Supplementary Material

A. Implementation Details

IVLN Experiments We used a batch size of 8, a learning rate of 1e-5 and trained with the AdamW optimizer for all the IVLN models. The maximum number of steps was 15 per episode. In training, we chose a dropout rate of 0.5 and a environment dropout rate of 0.4. For the baseline HAMT model, we trained for a total of 100k iterations and evaluated every 2k iterations. For the TourHAMT models, we trained for a total of 5k iterations and evaluated every 200 iterations. Note that TourHAMT models were trained with batches of tours and the HAMT model was trained with batches of episodes. We trained all models with teacher-forcing and updated gradients per episode. Thus, TourHAMT models did not train with fewer gradient updates than the HAMT model. We ran IVLN experiments on Quadro RTX 6000 GPUs. The fine-tuning of the baseline HAMT model took 12 GPU hours and the training of each TourHAMT model took 70 GPU hours. Altogether, a total of 876 GPU hours were used for the IVLN experiments.

IVLN-CE Experiments We used a batch size of 5 for all training phases. For both teacher-forcing and DAgger training phases, we used a learning rate of 2.5e-4 with the Adam optimizer. For DAgger training, we performed 10 iterations with 4 epochs of training per iteration. We collected 5000 rollouts per iteration. We took the oracle action with probability $\beta = 0.5^{n+1}$ where n is the index of the current iteration and took the agent action otherwise. We evaluated the model after each epoch of training. During inference the model takes the argmax of the predicted action distribution. IVLN-CE experiments were run on Tesla V100 GPUs. Each unstructured latent memory model required 24 GPU hours to train and each semantic map model required 120 GPU hours to train. Altogether, 2016 total GPU hours were used for training and evaluation to support these experiments.

B. Evaluation Server Details

As mentioned in Sec. 1, we will release an evaluation server and public leaderboard to standardize and benchmark progress on IVLN. Like existing evaluation servers for VLN[2] and VLN-CE,[3] we will establish IR2R and IR2R-CE evaluation servers that accept and score prediction files. Participants will run inference of their models locally on a Test split. The prediction file will be scored on the evaluation server and the results will be added to a public leaderboard.

In existing evaluation servers of the Room-to-Room dataset, the target paths of the Test split are kept private to preserve the integrity of the split. However, in IVLN, the *oracle navigation* phase involves conveying the agent to the target goal location at the end of each episode in a tour. Running model inference in such a setting requires access to the target path. Given this concern, we modify the IVLN paradigm for leaderboard evaluation as follows:

1. The *oracle navigation* phase is modified to only convey the agent to the start of the next episode, rather than first to the target goal of the current episode.

2. Episodes in a tour are ordered such the distance between the start locations of sequential episodes is minimized (tip-to-tip). Ordering episodes tip-to-tail as originally presented for IVLN would compromise the target goal locations with high certainty.

These two modifications ensure that the privacy of the Test split target paths of Room-to-Room are preserved. A second concern is the domain gap between the Test split and rest of the IVLN splits induced by the modifications above. We regenerate the Train and validation splits of IR2R-CE with these modifications. In Fig. 5, we show the observation coverage of both upcoming episodes and complete tours of the regenerated Train split for direct comparison to Fig. 2b. We find that these modifications result in a slightly lower AUC of Complete Tour coverage while Upcoming Episode coverage is mostly similar. Finally, we evaluate our best `Map-CMA` model (Tab. 4 row 14) against the regenerated IR2R-CE Val-Unseen split. Performance metrics are similar to those reported in Tab. 4: 48 `t-nDTW` (+1), 36 `SR` (+1), and 33 `SPL` (+1). Thus, we conclude these modifications for Test set evaluation on a public leaderboard ensure the privacy of the R2R test paths while preserving the core IVLN evaluation challenges.

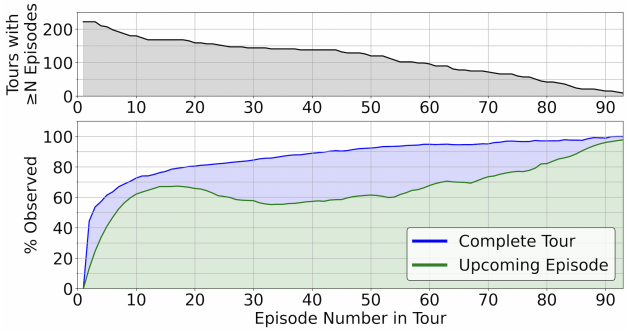

Figure 5. We repeat Fig. 2b for the IR2R-CE Train split regenerated using Test split modifications. This demonstrates that during a tour, the percentage already observed of both the Upcoming Episode and the Complete Tour are similar to the observation coverage experienced while following the tours proposed in the main paper.

