Human-in-the-Loop Local Corrections of 3D Scene Layouts via Infilling: Supplemental Material

Christopher Xie Armen Avetisyan Henry Howard-Jenkins Yawar Siddiqui
Julian Straub Richard Newcombe Vasileios Balntas Jakob Engel

Meta Reality Labs

https://projectaria.com/scenescript

	Eval Dataset	Global Pred. AvgF1 (↑) wall	door	window	Local Correction AvgF1 (↑) wall	door	window
Baseline [2]	iASE	**79.4**	**87.2**	**77.7**	87.2	88.0	83.1
+ joint training	iASE	77.5	87.0	77.5	**92.5**	**92.9**	**88.0**
Baseline	AEO	**34.9**	**27.8**	**22.9**	29.0	9.4	9.5
+ joint training	AEO	33.0	25.1	18.6	**56.0**	**23.4**	**25.1**

Table 5. Joint training ablation. These models are trained on our internal proprietary version of ASE (iASE). We show AvgF1 scores for the global prediction (GP) and local correction (LC) tasks. Note that the gaps in OOD performance on AEO are substantially amplified compared to Table 1 of the main paper.

Ordering	Local Correction AvgF1 (↑) wall	door	window
lex	**93.4**	**93.5**	**89.7**
angle	92.9	92.9	88.3
random	92.5	92.9	88.0

Table 6. Language ordering ablation for models trained on iASE.

Ordering	SPE	Local Correction AvgF1 (↑) wall	door	window
lex	✗	92.7	92.9	88.3
	✓	93.4 (0.7)	93.5 (0.6)	89.7 (1.4)
angle	✗	92.6	91.8	87.3
	✓	92.9 (0.3)	92.9 (1.1)	88.3 (1.0)
random	✗	91.5	92.7	88.1
	✓	92.5 (1.0)	92.9 (0.2)	88.0 (0.1)

Table 7. Subsequence positional embedding (SPE) ablation for models trained on iASE.

Ordering	Ego	Local Correction AvgF1 (↑) wall	door	window
lex	✗	92.9	92.7	86.6
	✓	92.7 (0.2)	92.9 (0.2)	88.3 (1.7)
random	✗	91.2	92.0	86.3
	✓	91.5 (0.3)	92.7 (0.7)	88.1 (1.8)

Table 8. Egocentric anchoring ablation for models trained on iASE.

Appendix

A. Additional Quantitative Results

A.1. Results on Internal Version of ASE

In this section, we repeat the experiments in Section 4.3 with our internal version of ASE [2] (iASE) that contains more complex scene layouts.

Joint Training. As observed in [1, 3, 5] and Table 1, we again see in Table 5 a slight drop in performance on global prediction AvgF1 but a large gain on local correction AvgF1 when evaluating on in-distribution synthetic validation data (iASE). These models are trained with random language ordering, and subsequence positional embeddings and egocentric anchoring for local corrections.

However, on OOD validation data (AEO [7]), the gains on local correction AvgF1 are significantly amplified compared to the results in Table 1 (trained on ASE, not iASE). The wall AvgF1 is nearly doubled in this setting, and the door/window AvgF1 are roughly 2.5x better than the baseline. Thus, training our models on the more complex iASE, which has less sim-to-real gap, provides strong evidence of the benefit of jointly training for both global prediction and local correction.

Language Ordering. We compare lexicographic, angle, and random language ordering in Table 6. These models are trained with joint training, subsequence positional em-

Ordering	Local Correction AvgF1 (↑)			
	$n_w = 1$	$n_w = 2$	$n_w = 3$	$n_w \geq 4$
lex	**99.9**	**97.6**	**96.0**	**95.9**
angle	99.8	96.9	95.1	95.4
random	99.8	96.0	95.2	94.8

Table 9. Language ordering ablation breakdown by number of walls (n_w) on ASE [2]. All AvgF1 numbers are shown for walls only.

Ordering	Local Correction AvgF1 (↑)			
	$n_w = 1$	$n_w = 2$	$n_w = 3$	$n_w \geq 4$
lex	99.4	**93.3**	**88.8**	**84.3**
angle	**99.5**	92.1	87.9	81.8
random	**99.5**	91.7	86.6	80.1

Table 10. Language ordering ablation breakdown by number of walls (n_w) on our internal version of ASE. All AvgF1 numbers are shown for walls only.

beddings, and egocentric anchoring. As in Table 2, we see that the lexicographic model slightly outperforms the angle model, which slightly outperforms the random model. Gaps performance are slightly larger on this more complex dataset.

