaction anticipation dataset. Since no ground-truth bounding boxes are available for EK100 and the noun classification domains of the datasets differ, we simply visualize the resulting top-scoring predictions for each frame in Figure 10. Importantly, we do not filter the results by the quality of the predictions. The action context is extracted just as for Ego4D. While the verb and noun domains are changed for the context extraction to match the new dataset, which introduces some mismatch due to unseen words now appearing in the context summaries, we do not finetune any part of our model to adapt to the new dataset. Still, our model shows a surprising robustness to the domain shift and yields useful predictions despite not having been trained on EK100 and dealing with novel vocabulary in the lan-

guage input. We hypothesize that the generic nature of the Ego4D dataset further facilitates the transfer to more specific datasets, such as the kitchen-based EK100.

5. Evaluation of action contexts

Correlation of model performance with language input. Table 2 provides a high-level comparison of the performance obtained using different types of action context language input. In Table 12, we further experiment with providing our models trained on $\mathcal{A}+\mathcal{N}_s$ and $\mathcal{N}_s$ different types of action context during inference. Specifically, we compare between (1) using the original action context class(es) each model was trained with as language input, (2) using the object to be interacted with next, taken from the ground-truth labels, as language input, and (3) an ablation where we omit all language input.

Ideally, our models trained with $\mathcal{N}_s$ and $\mathcal{A}+\mathcal{N}_s$ input should be able to make use of salient objects enumerated in the language input to better disambiguate between multiple possible next active objects (NAOs). The ground-truth NAO forms a reasonable "best-case" version of the $\mathcal{N}_s$ inputs: we would expect an increase in the models' performance if the NAO is highlighted to the model as the only salient object in the prediction frame. Indeed, as evidenced in Table 12, both the model trained on $\mathcal{N}_s$ and that trained on $\mathcal{A}+\mathcal{N}_s$ perform better when receiving the ground-truth NAO as input, than when receiving the salient objects $\mathcal{N}_s$ from the context generation models. On the other hand, the performance drops when the models do not receive any language input. We would like to point out that the performance of the language-aided models does not suffer significantly when omitting language input, suggesting that the models learn to benefit from the provided action context rather than becoming dependent on it.

We further conduct a comparison of our model's performance on samples for which the ground-truth NAO appears in the generated language input against that on samples for which it does not and calculate the absolute and relative frequencies of the ground-truth NAO's appearance. We showcase the results in Table 13. The aforementioned considerations strongly suggest that the models benefit from the $\mathcal{N}_s$ action context specifically due to its ability to highlight salient objects in the actor's environment.

Counterfactual analysis. We showcase how changing the action context alters the predictions of the model on various prediction frames from the Ego4D dataset's validation set. Figure 11 together with Figure 16 illustrate the difference in the predictions of an $\mathcal{N}_s$-trained model when using language input consisting of the ground-truth class name (left column), compared to using the class name of another object in the image or a similar-looking object (right column). We visualize the top 4 highest-scoring bounding boxes, along with the bounding box capturing the ground-truth next active object (in green) and the model's language input, shown on the bottom left of the images. The visualizations indicate qualitatively that TransFusion learns to effectively condition its predictions on action context encoded as language summaries to anticipate object interactions.

6. Comparison with state-of-the-art

In this section, we provide more details on the experiment setup reported in the main paper. The TTC values in this work are obtained using the provided Ego4D baseline checkpoint while keeping our original box, noun, and verb predictions. For versions of our framework making use of hand pose estimates and the TTC head for time-to-contact prediction, please check the official code repository. For the language encoder, we use SBERT 384 and for the visual encoder, we use the frozen Ego4D ResNet-50 weights. We discount the classification loss for the background class prediction by 0.8 such that the model focuses more on the actual object categories.

Validation-test performance variance. We observe some noticeable variance between the validation and test set performance, both for our model and the Ego4D method. We believe that this is caused by multiple factors, such as 1) only one validation fold being used during the training of the two models. Performing k-fold cross-validation provides a more reliable estimate of the *true* model performance, at the cost of a significantly larger computation time. 2) even when using a single validation fold, a reasonable performance estimate can be obtained. In our case, we identify a noticeable class distribution shift when moving from the training to the validation set. It is plausible that a similar distribution shift occurs between the validation and test set.

6.1. Additional comparisons

We provide additional insights on how our method performs compared to the Ego4D FRCNN+SF baseline, highlighting the effectiveness of our approach and the suitability of using language descriptions to summarize the action context.

Model performance as a function of label distribution. We evaluate the performance of the two models separately for the most frequent and for the tail class categories. This comparison confirms the effectiveness of our method: we register consistent improvements for both frequent and rare categories: over 15% and 22% for nouns and 132%, 161% for verbs respectively. This is very encouraging, seeing as improving performance in low-tail classes is a challenging aspect for many prediction tasks and the adoption of language descriptions could provide further breakthroughs. The results are presented in Table 14 and Table 15. We report the classification-only results (without conditioning on a correct box prediction) because we want to highlight the

Lang. used during training	Lang. used during inference	NO ↑	VO ↑	N ↑	N-V ↑
$\mathcal{A} + \mathcal{N}_s$	next active obj. (GT)	**37.40**	11.65	**21.86**	**8.05**
$\mathcal{A} + \mathcal{N}_s$	$\mathcal{A} + \mathcal{N}_s$	33.20	**12.00**	19.63	7.36
$\mathcal{A} + \mathcal{N}_s$	$\emptyset$	29.51	10.72	17.49	6.46
$\mathcal{N}_s$	next active obj. (GT)	35.42	9.24	20.78	7.33
$\mathcal{N}_s$	$\mathcal{N}_s$	33.33	10.61	19.62	7.13
$\mathcal{N}_s$	$\emptyset$	29.73	8.65	17.83	6.18
$\emptyset$	$\emptyset$	31.26	10.64	17.71	6.14