C. Illustrative Figures

Figures 6, 7, 8, and 9 further illustrate IVLN and proposed, initial models.

[2]R2R server: eval.ai/web/challenges/challenge-page/97
[3]R2R-CE server: eval.ai/web/challenges/challenge-page/719

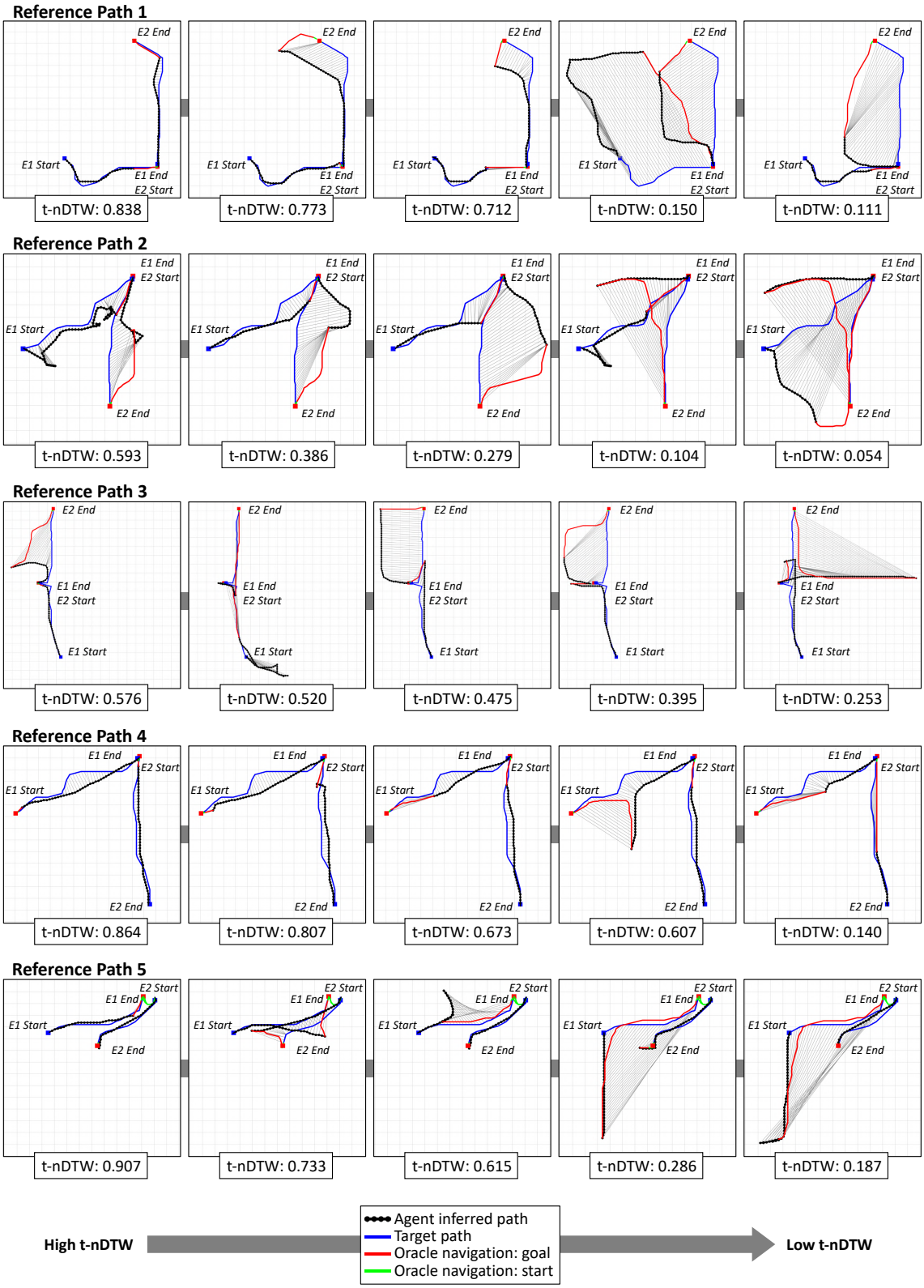

Figure 6. Ranked sample evaluations of `t-nDTW` scores applied to five reference paths. While `t-nDTW` evaluates the entire length of a tour, here we evaluate 2-episode sub-tours for illustration.