Subsequence Positional Embedding. We compare using a per-subsequence positional embedding against a conventional positional embedding that embeds absolute position. These models are trained with joint training and egocentric anchoring. As seen in Table 7, using the per-subsequence positional embedding shows a consistent gain in performance, corroborating the results in Table 3.

Egocentric Anchoring. Table 8 studies the effectiveness of anchoring the scene at the user's pose (for local corrections only) as opposed to translating the scene to the positive quadrant. These models are trained with joint training and conventional positional embeddings. Results show the efficacy of this design choice across almost all metrics in multiple configurations of the models, similar to Table 4.

A.2. Multi-Wall Evaluation

We note that in our evaluation of local corrections for walls, many of the examples only contain a single wall to infill. This problem is easy, as the corners for the missing wall are present in the prefix/suffix sequences, so the network only needs to learn to copy those corners from the correct places. When the number of walls $n_w > 1$, the network must predict the location of at least one corner.

In Tables 9 and 10, we show an analysis of this breakdown by n_w for our models trained with different language orderings on ASE [2] and our internal version of ASE, respectively. We separate the evaluation from 1 to ≥ 4 walls. In both tables, performance on $n_w = 1$ is nearly perfect as the problem is easy. However, the performance of all models degrades as n_w increases, because the problem becomes more difficult. Similarly, the performance gaps for the methods generally increase as the number of walls increases.

B. User Study

Due to the novelty of the local correction task and our proposed human-in-the-loop system, there are no existing baseline systems to compare to. Thus, to obtain insight into our system, we conducted a user study comparing different models integrated into our low-friction "one-click fix" system. In particular, we compare the vanilla SceneScript baseline vs. our multi-task SceneScript (lexicographically-ordered), both trained on iASE. Note that these models have not been trained with the extra capability of infilling prefix/suffix corners described in Section 4.4.2.

We used pre-recorded Aria trajectories in real-world home and office environments. We used our system in an offline fashion by manually selecting a frame and the entities to fix. At each local correction step, we ran both the baseline model and our multi-task model, and asked users to select the local correction that they preferred. Averaging over 10 users and 18 local correction examples, 57.2% preferred our model, 18.9% preferred the baseline, and 23.9% preferred neither. This highlights that the joint training is crucial in learning a model that can be integrated with our human-in-the-loop system.

We note that a common failure mode of the vanilla SceneScript baseline is to re-predict scene layout elements (walls/doors/windows) that already exist in the current estimate, leading to redundant scene entities. Although this behavior is uncommon when evaluating on in-distribution validation data, the out-of-distribution nature of real-world data likely causes ambiguity around when the baseline should predict the `<STOP>` token. The jointly trained model, however, is much more reliable in OOD scenarios and consistently predicts `<STOP>` at the correct time and avoids this failure mode.

C. Action Selection Heuristic

Here, we detail the action selection heuristic briefly mentioned in Section 4.4.1. Algorithm 1 provides the details. In lines 1-2, we compute sets of instance masks I_E by projecting the current scene layout estimate onto the image to create instance masks, and I_M by running Mask2Former [4] (M2F) for walls/doors/windows. We then compute Intersection over Union (IoU) scores in image space between all pairs of instance masks and threshold them by δ. Next, we loop through all masks in I_E to see if they match with masks in I_M (please see lines 5-20 for the details). Similarly, we

Algorithm 1 Action Selection Heuristic

Input: Visible layout entity set E. Video frame f. IoU threshold δ.
Output: Action a. Selected entity set S.

