UNDERSTANDING EMBODIED REFERENCE WITH TOUCH-LINE TRANSFORMER

Anonymous authors
Paper under double-blind review

ABSTRACT

We study embodied reference understanding, the task of locating referents using embodied gestural signals and language references. Human studies have revealed that objects referred to or pointed to do not lie on the *elbow-wrist line*, a common misconception; instead, they lie on the so-called *virtual touch line*. However, existing human pose representations fail to incorporate the virtual touch line. To tackle this problem, we devise the touch-line transformer: It takes as input tokenized visual and textual features and simultaneously predicts the referent's bounding box and a touch-line vector. Leveraging this touch-line prior, we further devise a geometric consistency loss that encourages the co-linearity between referents and touch lines. Using the touch-line as gestural information improves model performances significantly. Experiments on the *YouRefIt* dataset show our method achieves a +25.0% accuracy improvement under the 0.75 IoU criterion, closing 63.6% of the gap between model and human performances. Furthermore, we computationally verify prior human studies by showing that computational models more accurately locate referents when using the *virtual touch line* than when using the *elbow-wrist line*.

1 INTRODUCTION

Understanding human intents is critical to intelligent robots when interacting with humans. However, most prior work in the modern learning community neglects the multimodal aspect in human-robot communication scenarios. Let us take an example shown in Fig. 1, wherein a person commands the robot to interact with a chair behind the table. In response, the robot ought to understand what is being referred to before taking any actions (*e.g.*, moving towards it and cleaning it). Notably, both **embodied** gesture signals and language **reference** play significant roles. Without the pointing gesture, the robot would not be able to distinguish the two chairs using the utterance *the chair that is occluded*. Without the language expression, the robot would not be able to distinguish the chair from other objects in that region (*e.g.*, bags on the table). To fill this gap, we study the Embodied Reference Understanding (ERU) task introduced by Chen et al. (2021). This task requires an algorithm to detect the referent (referred object) using (i) an image/video with nonverbal communication signals and (ii) a sentence as a verbal communication signal.

Figure 1: **Both nonverbal and verbal expressions are necessary to properly locate the referent in complex scenes.** Without nonverbal expression (the pointing gesture in this case), the verbal expression ("the chair") cannot uniquely refer to *the* chair because multiple chairs are present in this scenario. If only nonverbal expression were used, one cannot distinguish the intended referent "the chair" from other objects in that region.

The **first** fundamental issue in addressing ERU is the representation of the human pose. The *de facto* human pose representation in modern computer vision is defined by COCO (Lin et al., 2014)—a graph consisting of 17 nodes (keypoints) and 14 edges (keypoint connectivity). This representation is prevalent; The existing model for ERU (Chen et al., 2021) take for granted to use pre-extracted COCO-style pose features as the algorithm input. However, we rethink the limitation of the COCO-

style pose graph in the context of ERU and identify a counter-intuitive fact: The referent does not lie on the elbow-wrist line (*i.e.*, the line that links the human elbow and wrist). As shown in Fig. 2, this line (in red) does not cross the referred microwave, illustrating a common misinterpretation of human pointing (Herbort & Kunde, 2018).

Interestingly, a recent psychology study (O'Madagain et al., 2019) provides strong evidence that supports the above hypothesis from the behavior perspective. It investigates how human beings mentally develop pointing gestures and argues that it is a virtual form of *reaching out to touch*. This new finding challenges conventional psychological theories (McGinn, 1981; Kita, 2003) that the pointing gesture is mentally a behavior of using the limb as an arrow. Inherited from the notations in O'Madagain et al. (2019), we coin the red line in Fig. 2 as an EWL and the yellow line in Fig. 2 (*i.e.*, a line that connects the eye and the fingertip) as a VTL. Inspired by this essential observation that VTLs are more accurate than EWLs in embodied reference, we augment the existing COCO-style pose graph with an edge that connects the eye and the fingertip. As validated in a series of experiments in Sec. 4, this augmentation brings significant performance boosts on *YouRefIt* (Chen et al., 2021).

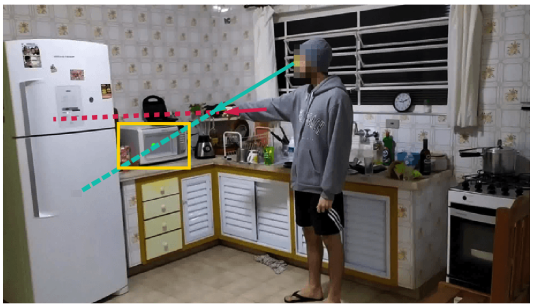


Figure 2: **Virtual Touch Line (VTL) (in green) vs. Elbow-Wrist Line (EWL) (in red).** VTLs affords a more accurate location of referents than EWLs.

The **second** fundamental issue in addressing ERU is how to model gestural signals and language references jointly. Inspired by the success of multimodal transformers (Chen et al., 2020; Li et al., 2020; Tan & Bansal, 2019; Lu et al., 2019; Kamath et al., 2021) in multimodal tasks (Hudson & Manning, 2019; Antol et al., 2015; Zellers et al., 2019), we devise the Touch-Line Transformer, a transformer-based model that takes inputs from both the visual and natural-language modalities. Our Touch-Line Transformer jointly models gestural signals and language references by simultaneously predicting the touch-line vector and the referent's bounding box. To further help our model utilize gestural signals (*i.e.*, the touch-line vector), we integrate a geometric consistency loss to encourage co-linearity between the touch-line and the predicted location of the referent, resulting in significant performance improvements.

Leveraging the above two insights, our proposed method achieves a +25.0% accuracy improvement under the 0.75 IoU criterion compared with state-of-the-art methods on the *YouRefIt* dataset, closing 63.6% of the gap between model performance and human performance.

This paper makes four contributions. (i) Inspired by recent behavior studies, we devise a novel human pose representation–Virtual Touch Line (VTL)–in computer vision. (ii) We introduce the Touch-Line Transformer that jointly models nonverbal gestural signals and verbal references. (iii) We encourage our model to utilize the touch line by a novel geometric consistency loss that improves the co-linearity between the touch line and the predicted object. (iv) Our model achieves a new state-of-the-art performance on ERU, attaining a +25.0% performance gain under the 0.75 IoU threshold.

