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.

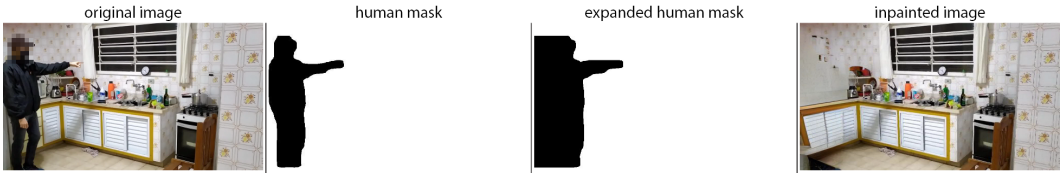

Figure 9: Illustration of the inpainting process. We remove gestural signals from input images before feeding images into our model to study the effects of implicitly learned postural signals.

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**