Table 12. **Performance of various action context combinations at train and inference time.** Comparison of performance obtained on the Ego4D validation set with models trained on $\mathcal{A}$ and $\mathcal{A} + \mathcal{N}_s$, as well as a visual-only model. For the language-aided models, we experiment with forwarding the ground-truth next active object labels, passing the intended action context to the model, and using no language input ($\emptyset$). Experimental results show that having the ground-truth NAO labels can further improve the model performance, and that even in the absence of language input ($\mathcal{A} + \mathcal{N}_s$ & $\emptyset$), our language-aided model performs competitively to a model trained without language input ($\emptyset$ & $\emptyset$).

	$\mathcal{A}$	$\mathcal{N}_h$	$\mathcal{N}_s$	$\mathcal{A} + \mathcal{N}_h$	$\mathcal{A} + \mathcal{N}_s$	GT
Abs.	4,662	1,119	5,390	5,184	7,484	11,576
Rel.	27.0%	6.49%	31.3%	30.1%	43.5%	67.2%
NO ↑	32.41	31.16	33.33	32.72	33.20	37.33
NO^+ ↑	**41.31**	**47.42**	**44.24**	**40.60**	**44.60**	**43.71**
NO^- ↑	30.52	30.97	31.09	31.60	31.22	30.78

Table 13. **Performance analysis based on the occurrence of next active object labels in language context.** Absolute (Abs.) and relative (Rel.) frequency of the occurrence of the ground-truth next active object class in different types of action context on the Ego4D validation set, together with the performance of models trained on these action context types. Performance is measured in terms of Noun-Only mAP and separately for the full validation set (NO), the subset where the ground-truth next active object class appears in the language summary (NO^+), and the subset where it does not (NO^-). The results show the benefit of the next active object appearing in the language input.

substantial classification improvements owed to improved semantic understanding. The corresponding metrics are denoted as NO and VO respectively.

Model	NO ↑	Top-10 NO ↑	Tail NO ↑
Ego4D	28.70	27.58	26.45
TransFusion	**33.80**	**31.84**	**32.37**
Improvement	18%	15%	22%

Table 14. **Noun-only mAP** for top 10 and tail noun categories on Ego4D dataset. We observe consistent gains over the full class spectrum, in particular in tail classes.

Model	VO ↑	Top-5 VO ↑	Tail VO ↑
Ego4D	5.22	10.11	4.38
TransFusion	**12.00**	**23.54**	**11.46**
Improvement	129%	132%	161%

Table 15. **Verb-only mAP** for top 5 and tail verb categories on Ego4D dataset. We observe consistently strong gains over the full class spectrum, in particular in tail classes.

Model performance dependence on bounding box size.. We additionally present a performance comparison based on the size of the ground-truth bounding box. We divide the ground-truth labels into 3 categories according to their area, each one containing a third of the total validation samples. The results for the Box-Noun mAP metric are shown in Table 16. We again notice consistent improvements across the different categories, which show that our method is flexible enough to perform well on difficult cases such as small bounding boxes or tail classes. This is encouraging because improving on these corner cases usually requires a lot of effort. The TransFusion model using language action context summaries achieves this without any special design choices aimed at improving this objective.

Model	Ns ↑	Nm ↑	Nl ↑
Ego4D	6.39	15.20	21.82
TransFusion	**7.91**	**18.65**	**22.75**
Improvement	23%	22%	4%

Table 16. **Box-Noun mAP** for small (Ns), medium (Nm) and large (Nl) boxes. Box-Noun-Verb mAP improvement is about 1.5 points for each of the categories hence it is omitted. Our method improves detection and classification on difficult cases with small bounding boxes, complementing the hard-to-encode visual cues.

With respect to the Box-Noun-Verb mAP, we notice regular gains of about 1.5 points over the Ego4D methods (hence, this result is omitted from the table). This suggests