2 Related Work

Misinterpretation of pointing gestures Pointing allows observers and pointers to direct visual attention to establish references in communication. Recent research reveals that observers surprisingly make systematic errors (Herbort & Kunde, 2016); they found that, while the pointer produces gesture using VTLs, observers interpret pointing gestures using the "arm–finger" line. The VTL mechanism is founded by O'Madagain et al. (2019): Pointing gestures orient toward their targets as if the pointers were to touch these targets. In neuroscience, gaze effects occur for tasks that require gaze alignment with finger pointing (Bédard et al., 2008). Together, the above evidence shows that eye position and gaze direction are significant for understanding pointing. Critically, Herbort & Kunde (2018) verify that when human observers were instructed to extrapolate the touch-line vector, this systematic misinterpretation during human-human communication is reduced. Inspired by these discoveries, we integrate the touch-line vector to improve the pointing gesture interpretation performance.

Gaze target detection The gazing gesture itself can refer to some objects in the absence of language expressions. Gaze target detection is the localization of the objects that people are gazing at. Recasens et al. (2015) and Chong et al. (2018) proposed two-state methods that generate saliency maps and estimate gaze directions first before combining them and predicting the gaze target. Tu et al. (2022) designed a one-stage end-to-end method to simultaneously locate human heads and gaze targets for all people in an image. Fang et al. (2021) designed a three-stage method that predicts 2D and 3D gaze direction, predicts field of views and depth ranges, and then predicts the gaze target. Zhao et al. (2020) maintains that humans find gaze targets from salient objects on the sight line. They designed a model to simulate the hypothesized human way of finding gaze targets in images and extended its ability to video frames. Li et al. (2021) extended gaze target detection to 360-degree images. Chong et al. (2020) and Recasens et al. (2017) extended gaze target detection into videos, where the gazed object may not be in the same frame as the gazing gesture.

Referring expression comprehension Language expressions can also refer to some objects. Referring expression comprehension is to locate referents in an image using only language expressions. Multiple datasets (*e.g.*, RefCOCO (Yu et al., 2016), RefCOCO+ (Yu et al., 2016), RefCOCO-g (Mao et al., 2016), and Guesswhat (De Vries et al., 2017)) help benchmark models for this task. On methods and models, Du et al. (2021) devise a transformer-based framework and use the language to guide the extraction of "discriminative visual features" when encoding images and sentences. Rohrbach et al. (2016) recognize the bi-directional nature between language expressions and objects in images, obviating the need for bounding box annotations. Yu et al. (2018) devise a modular network; three modules respectively attend to subjects, their location in images, and their relationships with nearby objects. While the above works locate referents using language expressions, they rarely use nonverbal gestures produced by humans in the input images.

Language-conditioned imitation learning Imitation helps robots to learn skills (Schaal, 1999), and robots need to identify target objects when completing goal-conditioned tasks (Stepputtis et al., 2020). Stepputtis et al. (2020) argue that specifying the target object via vectors and one-hot vectors, while helpful for robots to identify the target object, is not flexible enough to support continued learning in deployed robots. As a result, Stepputtis et al. (2020) propose to use the more flexible natural language to refer target objects to robots for goal-conditioned manipulation tasks. Specifically, their architecture contains a semantic module to help robots understand referents referred to by languages. Their architecture helps robots to locate and attend to the target object when the natural language expression non-ambiguously refers to an object. However, their approach does not consider pointing gestures and has difficulties in handling ambiguities when referring involves pointing gestures. As a result, some recent works in language-conditioned imitation learning (Stepputtis et al., 2020; Lynch & Sermanet, 2021; Mees et al., 2022) will potentially benefit from our proposed VTL by more accurately locating the target object, especially when the natural language alone is ambiguous.

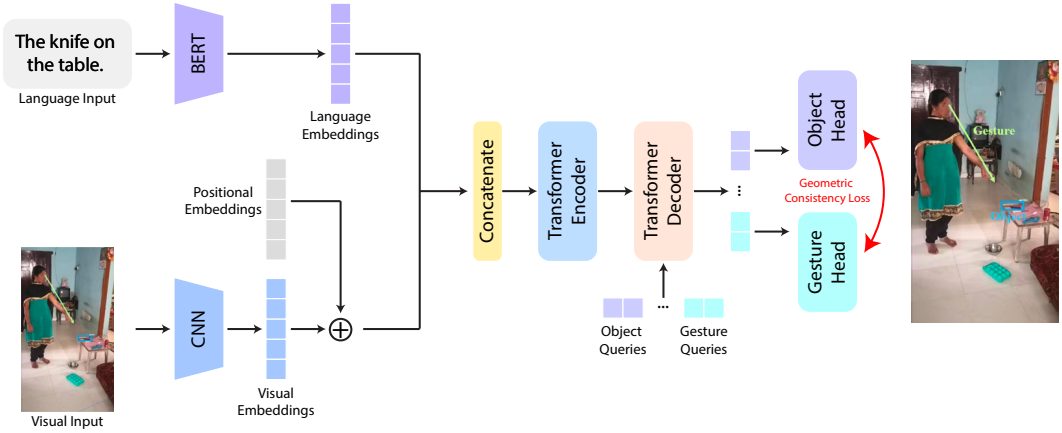


Figure 3: **Overall network architecture.** Language and visual inputs are first encoded by the text encoder and visual encoder to obtain language and visual embeddings, respectively. Next, these embeddings are concatenated and fed into the transformer encoder to learn multimodal representations. The transformer decoder and prediction heads output the predicted bounding box and VTL/EWL. A geometric consistency loss is integrated to encourage the use of gestural signals.

3 METHOD

3.1 NETWORK ARCHITECTURE

Our framework, shown in Fig. 3, consists of a multi-modal encoder, a transformer decoder, and prediction heads. We detail each component below.

Multimodal encoder We generate a visual embedding vector by extracting visual features from input images with a pre-trained ResNet (He et al., 2016) backbone, flattening them, and adding them to a set of position embeddings (Parmar et al., 2018; Bello et al., 2019). Meanwhile, we generate a textural embedding vector from input texts using a pre-trained BERT (Liu et al., 2019). After obtaining visual and textural embedding vectors, we concatenate and feed them into a transformer encoder to learn multi-modal representations.

Transformer decoder We feed multi-modal representations generated above to our transformer decoder. Our transformer decoder, additionally, takes a set of learnable object queries and gestural key point queries as inputs. With multi-modal representations and input queries, our transformer decoder generates object output embeddings and gestural output embeddings. We use the pre-trained MDETR (Kamath et al., 2021) decoder.

Prediction heads Object and gestural output embeddings from the transformer decoder are inputs for our prediction heads (MLPs). Our prediction heads use them to predict bounding boxes (for referents) and gestural key points. We keep one object bounding box and one pair of gestural key points with the highest scores as the final prediction. Specifically, we define the score of a bounding box as 1 minus the non-object class column of the softmax of the predicted object logits. For a pair of gestural key points, the score is the is-a-VTL/EWL column of the predicted arm logits' softmax.