1: Convert E to instance masks to obtain set of instance masks I_E.
2: Run Mask2Former [4] on f to obtain set of instance masks I_M.
3: Compute all-pairs IoU for I_M, I_E, threshold by δ to obtain matches.
4: $P = \{\}$ ▷ potential actions
5: **for** mask e in I_E **do**
6: $c = \text{class}(e)$ ▷ entity class
7: **if** $c =$ wall **then**
8: **if** e is matched with $C \subseteq I_M[c]$ where $|C| \geq 2$ **then**
9: $P \leftarrow P \cup \{(\texttt{Infill}, \{e\})\}$ ▷ potentially split wall via `Infill`
10: **end if**
11: **else** ▷ door or window
12: **if** $I_M[c] = \emptyset$ **then**
13: $P \leftarrow P \cup \{(\texttt{Delete}, \{e\})\}$ ▷ M2F detects no entities of class c: `Delete`
14: **else**
15: **if** no matches for e in $I_M[c]$ **then**
16: $P \leftarrow P \cup \{(\texttt{Infill}, \{e\})\}$ ▷ M2F detects an entity of class c but has low overlap with e: `Infill`
17: **end if**
18: **end if**
19: **end if**
20: **end for**
21: **for** mask m in I_M **do**
22: $c = \text{class}(m)$ ▷ entity class
23: **if** $c =$ wall **then**
24: **if** m is matched with $T \subseteq I_E[c]$ where $|T| \geq 2$ **then**
25: $P \leftarrow P \cup \{(\texttt{Infill}, T)\}$ ▷ potentially merge walls in T via `Infill`
26: **end if**
27: **else** ▷ door or window
28: **if** $I_E[c] = \emptyset$ **then**
29: $P \leftarrow P \cup \{(\texttt{Add}, c)\}$ ▷ `Add` the class
30: **end if**
31: **end if**
32: **end for**
33: $(a, S) \sim \text{Uniform}(P)$ ▷ Select a random action
34: **return** (a, S)

loop through masks in I_M to see if they match with masks in I_E (please see lines 21-32 for the details). Finally, we randomly sample an action from the list of possible actions in line 33.

We define the following notation for instance mask sets I to select the masks of a certain class: $I[c] = \{x \in I \mid \text{class}(x) = c\}$. Note that while Algorithm 1 shows the heuristic for only 1 frame, Project Aria [6] is a multi-camera system that provides multiple images per capture, thus we run the algorithm over the images (we use the SLAM camera images).

D. Conceptual Comparison to Other Global Prediction Methods

Direct quantitative comparison to existing global prediction methods is not applicable in our setting due to the novelty of the local correction task (thus we created a baseline using vanilla SceneScript, i.e. trained only on global predictions). However, we note that our model performs a global prediction as an initial step (which can be quantiatively compared). By incorporating iterative local corrections, our method further refines this initial prediction, making the final refined layout quantitatively better than its initial global prediction.

In this respect, if our model's global prediction outper-

forms a competitor's global prediction results, we'd expect that our full system would further outperform the competitor quantitatively. We report RoomFormer's [8] global prediction results on ASE (from [2]) for completeness: 85.2 / 79.8 / 72.3 Global Prediction AvgF1, which can be compared to Table 1 in the main paper. Note that these numbers differ slightly from [2] due to a difference in F1 thresholds as mentioned in Section 4.2. As our model's global prediction is similar to vanilla SceneScript and still outperforms RoomFormer, we expect that refining our global prediction with our local corrections will yield an even larger quantitative gap.

E. Additional Results for Human-in-the-Loop System

E.1. Global Prediction with Accumulated Points

As mentioned in Section 4.4.2, the human-in-the-loop system accumulates more semi-dense points over time. This allows for users to re-scan poorly scanned areas to obtain more signal for local corrections. In Figure 5, we demonstrate that this additional signal may not be helpful for the global prediction paradigm, likely due to the fact that the ground truth layout is OOD with respect to the training distribution. We show our refined layouts compared to running the model in global prediction mode on the final point cloud with all the accumulated semi-dense points. The global prediction lacks the correct amount and placement of doors and windows, and the wall structure is simplistic in comparison to our refined layout. We hypothesize that these real-world scenes are out of distribution with respect to the training distribution. This again highlights that our human-in-the-loop framework can produce scene layouts that the single-shot global prediction (i.e. vanilla SceneScript) cannot produce.