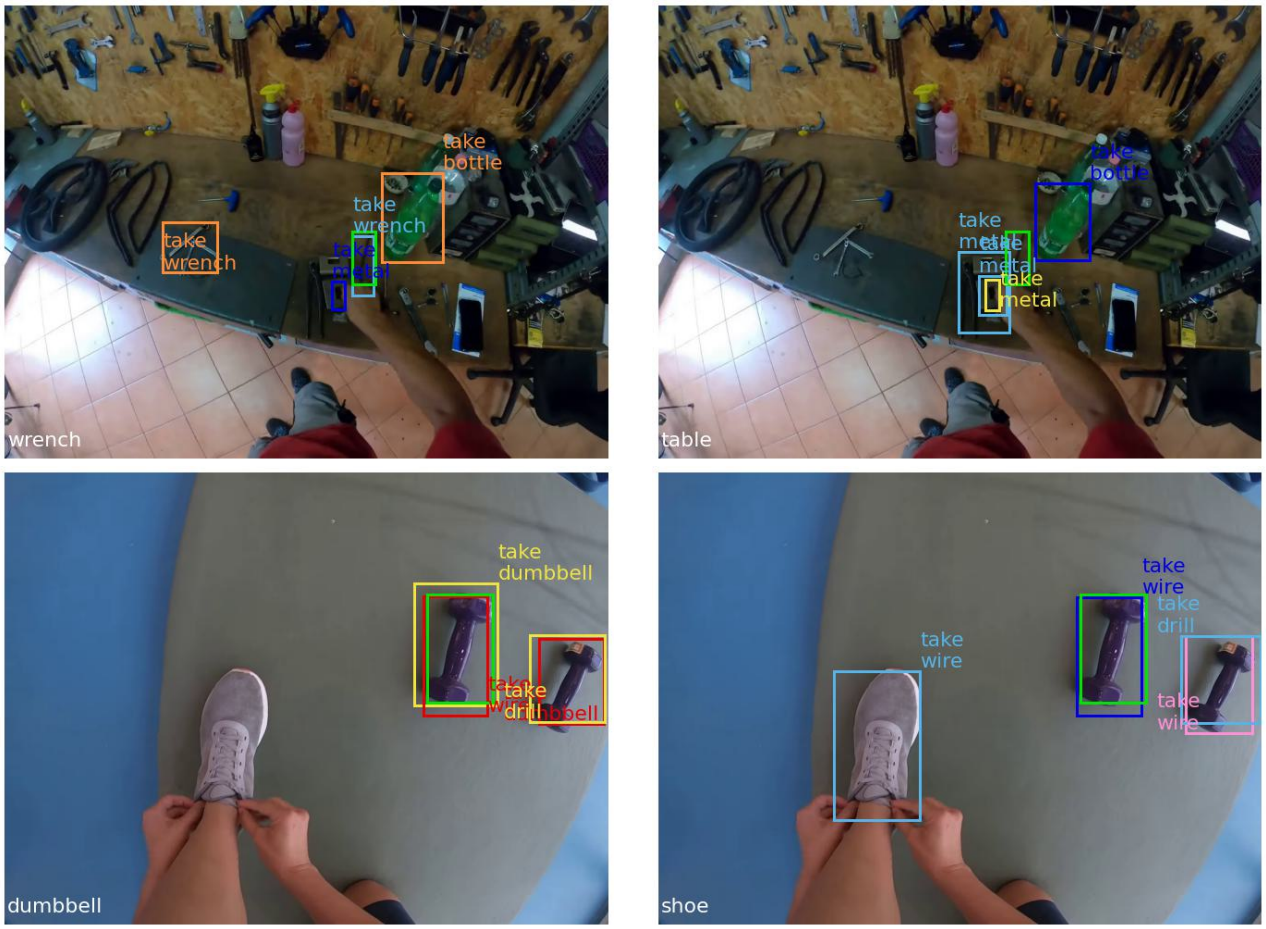

Figure 11. **Examples from the counterfactual analysis experiment.** We show the changes in our model's predictions when altering the language input from *wrench* to *table* on the top row, and *dumbbell* to *shoe* on the bottom row. Additional, similar visualizations are available in Figure 16.

that the verb label prediction is more dependent on understanding the scene and action context in a holistic manner.

7. Architecture ablation

Fusion module ablation. We further ablate the TransFusion design choices and investigate whether the model can benefit from feature reuse at the different scale levels. First, we investigate the effect of sharing the Transformer encoder parameters over the multiscale fusion levels. The new setup leads to a decrease in performance as can be seen in Table 17. We further experiment with forwarding the fused language tokens to the latter stages, with and without residual connections, as an alternative way of sharing fused features. The base TransFusion implementation simply copies the encoded language features at each fusion level. The results are presented in Table 18 where we notice the superior results obtained with the "copy" strategy as opposed to reusing features from previous scale levels. These suggest that the model requires different representations at different feature map scales for effective prediction.

Parameter sharing	# Parameters	L2	N ↑	N-V ↑
✗	122 mln	2e-4	**20.19**	**7.55**
✓	171 mln	3e-5	19.18	6.94

Table 17. **Multiscale fusion parameter sharing.**

Forwarding strategy	Copy	Simple	Residual
N ↑	**20.19**	18.97	18.38
N-V ↑	**7.55**	7.27	6.96

Table 18. **Language feature forwarding strategies.**

7.1. Learning scale-specific features

Shared multiscale fusion weights. An alternative take to combat overfitting is to share model weights over multiple similar inputs, such that they are forced to learn more general data representations. We apply this principle by resharing the transformer fusion weights over the multiple input

scales. Because this setting imposes additional constraints on the fusion weights, we increase the fusion module's capacity by 33% (such that it still fits on one of our GPUs) and reduce the L2 weight decay to 3e-5 to allow more optimization freedom. Using this setup, we register a notable decrease in performance. We perform multiple runs, but do not manage to score more than 7 MAP Box-Noun-Verb and 19.2 MAP Box-Noun on the validation set (approximately .5 and 1 absolute point difference). This indicates that using shared fusion weights at multiple scales is counterproductive for our task and that the model learns different representations at different feature map scales, all needed for effective prediction. The other hyperparameters are kept fixed as in section 3, with a context length of 3 used for $\mathcal{A}$.