3.2 EXPLICIT LEARNING OF NONVERBAL GESTURAL SIGNALS

Measure co-lineaerity Ideally, a referent should have a high co-linearity with the VTL. We measure this co-linearity using cosine similarity:

$$cos_sim = \text{CosineSimilarity}[(x_f - x_e, y_f - y_e),\ (x_o - x_e, y_o - y_e)], \tag{1}$$

where (x_f, y_f), (x_e, y_e), and (x_o, y_o) are the x and y coordinates of the fingertip, the eye, and the center of referent bounding box, respectively.

Encourage co-lineaerity We encourage our model to predict referent bounding boxes that are highly co-linear with VTLs using a referent alignment loss, which is a geometric consistency loss:

$$L_{aligned} = \text{ReLU}(cos_sim_{gt} - cos_sim_{pred}), \tag{2}$$

where cos_sim_{gt} is computed using ground truth referent box and cos_sim_{pred} is computed using predicted referent box. In both cos_sim_{gt} and cos_sim_{pred}, ground truth VTLs are used. Of note, the design of the offset cos_sim_{gt} ensures the predicted object is as co-linear as the ground-truth object to the VTL. In other words, the loss in Eq. (2) is minimized to zero when the predicted object box is as co-linear as the ground-truth one to the VTL.

Modify for EWLs In experiments using EWLs instead of VTLs, we replace the fingertip (x_f, y_f) and the eye (x_e, y_e) in Eq. (1) using the wrist (x_w, y_w) and the elbow (x_l, y_l), respectively.

3.3 IMPLICIT LEARNING OF NONVERBAL GESTURAL SIGNALS

We eliminate postural signals (by removing humans) from images to investigate whether our model can implicitly learn to utilize useful nonverbal gestural signals without being explicitly asked to learn. Please refer to Appendix A.1 for details of our procedure.

3.4 ADDITIONAL LOSSES

The total loss in training is defined as:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{box}} + \lambda_2 \mathcal{L}_{\text{gesture}} + \lambda_3 \mathcal{L}_{\text{align}} + \lambda_4 \mathcal{L}_{\text{token}} + \lambda_5 \mathcal{L}_{\text{contrastive}}, \tag{3}$$

where λ_is are the weights of various losses, $\mathcal{L}_{\text{box}}$ is the weighted sum of the L1 and GIoU losses for predicted referent bounding boxes, and $\mathcal{L}_{\text{gesture}}$ is the L1 loss for predicted VTLs or EWLs. The soft token loss $\mathcal{L}_{\text{token}}$ and the contrastive loss $\mathcal{L}_{\text{contrastive}}$ follow those in Kamath et al. (2021); they help our model align visual and textural signals.

4 EXPERIMENTS

4.1 DATASET, EVALUATION METRIC, AND IMPLEMENTATION DETAILS

We use the *YouRefIt* dataset (Chen et al., 2021), consisting of $2,950$ training instances and $1,245$ test instances. It has two portions: videos and images. We use the image portion; the inputs to our model are images that do not have temporal information. Each instance contains an image, a sentence, and the location of the referent. We provide additional annotations of VTLs and EWLs for the *YouRefIt* dataset.

To make a fair comparison with prior models, we follow Chen et al. (2021) and report precision under three different IoU thresholds: 0.25, 0.50, and 0.75. A prediction is correct if its IoU with the ground-truth box is greater than the threshold. Additionally, we choose our best models by the precision at the GIoU (Rezatofighi et al., 2019) threshold of 0.75; compared with IoU, GIoU is an improved indicator of the model's ability to locate objects accurately. We report the results using GIoU thresholds in Sec. 4.5.

During training, we use the Adam optimizer's AMSGrad variant (Reddi et al., 2018) and train our models for 200 epochs. We use the AMSGrad variant because we observe a slow convergence of the standard Adam optimizer in experiments. We set the learning rate to 5e-5 except for the text encoder, whose learning rate is 1e-4. We do not perform learning rate drops because we rarely observe demonstrable performance improvements after dropping them. We use A100 GPUs. The sum of the batch sizes on all graphic cards is 56. Augmentations follow those in Kamath et al. (2021). The total number of queries is 20 (15 for objects; 5 for gestural key points).

4.2 COMPARISON WITH STATE-OF-THE-ART METHODS

Our approach outperforms prior state-of-the-art methods by 16.4%, 23.0%, and 25.0% under IoU threshold of 0.25, 0.50, and 0.75, respectively; see Tab. 1. Specifically, our model performs better than visual grounding methods (Yang et al., 2019; 2020) which do not explicitly utilize nonverbal gestural signals. Our method also performs better than the method proposed in *YouRefIt* (Chen et al., 2021), which did not leverage the touch line or the transformer models for multimodal tasks.

4.3 EXPLICITLY LEARNED NONVERBAL SIGNALS

We report the model performance that explicitly predicts either the VTLs or the EWLs.

Table 1: **Comparison with the state-of-the-art methods.**

	IoU=.25	IoU=.50	IoU=.75
FAOA (Yang et al., 2019)	44.5	30.4	8.5
ReSC (Yang et al., 2020)	49.2	34.9	10.5
YouRefIT PAF-only (Chen et al., 2021)	52.6	37.6	12.7
YouRefIt Full (Chen et al., 2021)	54.7	40.5	14.0
Ours (Inpainting)	59.1 (+4.4)	51.3 (+10.8)	32.4 (+18.4)
Ours (No Explicit Gestural Key Points)	64.9 (+10.2)	57.4 (+16.9)	37.2 (+23.2)
Ours (EWL)	69.5 (+14.8)	60.7 (+20.2)	35.5 (+21.5)
Ours (VTL)	71.1 (+16.4)	63.5 (+23.0)	39.0 (+25.0)
Human	94.2	85.8	53.3

Results Overall, the model trained to predict the VTLs performs better than the one trained to predict the EWLs under all three IoU thresholds; see Tab. 2. The model trained to explicitly predict the EWLs performs even worse than the one that was not trained to explicitly predict any gestural signals under the IoU threshold of 0.75.