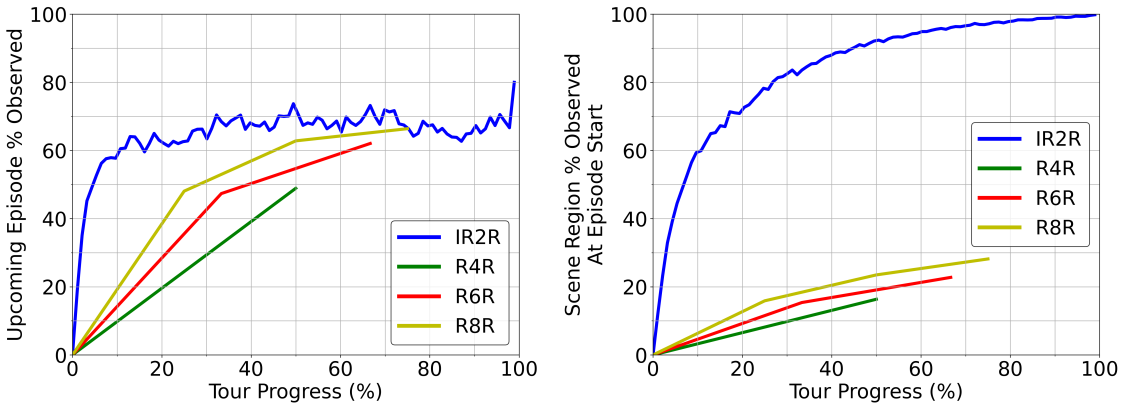

Figure 7. We compare IR2R-CE against the continuous environment versions of R4R, R6R, and R8R. Room-for-Room (R4R) is a benchmark in VLN consisting of long-distance instruction-following episodes synthesized from the Room-to-Room (R2R) dataset by joining two adjacent paths (and consequently instructions) tip-to-tail [21]. Joining 3 paths tip-to-tail is known as R6R and joining 4 paths tip-to-tail is known as R8R [51]. In this comparison, at the start of each episode we measure what percentage of that episode's target path has been observed earlier in the tour (left). Since the tours of IR2R-CE are significantly longer on average (100 episodes *vs*. 2, 3, or 4), the agent has a higher percentage of prior observability when tasked with a new instruction to follow. Also at the start of each episode, we measure what percentage of the observable scene region has been observed (right). Agents performing tours in IR2R-CE smoothly gain observation coverage of the scene region, eventually observing the entire space. On the other hand, agents performing tours in R8R average less than 30% observability coverage by the beginning of the final episode. The significantly higher and iteratively-procured episode coverage and scene region coverage in IVLN provides a realistic and rich signal for navigation planning that has yet to be featured in any other VLN benchmark.

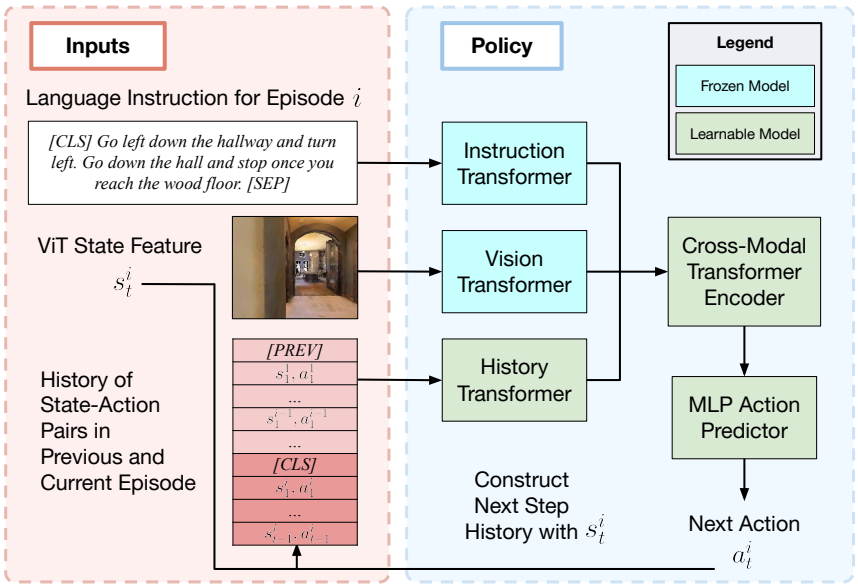

Figure 8. The TourHAMT model architecture. Unlike the original HAMT model [11], we unfreeze the History Transformer module, and append the previous episodes' state-action pairs as history (light red top part of the history box) on top of the current episode's history (dark red bottom part of the history box). The state-action history is empty at the beginning of the tour and accumulates as the number of experienced episodes grows.