7.2. Language modeling loss

We also investigate including an additional language model loss to further accelerate the action description learning. Specifically, we include an additional loss term, $\mathcal{L}_{lm} = \frac{1}{2}(\mathcal{L}_{lm_N} + \mathcal{L}_{lm_s})$ in the final optimization objective

$$\mathcal{L} = \mathcal{L}_{box} + \mathcal{L}_{noun} + \mathcal{L}_{verb} + \mathcal{L}_{ttc} + \mathcal{L}_{lm} \quad (13)$$

The target categories for this loss are the ground-truth noun and verb labels; $\mathcal{L}_{lm_N}$ and $\mathcal{L}_{lm_s}$ are regular cross-entropy losses. The difference between $\mathcal{L}_{noun}$, $\mathcal{L}_{verb}$ and $\mathcal{L}_{lm}$ is that the latter term is applied on the mean-pooled *fused language tokens* at each multiscale fusion level (i.e. the transformer encoder outputs). The former ones work on the ROI-pooled bounding-box features in the Faster R-CNN prediction heads. The experimental results however did not show any improvement when including $\mathcal{L}_{lm}$, which instead decreased the final Box-Noun-Verb performance. We believe that this task in itself can be quite challenging. Without considerably increasing the model capacity, performing it concomitantly with the detection-based objectives can negatively impact the target model's performance.

8. Video features and context length

8.1. Computational cost analysis

Figure 12 illustrates the inference computational cost of TransFusion using action context summaries and TransFusion-Video and their marginal difference. Given a video clip of one second, language can summarize it in two words whereas corresponding video features take up more than 12 times the space (e.g. smaller SBERT feature of 2×384 compared to SlowFast feature of 4×2304). Besides, the Ego4D 2nd stage SlowFast model has 33 million parameters and is trained end-to-end while the small SBERT encoder has 22 million parameters out of which we finetune only 1.7 million. for L_c=3, the TransFusion model has 122M trainable parameters, 11777 GFLOPs and an inference latency of 457 ms. TransFusion-Video has 124M parameters, 11697 GFLOPs and an inference latency of 433 ms. Training costs reveal a similar picture, with 2000 GFLOPs and 250ms latency increase for both models.

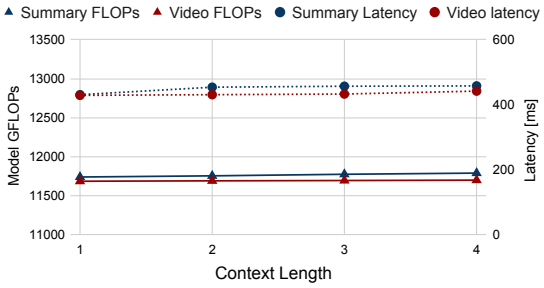

Figure 12. **Computational cost comparison to video features.** Comparison between TransFusion using context description summaries (Summary) and TransFusion-Video (Video). The differences between the two methods are negligible and they scale similarly with the input context length.

8.2. Additional comparison details

To further confirm our findings, we perform an additional run with a video context length of 6, which covers more than 100 frames before the prediction moment. We find that this does not bring any additional performance boost compared to the runs presented in Figure 5 from the main section. It registers a top-5 Box-Noun mAP performance of 18.83 (more than 1 point below the average summary-based run) and 6.40 Box-Noun-Verb mAP (also about 1 point lower). This generally agrees with results presented by the Ego4D work [7] in their Table 39, where the improvements from increasing the number of SlowFast clips suffer slightly diminishing returns. While this work is not an exhaustive comparison of different types of video feature extractors, SlowFast features are still considered effective enough to be used for the latest works, including the Ego4D 2nd-stage, hence we consider them a relevant baseline. Finally, when contrasted with the language encoder run without finetuning presented in Table 3, the difference in Box-Noun-Verb mAP is 0.8 points (or about 12%). This indicates the benefit of using language descriptions even when not finetuning the language encoder. To perform the context length comparison, we use the default parameters presented in section 3.

9. Qualitative results

We present in Figure 13 and Figures 14 & 15 a qualitative comparison between our method and that of Ego4D, using a context length of 3 for action verb-noun pairs $\mathcal{A}$ and 3 salient objects $\mathcal{N}_s$ for our language-aided model. The green bounding box represents the ground truth location of the next object interaction. The white text represents the input action summary context description: the first row represents the salient objects, and the second the action descriptions.