Analysis Under the IoU threshold of 0.75, the worse performance of the model with explicit EWL prediction can be partly attributed to the unreliability of the arm's orientation and the stringent precision requirement of the 0.75 IoU threshold. As we observed in Fig. 4, the EWLs are not reliable for predicting objects' locations, simply because they oftentimes do not pass through the referents. This mismatch prevents the model from correctly determining the referents' locations using the EWLs' orientations. For instance, in Fig. 4a (right), the EWL fails to pass the glass cup in the yellow

Figure 4: **The model explicitly trained to predict VTLs more accurately locates referents than the one trained to predict EWLs.** We draw green arrows (different from the predictions) to illustrate eyes and fingertips more accurately indicate object locations. (Green sentence: natural language inputs; Green box: ground-truth referent location; Red box: predicted referent location; Red numbers near predicted box: GIoU between predicted box and ground truth box.)

box; it passes a ceramic cup instead. With contradictory nonverbal (the ceramic cup) and verbal (the glass cup) signals, the model struggles to identify the referent correctly.

In contrast, VTLs partly explain the model's improved performance. For example in Fig. 4b, the VTL passes the chair in the yellow box while the EWL fails. Similarly, in Fig. 4c, 4e, 4f, 4h and 4l, VTLs passes the referent while EWLs do not. Additionally in Fig. 4a, 4d, 4g and 4i to 4k, VTLs are closer to the referents' box centers than EWL are. The higher consistency between verbal and nonverbal signals increases the chance of successfully identifying the referents.

Table 2: **Effects of learning two different types of postural key points.**

IoU	None	EWL	VTL
0.25	64.9	69.5 (+4.6)	71.1 (+6.2)
0.50	57.4	60.7 (+3.3)	63.5 (+6.1)
0.75	37.2	35.5 (-1.7)	39.0 (+1.8)

Under the IoU thresholds of 0.25 and 0.50, the improved model performance can be partly attributed to the rough orientations provided by the EWLs and the more lenient precision requirements of the lower thresholds. Specifically, the rough orientations provided by the EWLs might help the model eliminate objects that significantly deviate from this direction. Hence, the model can choose from a smaller number of objects, leading to a higher probability of correctly locating the referents.

4.4 Implicitly Learned Nonverbal Signals

We further investigate whether our model can implicitly learn nonverbal gestural signals without being explicitly asked to predict the VTLs or the EWLs.

Table 3: **Effects of implicitly learned nonverbal signals.**

IoU	Original	Inpainting
0.25	64.9	59.1 (-5.8)
0.50	57.4	51.3 (-6.1)
0.75	37.2	32.4 (-4.8)

To investigate this, we conduct two sets of experiments: one using original images and another one using inpainted images. Specifically, when using original images, we do not provide our annotated gestural key points (eyes, fingertips, elbows, and wrists) to our model and we do not ask our model to predict any of these gestural key points. When using the inpainted images in which humans are removed, we still do not provide annotated gestural key points or ask our model to predict these key points.

The set of experiment using original images but does not provide or predict gestural key points corresponds to the "Original" column in Tab. 3, the "No Explicit Gestural Key Points" column in Tab. 1, the "No Exp. Ges. K. Poi." column in Tab. 7, and the "None" column in Tab. 2.

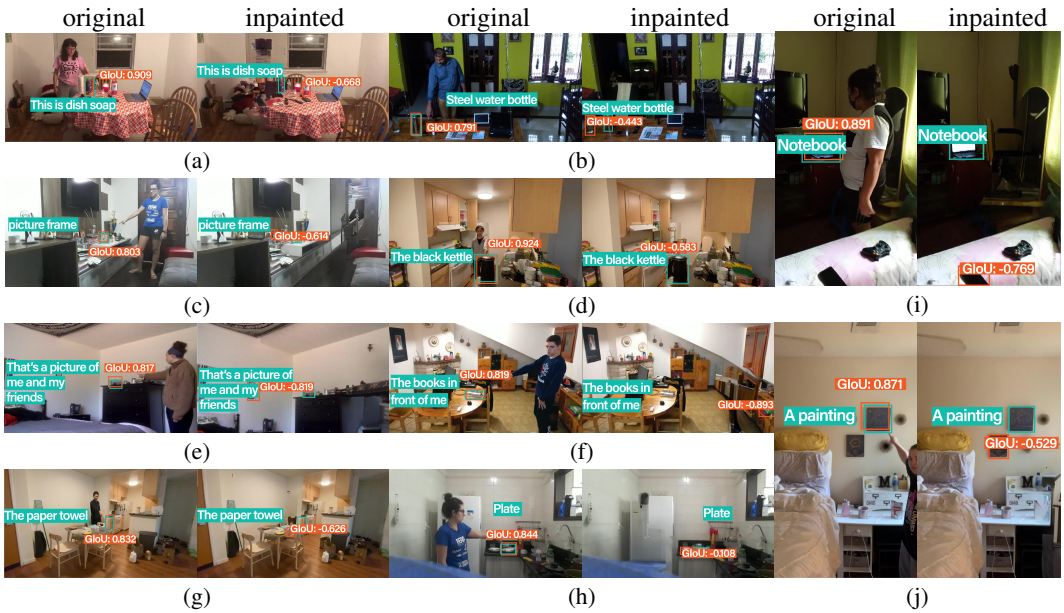


Figure 5: **Training with inpainted images leads to a performance drop.** Removing human contexts leads to scenarios wherein determining which object is referred to by the natural language expression becomes difficult (a, b, c, i) or even impossible (d, e, f, g, h, j). We call it impossible to determine the referent using natural language input when multiple objects in the image satisfy the descriptions in the natural language input sentence. (Green sentence: natural language input; Green box: ground-truth referent location; Red box: predicted referent location; Red numbers near boxes: GIoU between the predicted box and ground truth box.)

Our model performs much worse when nonverbal gestural signals are absent. Specifically, removing humans from input images results in 5.8%, 6.1%, and 4.8% performance drop under the IoU threshold of 0.25, 0.50, and 0.75, respectively; see details in Tab. 3.

These performance drops indicate that the implicitly learned model lacks useful information given only the inpainted images. Specifically, removing humans in the scenes also eliminates useful communicative signals, including pointing, poses, and the gaze direction (Sebanz et al., 2006). The lack of human context introduces additional ambiguities in identifying the referents, especially when the natural language input alone cannot uniquely refer to an object in the scene.

For example in Fig. 5f, given the natural language input "the books in front of me," determining which piles of books this person refers to is difficult: one on the table and one in the scene's lower right corner, thus in need of nonverbal gestural signals, such as pointing. Similarly in Fig. 5e, we cannot distinguish which picture frame the person is referring to without considering the pointing gesture.

4.5 Additional Results

Attention weights visualizations We visualize the attention weights of our model trained with VTL (Fig. 6). We use yellow to visualize the attention weights of matched gesture keypoint queries and blue for matched object queries. Visualizations show that the attention of object queries can attend to the regions of the target objects while the attention of keypoint queries primarily focus on humans' heads and hands. Taken together, these results indicate that our model successfully learns gestural features that boost the model performance.