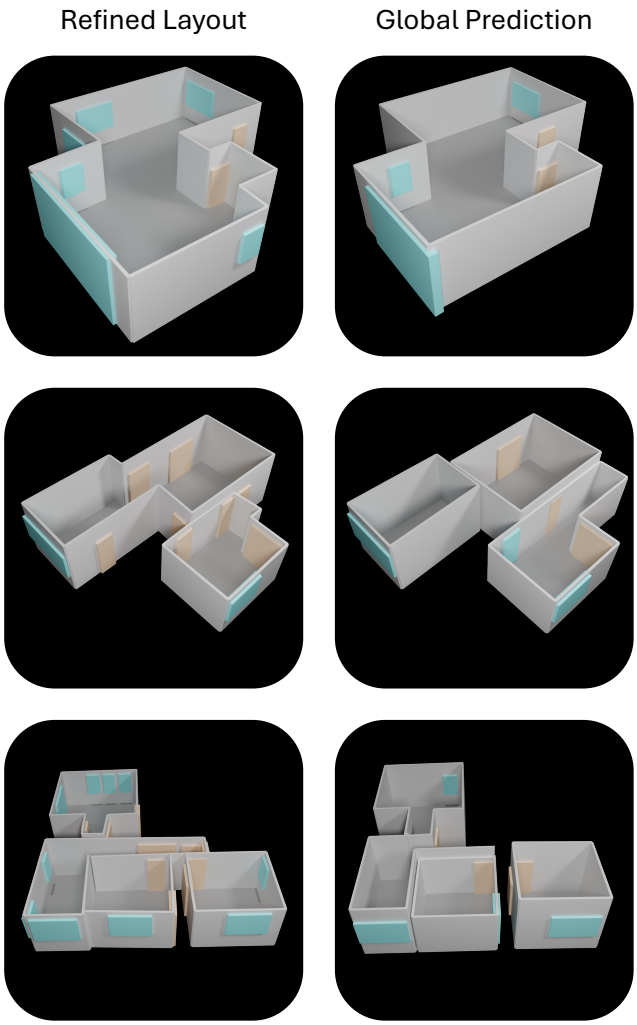

Figure 5. Refined Layout from our Human-in-the-Loop System vs. Global Prediction using all accumulated points. The top row is the scene from Figure 1, and the bottom two rows are the scenes from Figure 4 (please reference this figure to see the manually-annotated ground truth for qualitative purposes only). See the text for more details.

E.2. Demo Videos

At our project webpage, we provide videos of our human-in-the-loop system running live on a Meta Quest 3 with an Aria device rigidly attached. The three videos correspond to the three real-world scenes used for qualitative results in the main paper (one from Figure 1 and two from Figure 4). On the left of the video, we highlight the current layout estimate, action, selected entities, and infilled entities.

E.3. Failure Cases

In our supplemental materials, we provide a video of failure cases of our human-in-the-loop system. We detail them here:

- Our model struggles to accurately identify individual windows when multiple small windows are positioned closely together. Groups of windows are often misidentified as a single large window. We posit that this is due to the nature of the synthetic training data. This can also be seen in our qualitative results in Figure 4.
- Although users typically expect a newly added entity to appear directly in front, our model does not consistently follow this behavior in practice. This is likely due to the heuristics used to select data (e.g. visibility as discussed in Section 3.1.2). Future work involves curating the data to better align the model's behavior with human expectations.
- Lastly, we show an example where the model is not capable of producing the correct geometry for a pair of small windows. We speculate that this style of window is not present anywhere in our synthetic training data. Although

the local correction task demonstrates better generalization when used in a human-in-the-loop system compared to the global prediction distribution, it is inherently limited to the learned local distributions. As a result, it cannot predict outcomes for patterns or scenarios that fall outside these distributions.

References

[1] Loubna Ben Allal, Raymond Li, Denis Kocetkov, Chenghao Mou, Christopher Akiki, Carlos Munoz Ferrandis, Niklas Muennighoff, Mayank Mishra, Alex Gu, Manan Dey, et al. Santacoder: don't reach for the stars! *arXiv preprint arXiv:2301.03988*, 2023. 1

[2] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, et al. Scenescript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision (ECCV)*, 2024. 1, 2, 4

[3] Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. Efficient training of language models to fill in the middle. *arXiv preprint arXiv:2207.14255*, 2022. 1

[4] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3

[5] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023. 1

[6] Kiran Somasundaram, Jing Dong, Huixuan Tang, Julian Straub, Mingfei Yan, Michael Goesele, Jakob Julian Engel, Renzo De Nardi, and Richard Newcombe. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*, 2023. 3

[7] Julian Straub, Daniel DeTone, Tianwei Shen, Nan Yang, Chris Sweeney, and Richard Newcombe. Efm3d: A benchmark for measuring progress towards 3d egocentric foundation models. *arXiv preprint arXiv:2406.10224*, 2024. 1

[8] Yuanwen Yue, Theodora Kontogianni, Konrad Schindler, and Francis Engelmann. Connecting the dots: Floorplan reconstruction using two-level queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 4