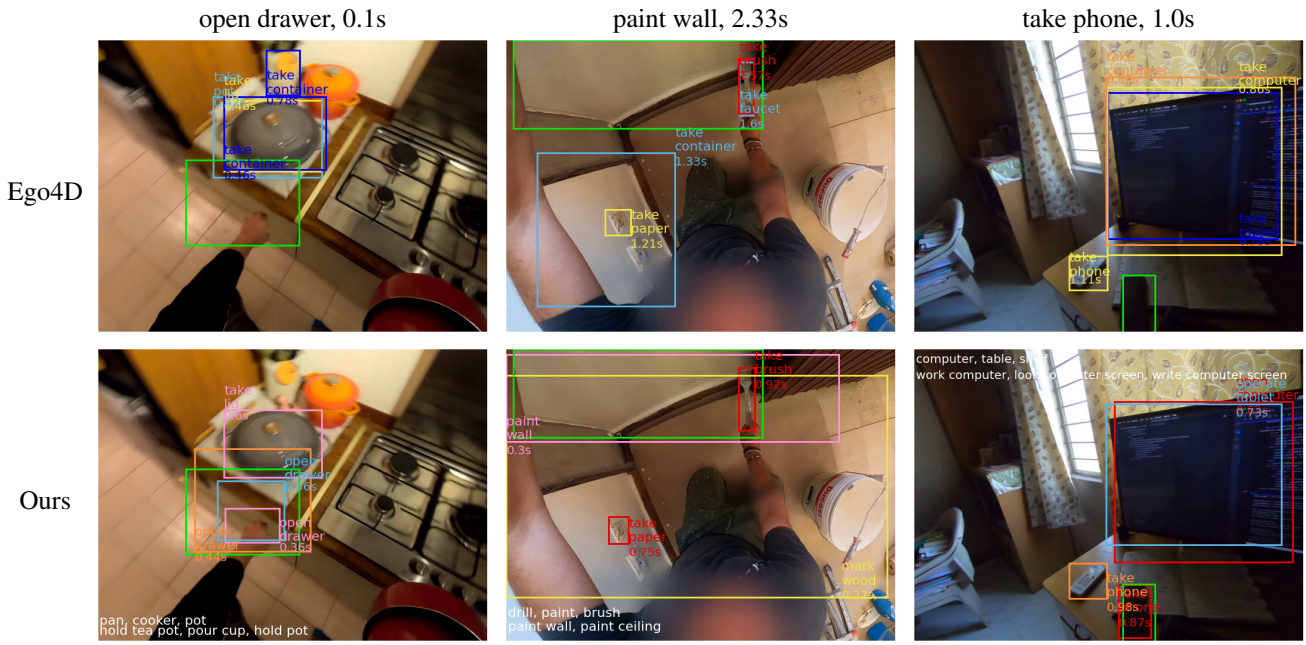

Figure 13. **Qualitative examples of Ego4D and TransFusion (ours) predictions**. The ground-truth action label and TTC is represented on top of each column. The bright green bounding boxes denote the ground-truth location of the next object interaction. Action contexts used as language input in TransFusion are shown at the bottom in white. On average, our model manages to get more accurate predictions.

We show the top-4 most confident bounding box predictions and the associated noun, verb, and TTC estimates.

References

[1] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual String Embeddings for Sequence Labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649, 2018. 1

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization, 2016. 4

[3] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing With Python: Analyzing Text With the Natural Language Toolkit*. " O'Reilly Media, Inc.", 2009. 1

[4] Guo Chen et al. InternVideo-Ego4D: A Pack of Champion Solutions to Ego4D Challenges. *arXiv preprint arXiv:2211.09529*, 2022. 5

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling Egocentric Vision. *CoRR*, abs/2006.13256, 2020. 6

[6] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. EPIC-KITCHENS VISOR Benchmark: VIdeo Segmentations and Object Relations. In *Proceedings of the Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2022. 1

[7] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4D: Around the World in 3,000 Hours of Egocentric Video. *CoRR*, abs/2110.07058, 2021. 5, 10

[8] Dan Hendrycks and Kevin Gimpel. Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. *CoRR*, abs/1606.08415, 2016. 4

[9] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft

COCO: Common Objects in Context. *CoRR*, abs/1405.0312, 2014. 1

[10] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, 2014. 3

[11] Francesco Ragusa et al. StillFast: An End-to-End Approach for Short-Term Object Interaction Anticipation. *arXiv preprint arXiv:2304.03959*, 2023. 5

[12] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks From Overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 4

[13] Sanket Thakur, Cigdem Beyan, Pietro Morerio, Vittorio Murino, and Alessio Del Bue. Enhancing Next Active Object-Based Egocentric Action Anticipation With Guided Attention. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 1450–1454. IEEE, 2023. 5

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *CoRR*, abs/1706.03762, 2017. 4

[15] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework. *CoRR*, abs/2202.03052, 2022. 1

[16] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple Multi-Dataset Detection. *CoRR*, abs/2102.13086, 2021. 1