Failure cases While the VTL effectively helps the model leverage the nonverbal gestural signals, the performance of our model still could not surpass the 74% human performance. The gap between model and human performance is partly attributed to the model's inability to distinguish the subtle differences between similar objects. In other words, while explicitly predicting the VTLs helps our model utilize gestural signals to determine the orientation of the object, it does not help the model recognize the minor differences between multiple objects in the same orientation.

For example, the VTL passes two bottles in Fig. 7a. While the VTL indeed helps to narrow down the options to these two bottles, it does not help our model recognize which one of them is body lotion.

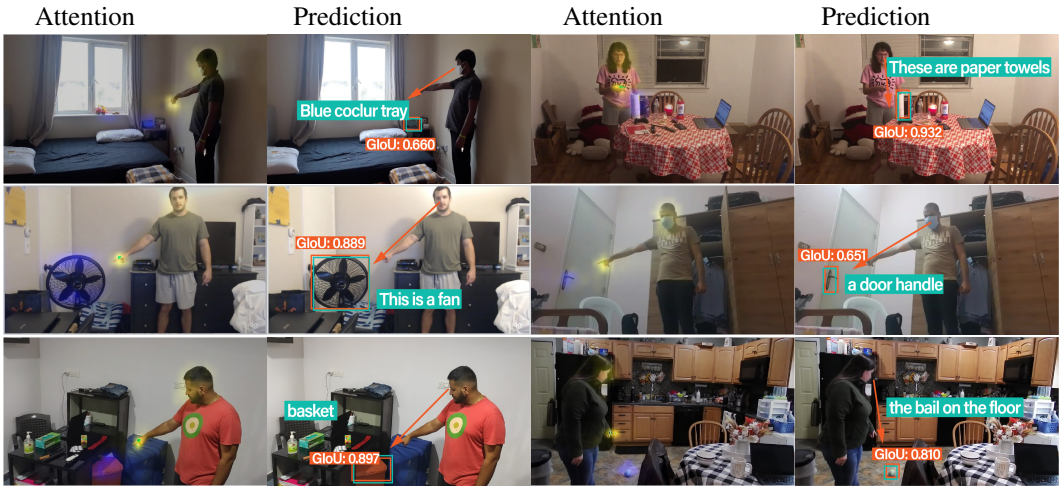


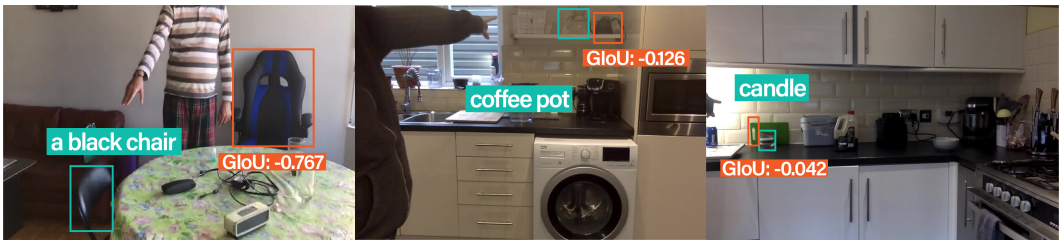
Figure 6: **Attention weights visualizations.** Attention visualization of models trained to explicitly learn VTLs. (Blue: attention from object tokens; Yellow: attention from gestural key point tokens; Green sentence: natural language input; Green box: ground-truth referent location; Red box: predicted referent location; Red numbers near boxes: GIoU between the predicted box and ground truth box.)

Similarly in Fig. 7b, the VTL indicates two bottles—one with a red lid and the other with a yellow lid; yet it does not help the model distinguish the subtle differences between the lid colors.

(a) (b) (c) (d)

Figure 7: **Examples of failure cases.** Green sentence: natural language input; Green box: ground-truth referent location; Red box: predicted referent location; Red numbers near boxes: GIoU between the predicted box and ground truth box.

Additionally, in very rare cases (Fig. 8), the application of our VTL is limited because the human head is not visible from the input image. Without human heads, a touch line cannot be drawn in the image.



(a) (b) (c)

Figure 8: **Rare cases where the human head is not visible.**

Computed cosine similarities The computed cosine similarity is much higher when using the VTLs (Tab. 4), indicating the VTLs are more co-linear to the referents. This result verifies our hypothesis that the VTLs are more reliable in referring to the objects' locations.

Table 4: **Computed cosine similarities.** (tgt: computed using ground-truth box centers; pred: computed using predicted box centers.)

gesture	cos sim tgt	cos sim pred
EWL	0.9580	0.9579
VTL	0.9901	0.9878

Table 5: **Effects of the referent alignment loss $\mathcal{L}_{\text{align}}$.** Removing the reference alignment loss results in performance drops in locating referents.

gesture	ref. align. loss	IoU=.25	IoU=.50	IoU=.75	cos sim pred
VTL	True	71.1	63.5	39.0	0.9878
VTL	False	67.1	59.0	36.4	0.9815
		(-4.0)	(-4.5)	(-2.6)	(-0.0063)

Referent alignment loss An ablation study (Tab. 5) shows that our reference alignment loss plays a significant role in leveraging nonverbal gestural signals.

Effects of object sizes We evaluate the model performance w.r.t. the object detection of different object sizes (Tab. 6). We define small (S), medium (M), and large (L) objects following the size thresholds in Chen et al. (2021). Compared to the relatively consistent human performance, deep learning models' performances are significantly lower when detecting diminutive objects than when detecting larger objects, especially under the less stringent IoU thresholds of 0.25 and 0.50. The degraded performance indicates that future artificial intelligence models need to improve the performance of small object detection.

Table 6: **Model performances w.r.t. the object detection objects of different object sizes.** Red and blue represents the first and the second highest performance, respectively.