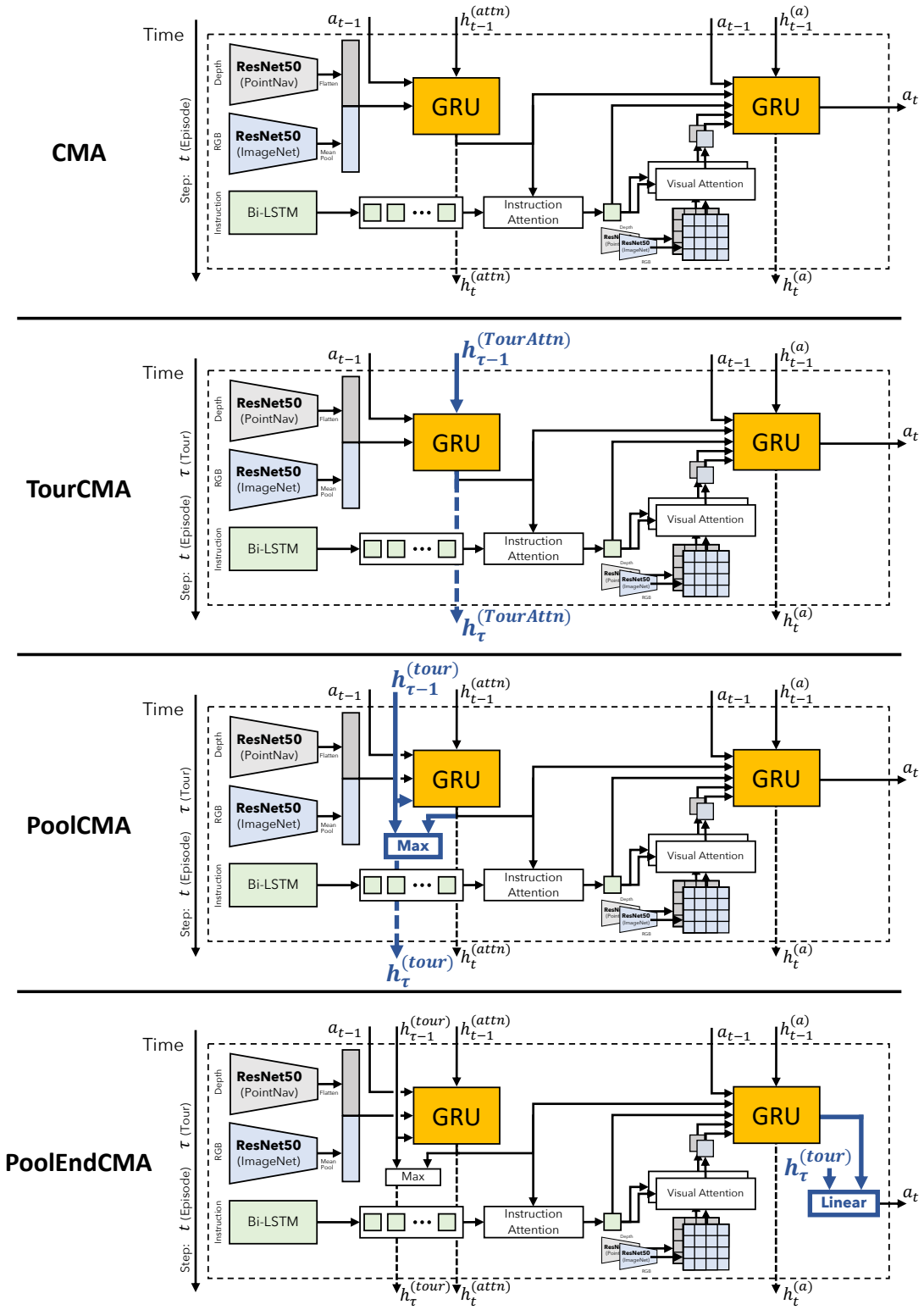

Figure 9. Architectures of unstructured memory models in IR2R-CE. Episode-based CMA as defined in [25] is shown on top. Tour-adapted versions are shown below with t indicating a step in the current episode and τ indicating a step in the current tour. Memory structures with episode steps, h_t, are reset for new episodes, whereas memory structures with tours steps, h_τ, are reset for new tours. Changes from the model immediately above are shown in bold in dark blue.