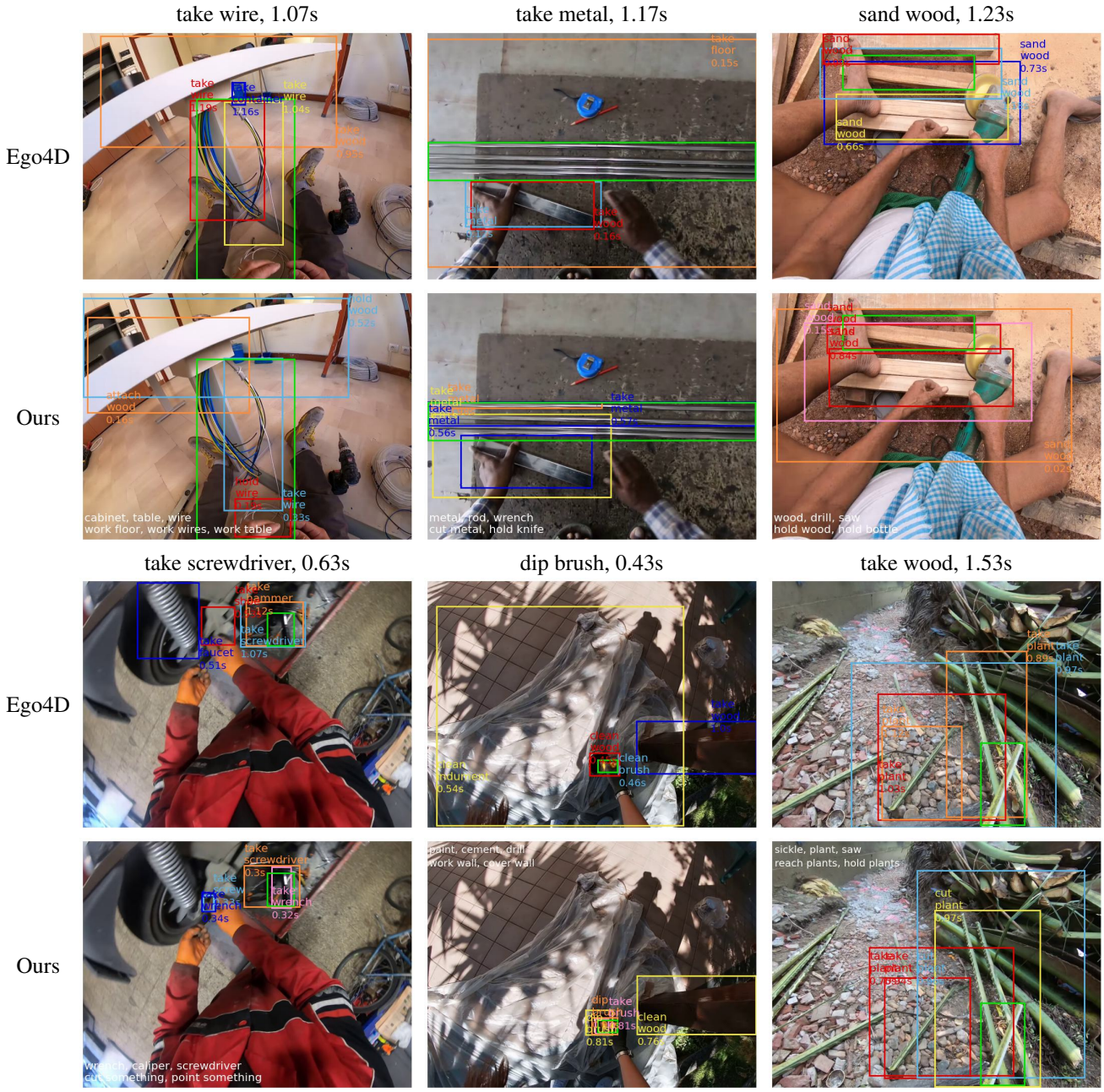

Figure 14. **Additional qualitative examples of Ego4D and TransFusion (ours) predictions (I)**. The ground-truth action description and TTC are represented at top of each column. The bright green bounding boxes denote the ground-truth location of the next object interaction. Context summaries used in TransFusion are shown at the bottom in white. On average, our model manages to get more accurate predictions. More examples are shown in Figure 15.

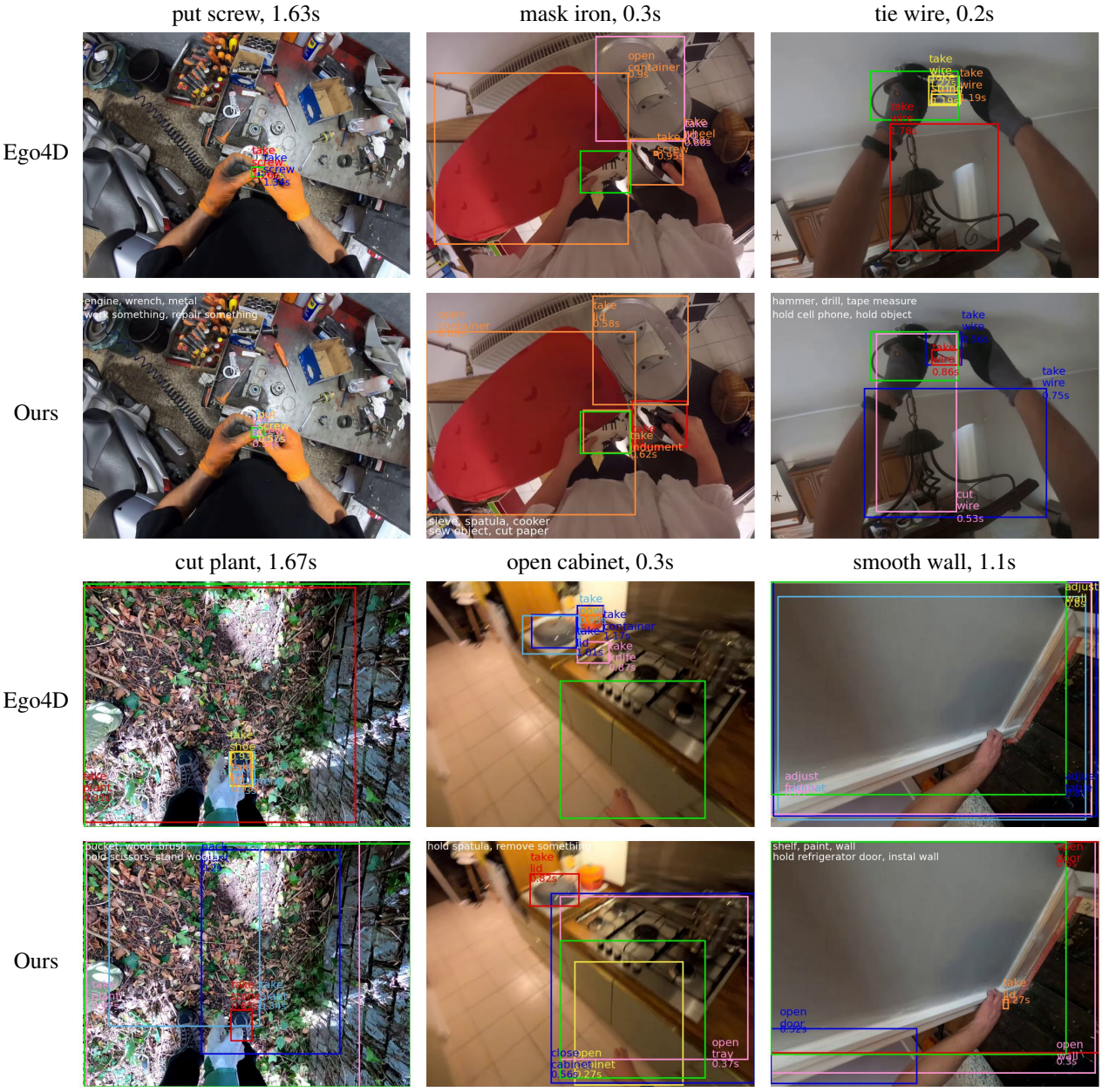

Figure 15. **Additional qualitative examples of Ego4D and TransFusion (ours) predictions (II)**. See Figure 14 for details.