IoUs	0.25				0.50				0.75			
Object Sizes	All	S	M	L	All	S	M	L	All	S	M	L
FAOA (Yang et al., 2019)	44.5	30.6	48.6	54.1	30.4	15.8	36.2	39.3	8.5	1.4	9.6	14.4
ReSC (Yang et al., 2020)	49.2	32.3	54.7	60.1	34.9	14.1	42.5	47.7	10.5	0.2	10.6	20.1
YourefIt PAF (Chen et al., 2021)	52.6	35.9	60.5	61.4	37.6	14.6	49.1	49.1	12.7	1.0	16.5	20.5
YouRefIt Full (Chen et al., 2021)	54.7	38.5	64.1	61.6	40.5	16.3	54.4	51.1	14.0	1.2	17.2	23.3
Ours (Inpainting)	59.0	41.3	59.3	75.8	51.3	32.1	54.6	66.7	32.4	9.4	33.3	53.4
Ours (No Explicit Gestural Key Points)	64.9	49.6	67.9	76.7	57.4	40.8	62.1	69.0	37.2	14.4	39.7	56.7
Ours (EWL)	69.5	56.6	71.7	80.0	60.7	44.4	66.2	71.2	35.5	11.8	38.9	55.0
Ours (VTL)	71.1	55.9	75.5	81.7	63.5	47.0	70.2	73.1	39.0	13.4	45.2	57.8
Human (Chen et al., 2021)	94.2	93.7	92.3	96.3	85.8	81.0	86.7	89.4	53.3	33.9	55.9	68.1

Precision computed using GIoU thresholds We provide the performance of our models when evaluated using GIoU instead of IoU thresholds; see Tab. 7.

Table 7: **Model performances when evaluated using GIoU thresholds.**

	IoU=.25	IoU=.50	IoU=.75
Ours (Inpainting)	57.9	50.9	31.4
Ours (No Exp. Ges. K. Poi.)	63.7	56.5	36.2
Ours (EWL)	67.9	59.7	34.8
Ours (VTL)	70.0	62.5	38.2

Corrupted language inputs We investigate the performance of our model when using corrupted language inputs by changing each input sentence to the word "object." Our results (Tab. 8) show that corrupting language inputs severely impair the performance of our models.

Table 8: **Model performances when using corrupted language inputs.**

	Corrupted language	Re-training	IoU=.25	IoU=.50	IoU=.75
Ours (No Exp. Ges. K. Poi.)	True	False	19.5	16.0	7.8
Ours (VTL)	True	False	31.4	25.3	13.1
Ours (No Exp. Ges. K. Poi.)	True	True	49.2	42.1	23.4
Ours (VTL)	True	True	49.4	41.7	23.9
Ours (No Exp. Ges. K. Poi.)	False	False	64.9	57.4	37.2
Ours (VTL)	False	False	71.1	63.5	39.0

5 Conclusion and Limitations

We presented an effective approach, named Touch-Line Transformer, to utilize the simple but effective VTLs to improve an artificial agent's ability to locate referents referred to by humans in the wild. Our approach is inspired by recent findings in psychology studies on the touch-line hypothesis, which further revealed that people frequently misinterpret other people's referring expressions. Our proposed architecture, combined with the proposed VTL, significantly reduced the gap between model performance and human performance.

Some limitations exist in our work. First, resizing operations before and after inpainting might influence model performance. Next, we primarily study the eyes' location and the upper limbs' orientation regarding nonverbal signals, leaving the study of other types of nonverbal signals to future works, such as gazing direction, the direction of the finger, and the orientation of the lower limbs.

REFERENCES

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *International Conference on Computer Vision (ICCV)*, 2015. 2

Patrick Bédard, Arul Thangavel, and Jerome N Sanes. Gaze influences finger movement-related and visual-related activation across the human brain. *Experimental Brain Research*, 188(1):63–75, 2008. 2

Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *International Conference on Computer Vision (ICCV)*, 2019. 4

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision (ECCV)*, 2020. 2

Yixin Chen, Qing Li, Deqian Kong, Yik Lun Kei, Song-Chun Zhu, Tao Gao, Yixin Zhu, and Siyuan Huang. Yourefit: Embodied reference understanding with language and gesture. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1385–1395, 2021. 1, 2, 5, 9

Eunji Chong, Nataniel Ruiz, Yongxin Wang, Yun Zhang, Agata Rozga, and James M Rehg. Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of gaze and scene saliency. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 383–398, 2018. 3

Eunji Chong, Yongxin Wang, Nataniel Ruiz, and James M Rehg. Detecting attended visual targets in video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5396–5406, 2020. 3

Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. Guesswhat?! visual object discovery through multi-modal dialogue. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Visual grounding with transformers. *arXiv preprint arXiv:2105.04281*, 2021. 3

Yi Fang, Jiapeng Tang, Wang Shen, Wei Shen, Xiao Gu, Li Song, and Guangtao Zhai. Dual attention guided gaze target detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11390–11399, 2021. 3

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

Oliver Herbort and Wilfried Kunde. Spatial (mis-)interpretation of pointing gestures to distal referents. *Journal of Experimental Psychology: Human Perception and Performance*, 42(1):78, 2016. 2

Oliver Herbort and Wilfried Kunde. How to point and to interpret pointing gestures? instructions can reduce pointer–observer misunderstandings. *Psychological Research*, 82(2):395–406, 2018. 2

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *International Conference on Computer Vision (ICCV)*, 2021. 2, 4, 5, 15

Sotaro Kita. *Pointing: Where language, culture, and cognition meet*. Psychology Press, 2003. 2

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 2

Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 13

Yunhao Li, Wei Shen, Zhongpai Gao, Yucheng Zhu, Guangtao Zhai, and Guodong Guo. Looking here or there? gaze following in 360-degree images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3742–3751, 2021. 3

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 1

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 4

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. In *Robotics: Science and Systems (RSS)*, 2021. 3

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of instance segmentation and object detection algorithms in pytorch, 2018. 13

Colin McGinn. The mechanism of reference. *Synthese*, pp. 157–186, 1981. 2

Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters (RA-L)*, 2022. 3

Cathal O'Madagain, Gregor Kachel, and Brent Strickland. The origin of pointing: Evidence for the touch hypothesis. *Science Advances*, 5(7):eaav2558, 2019. 2

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning (ICML)*, 2018. 4

Adria Recasens, Aditya Khosla, Carl Vondrick, and Antonio Torralba. Where are they looking? *Advances in neural information processing systems*, 28, 2015. 3

Adria Recasens, Carl Vondrick, Aditya Khosla, and Antonio Torralba. Following gaze in video. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1435–1443, 2017. 3

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations (ICLR)*, 2018. 5

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision (ECCV)*, 2016. 3

Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6): 233–242, 1999. 3

Natalie Sebanz, Harold Bekkering, and Günther Knoblich. Joint action: bodies and minds moving together. *Trends in Cognitive Sciences*, 10(2):70–76, 2006. 7

Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. 2

Danyang Tu, Xiongkuo Min, Huiyu Duan, Guodong Guo, Guangtao Zhai, and Wei Shen. End-to-end human-gaze-target detection with transformers. *arXiv preprint arXiv:2203.10433*, 2022. 3

Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *International Conference on Computer Vision (ICCV)*, 2019. 5, 9

Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive sub-query construction. In *European Conference on Computer Vision (ECCV)*, 2020. 5, 9

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision (ECCV)*, 2016. 3

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

Hao Zhao, Ming Lu, Anbang Yao, Yurong Chen, and Li Zhang. Learning to draw sight lines. *International Journal of Computer Vision*, 128(5):1076–1100, 2020. 3

A APPENDIX

A.1 INPAINTING

We hypothesize that our model may learn to use postural signals without being explicitly asked to. To test our hypothesis, we remove postural signals (by removing humans) from the images mainly using the Mask R-CNN (Massa & Girshick, 2018) and MAT (Li et al., 2022). We investigate whether our model performs worse when trained using images without gestural signals.