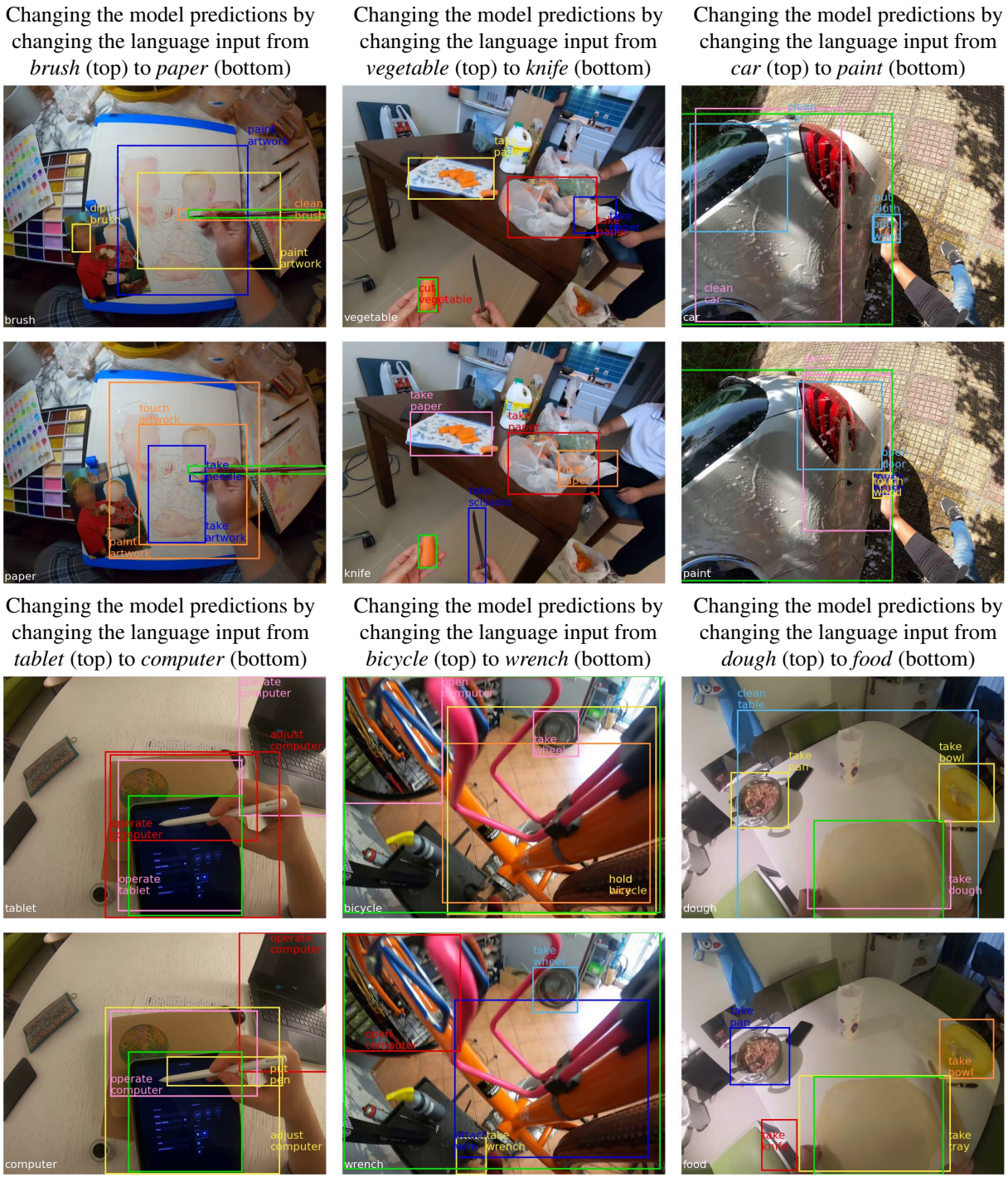

Figure 16. **Additional qualitative examples of predictions when changing the language input**. Our model modifies the predicted labels and locations *dynamically* based on the input language context descriptions. We show the input language context in white, in the bottom left corner of each image.

Figure 17. **Successful detections of held objects for $\mathcal{N}_h$ construction.** We show the ground-truth next-active object in red, hand bounding boxes in purple, active object bounding boxes in yellow, the bounding boxes of the inferred held objects, selected from those detected by the object detector and having a sufficiently high IoU with an active object bounding box from the hand interaction model, in blue, and the extracted held object context $\mathcal{N}_h^F$ in the bottom left corner of each image.

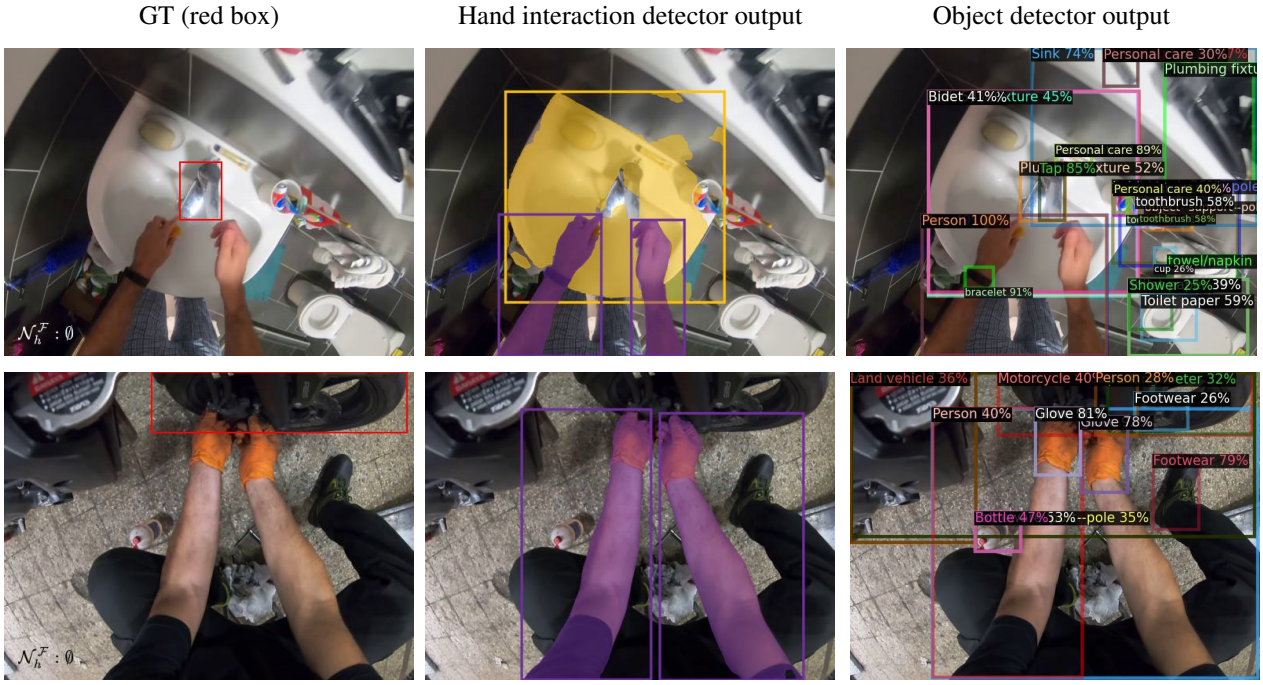

Figure 18. **Failures of hand-object interaction detector during $\mathcal{N}_h$ construction**. The hand-object interaction detector produces an oversegmentation (top) or does not detect the object (bottom). See Figure 17 for details.

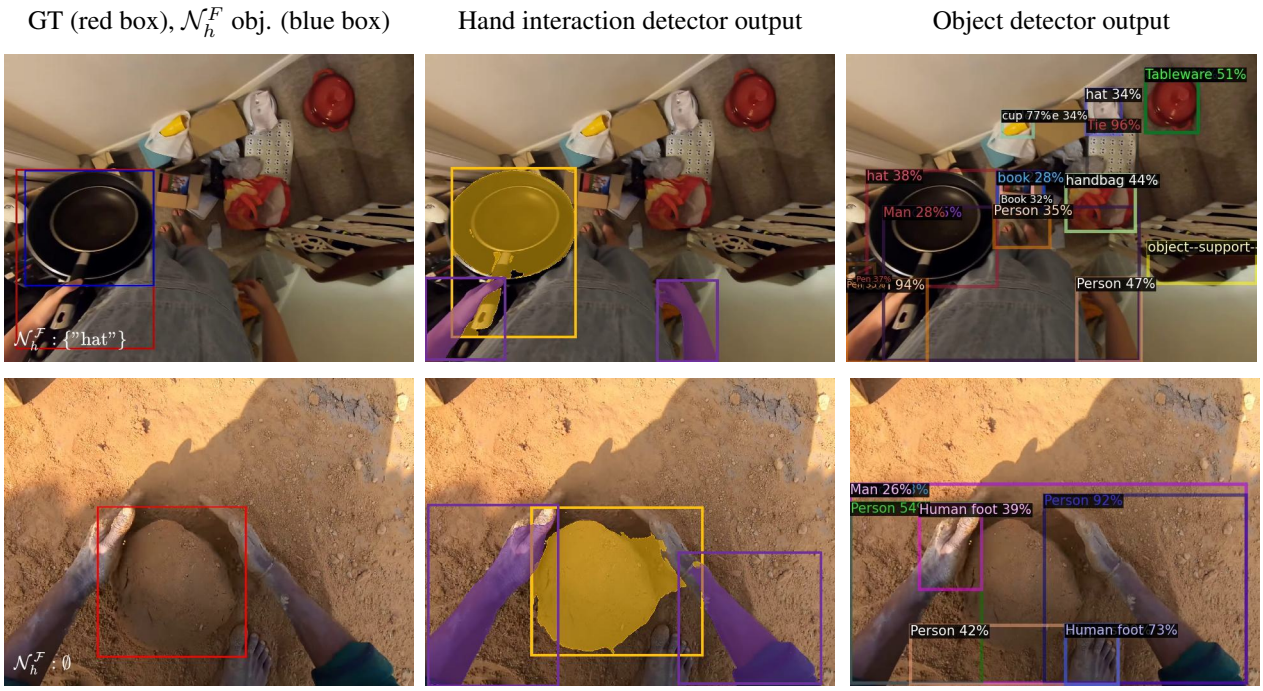

Figure 19. **Failures of object detector during $\mathcal{N}_h$ construction**. The object is misclassified (top) or not detected (bottom). See Figure 17 for details.