We hypothesize that transformer models may learn to use postural signals without being explicitly asked to learn this type of signals. To test our hypothesis, we conduct two groups of experiments: the inpainting group and the control group. In the inpainting group, we remove postural signals in the input image. In the control group, we do not modify the input image.

In the inpainting group, we modify input images. Specifically, we remove postural signals from input images by removing humans and filling missing portions of the image (which were originally occupied by humans) using MAT (Li et al., 2022). Specifically, we first use Mask R-CNN X-101 (Massa & Girshick, 2018) to produce human masks. After that, we expand the human masks produced by the mask rcnn to both the left and the right sides to completely cover the edge of humans. We make sure that the expanded mask never encroaches on regions occupied by the ground truth bounding box for the referent. After that, we feed the expanded masks into MAT. With input masks, MAT removes the regions covered by the masks and fills these regions. Examples of masks, expanded masks, and inpaintings are in Fig. 9.

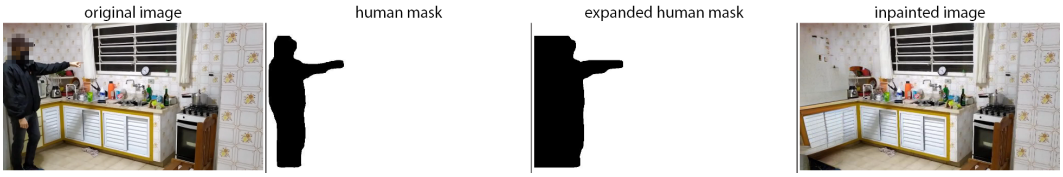


Figure 9: Illustration of the inpainting process. We remove gestural signals from input images before feeding images into our model to study the effects of implicitly learned postural signals.

We remove gestural signals (through inpainting) from images when studying our model's ability to implicitly learn these signals. Before generating inpaintings, we expand the human mask to both sides by 50 pixels. We reshape masks and images to 512×512 before feeding them into the MAT model because the checkpoint produced by MAT only works for inputs of size 512×512. Outputs of the MAT model are reshaped to their original sizes before feeding into our model. We observe that, for a very small number of images, human masks cannot be generated by F-RCNNs. In these very rare cases, we use the original image instead.

A.2 ADDITIONAL LOSSES

A.2.1 LOSSES FOR PREDICTED REFERENT BOUNDING BOXES

We use a weighted sum of L1 and GIoU losses for predicted bounding boxes. Each bounding box $B = (x, y, w, h)$ is represented using the x and y coordinated of the box center (x and y), width (w), and height (h). We denote predicted box as B_p and its ground truth as B_t.

L1 Loss For each pair of predicted box B_{p_i} and ground truth box B_{t_i}, the L1 loss is:

$$L_{L1_i} = |B_{p_i} - B_{t_i}| = |x_{p_i} - x_{t_i}| + |y_{p_i} - y_{t_i}| + |w_{p_i} - w_{t_i}| + |h_{p_i} - h_{t_i}| \tag{4}$$

Each time, there are n pairs of predicted and ground truth boxes. The total L1 loss for all pairs is:

$$L_{L1} = \frac{1}{n} \sum_{i=1}^{n} L_{L1_i} \tag{5}$$

GIoU Loss Before computing the GIoU Loss [TODO: citations], each box $B = (x, y, w, h)$ is transformed to $\bar{B} = (x_{min}, y_{min}, x_{max}, y_{max})$, where $x_{min} = x - \frac{w}{2}, x_{max} = x + \frac{w}{2}, y_{min} = y - \frac{h}{2}, y_{max} = y + \frac{h}{2}$.

The area of predicted box $\bar{B}_p$ and ground truth box $\bar{B}_t$ are computed as:

$$Area_{p_i} = (x_{p_{max_i}} - x_{p_{min_i}}) \times (y_{p_{max_i}} - x_{y_{min_i}}) \tag{6}$$

$$Area_{t_i} = (x_{t_{max_i}} - x_{t_{min_i}}) \times (y_{t_{max_i}} - x_{t_{min_i}}) \tag{7}$$

The IoU and Union of $\bar{B}_p$ and $\bar{B}_t$ are computed as:

$$x_{left_i} = \max(x_{p_{min_i}}, x_{t_{min_i}}) \tag{8}$$

$$y_{top_i} = \max(y_{p_{min_i}}, y_{t_{min_i}}) \tag{9}$$

$$x_{right_i} = \min(x_{p_{max_i}}, x_{t_{max_i}}) \tag{10}$$

$$y_{bottom_i} = \min(y_{p_{max_i}}, y_{t_{max_i}}) \tag{11}$$

$$Intersection_i = (x_{right_i} - x_{left_i}) \times (y_{bottom_i} - y_{top_i}) \tag{12}$$

$$Union_i = Area_{p_i} + Area_{t_i} - Intersection_i \tag{13}$$

$$IoU_i = \frac{Intersection_i}{Union_i} \tag{14}$$

For each pair of $\bar{B}_{p_i}$ and $\bar{B}_{t_i}$, the GIoU is:

$$x'_{left_i} = \min(x_{p_{min_i}}, x_{t_{min_i}}) \tag{15}$$

$$y'_{top_i} = \min(y_{p_{min_i}}, y_{t_{min_i}}) \tag{16}$$

$$x'_{right_i} = \max(x_{p_{max_i}}, x_{t_{max_i}}) \tag{17}$$

$$y'_{bottom_i} = \max(y_{p_{max_i}}, y_{t_{max_i}}) \tag{18}$$

$$Area'_i = (x'_{right_i} - x'_{left_i}) \times (y'_{bottom_i} - y'_{top_i}) \tag{19}$$

$$GIoU_i = IoU_i - \frac{Area'_i - Union_i}{Area'_i} \tag{20}$$

The GIoU loss for one pair of predicted box and target box is:

$$L_{GIoU_i} = 1 - GIoU_i \tag{21}$$

The GIoU loss for all pairs of predicted and target boxes is:

$$L_{GIoU} = \frac{1}{n} \sum_{i=1}^{n} L_{GIoU_i} \tag{22}$$

A.2.2 LOSSES FOR GESTURAL KEY POINTS

For the predicted eyes and fingertips or elbows and wrists, we use L1 loss. Specifically, each time, our model predicts m pairs of gestural key points. Each pair is denoted as $pair_{p_i} = (x_{eye_{p_i}}, y_{eye_{p_i}}, x_{fgt_{p_i}}, y_{fgt_{p_i}})$ (for VTL) or $pair_{p_i} = (x_{elb_{p_i}}, y_{elb_{p_i}}, x_{wst_{p_i}}, y_{wst_{p_i}})$ (for EWL). The ground truth gestural key points are represented as $pair_{t_i} = (x_{eye_{t_i}}, y_{eye_{t_i}}, x_{fgt_{t_i}}, y_{fgt_{t_i}})$ (for VTL) or $pair_{t_i} = (x_{elb_{t_i}}, y_{elb_{t_i}}, x_{wst_{t_i}}, y_{wst_{t_i}})$ (for EWL).

The loss for each predicted gestural pairs is defined as:

$$L_{gesture_L1_i} = |x_{eye_{p_i}} - x_{eye_{t_i}}| + |y_{eye_{p_i}} - y_{eye_{t_i}}| + |x_{fgt_{p_i}} - x_{fgt_{t_i}}| + |y_{fgt_{p_i}} - y_{fgt_{t_i}}| \tag{23}$$

for VTL, or

$$L_{gesture_L1_i} = |x_{elb_{p_i}} - x_{elb_{t_i}}| + |y_{elb_{p_i}} - y_{elb_{t_i}}| + |x_{wst_{p_i}} - x_{wst_{t_i}}| + |y_{wst_{p_i}} - y_{wst_{t_i}}| \tag{24}$$

for EWL.

The L1 gestural key point loss is:

$$L_{gesture_L1} = \min L_{gesture_i}, i \in \{1, ..., m\} \tag{25}$$

Additionally, we apply a cross entropy loss $L_{gesture_CE}$ for predicted gestural key points with two classes: "are gestural key points" and "are not gestural key points."

The total loss for gestural key points is:

$$L_{gesture} = \alpha_{gesture_L1} \cdot L_{gesture_L1} + \alpha_{gesture_CE} \cdot L_{gesture_CE} \tag{26}$$

where $\alpha_{gesture_L1} = 6$ and $L_{gesture_CE} = 1.5$.

A.2.3 SOFT TOKEN LOSS

For each object, we predict token spans produced by the BPE scheme [TODO: citations], instead of object categories, and set the maximum number of tokens to 256 following Kamath et al. (2021). Following Kamath et al. (2021), we use a soft token loss L_{token} for the predicted token spans. The soft token loss is a cross entropy loss. Specifically, an object may correspond to k token locations ($1 \leq k \leq 256$), and the ground truth probability for each of the k token location is $\frac{1}{k}$.

A.2.4 MATCHING STRATEGY

We match prediction and ground truth using the Hungarian algorithm (TODO: citations) by minimizing the cost C:

$$C = \alpha_{L1} \cdot L_{L1} + \alpha_{GIoU} \cdot L_{GIoU} + \alpha_{token} \cdot L_{token} \tag{27}$$

where $\alpha_{L1} = 5$, $\alpha_{GIoU} = 2$, and $\alpha_{token} = 1$.

A.2.5 CONTRASTIVE ALIGNMENT LOSS

The contrastive alignment loss is used to encourage alignment between object feature from transformer decoder and text feature from transformer encoder.

The number of predicted objects is n, and the number of text tokens is l. Let F_{o_i} and F_{t_j} denotes the feature of the i th object and the feature of the j th token, respectively. At the same time, let T_i^+ denotes the set of tokens to be aligned to by the i th object, and let O_i^+ denotes the set of objects to be aligned to the i th token. Meanwhile, $\tau = 0.07$ is the temperature.

For all objects, the contrastive alignment loss is:

$$L_{contrastive_o} = \sum_{i=1}^{n} \frac{1}{|T_i^+|} \sum_{j \in T_i^+} -\log \frac{\exp(F_{o_i}^T F_{t_j} / \tau)}{\sum_{k=1}^{l} \exp(F_{o_i}^T F_{t_k} / \tau)} \tag{28}$$

For all text tokens, the contrastive alignment loss is:

$$L_{contrastive_t} = \sum_{i=1}^{l} \frac{1}{|O_i^+|} \sum_{j \in O_i^+} -\log \frac{\exp(F_{t_i}^T F_{o_j}/\tau)}{\sum_{k=1}^{n} \exp(F_{t_i}^T F_{o_k}/\tau)} \tag{29}$$

The final contrastive alignment loss is the average of $L_{contrastive_o}$ and $L_{contrastive_t}$:

$$L_{contrastive} = \frac{L_{contrastive_o} + L_{contrastive_t}}{2} \tag{30}$$

A.3 Computing Cosine Similarities using Different Lines

The cosine similarity in Eq. (1) can be computed using different lines. Specifically, the eye, the fingertip, and the object can form a triangle, and there are three ways of choosing two lines from a triangle. In Eq. (1), we use the two lines connected by the eye.

We investigate the effects of using different lines for cosine similarity computation. Specifically, we conduct an additional experiment using two lines connected by the fingertip. In other words, we use the following two vectors: one from the fingertip to the eye, and the other from the object center to the fingertip.

Our results (Tab. 9) show that using the lines connected by the eye and using the two lines connected by the fingertip can be regarded as fungible.

In specific, using the two lines connected by the fingertip, the model's performance is 69.0, 60.8, and 37.3 under the IoU threshold of 0.25, 0.50, and 0.75, respectively. Compared to the EWL model, it obtains a +1.5 performance boost under the IoU threshold of 0.75. Compared to the No Explicit Gestural Key Points Model, it obtains a +4.1 and +3.4 performance boost under the IoU threshold of 0.25 and 0.50, respectively.

Table 9: **Using different lines for cosine similarity computation in the VTL model.**

	IoU=.25	IoU=.50	IoU=.75
Ours (No Explicit Gestural Key Points)	64.9	57.4	37.2
Ours (EWL)	**69.5**	60.7	35.5
Ours (VTL, cos_sim Vertex = Fingertip)	69.0	**60.8**	**37.3**