A. Supplementary Material

Our appendix is structured to provide both corresponding qualitative examples for the quantitative results in the paper and additional implementation details for replication.

A.1. Qualitative comparison

Figures A1 through A3 show three examples comparing our approach to the previous state-of-the-art. In addition, the following URL includes a 90 second video (https://youtu.be/AD9TNohXoPA) showing a first-person view of several agents navigating the environment with corresponding birds-eye-view maps.

A.2. Candidate Reranker

Given a collection of candidate trajectories, our reranker module assigns a score to each of the trajectories. The highest scoring trajectory is selected for the FAST agent's next step. In our implementation, we use a 2-layer MLP as the reranker. We train the neural network using pairwise cross-entropy loss [4].

As input to the reranker, we concatenate the following features to obtain a 6-dimensional vector:

- Sum of score logits for actions on the trajectory.
- Mean of score logits for actions on the trajectory.
- Sum of log probabilities for actions on the trajectory.
- Mean of log probability for actions on the trajectory.
- Progress monitor score for the completed trajector.
- Speaker score for the completed trajectory.

We feed the 6-dimensional vector through an *MLP*: `BN` → `FC` → `BN` → `Tanh` → `FC`, where `BN` is a layer of `Batch Normalization`, `FC` is a `Fully Connected` layer, and `Tanh` is the nonlinearity used. The first `FC` layer transforms the 6-dimensional input vector to a 6-dimensional hidden vector. The second `FC` layer project the 6-dimensional vector to a single floating-point value, which is used as the score for the given partial trajectory.

To train the *MLP*, we cache the candidate queue after running FAST for 40 steps. Each candidate trajectory in the queue has a corresponding score s_i. To calculate the loss, we minimize the pairwise cross-entropy loss:

$$-(s_1 - s_2) + \log(1 + \exp(s_1 - s_2))$$

where s_1 is the score for a qualified candidate and s_2 is the score for an unqualified candidate. We define *qualified candidate trajectories* as those that end within 3 meters of ground truth destination. In our cached training set, we have $4,378,729$ pairs of training data. We train using a batch size of 3600, SGD optimizer with a learning rate of $5e^{-5}$, and momentum 0.6; We train for 30 epochs.

Instructions: Walk across living room, at hallway on the right turn right and go down. Turn right at first door, enter pantry and stop in the middle of counter.

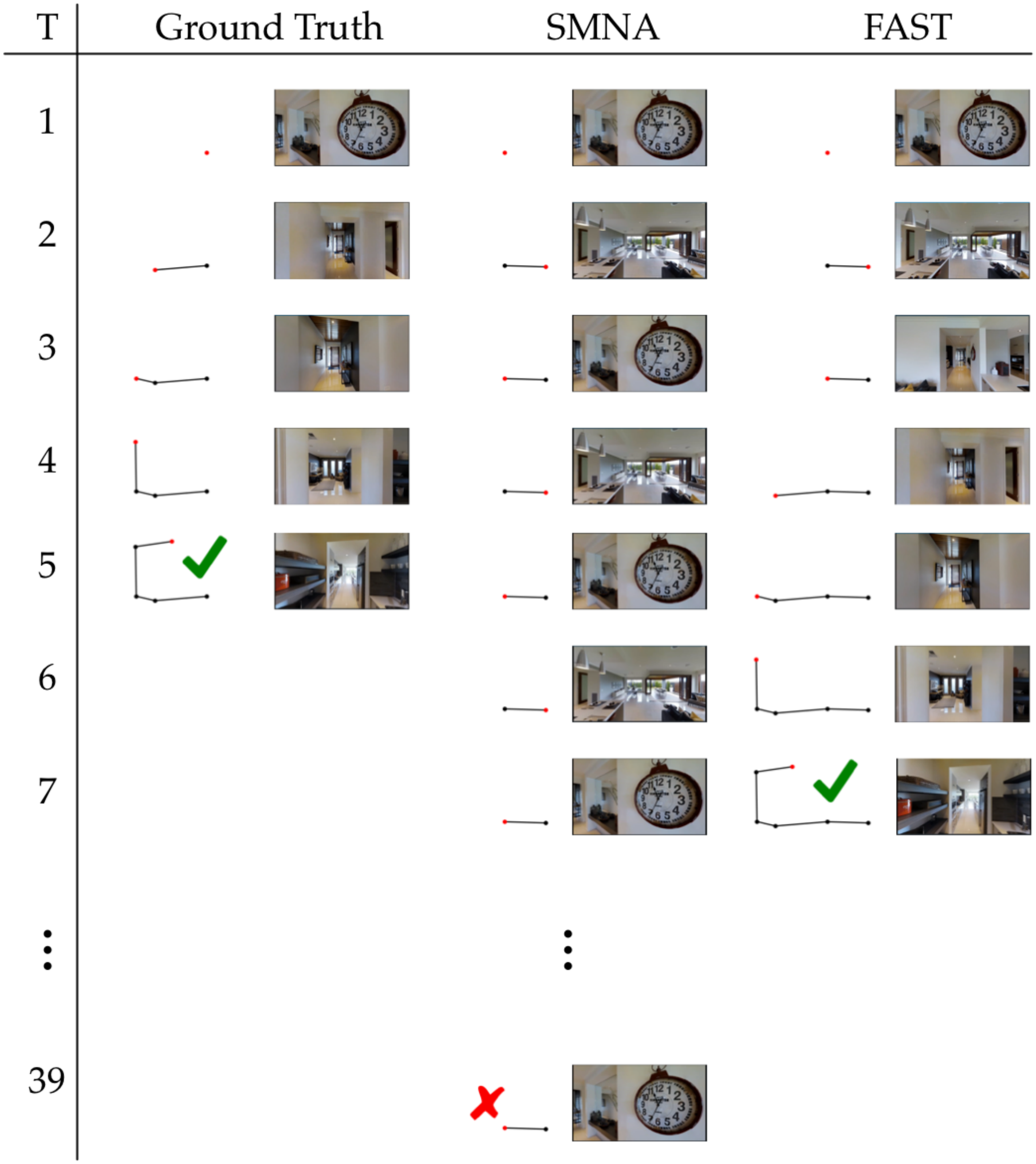

Figure A1. Comparison of the previously state-of-the-art SMNA model [13] to our FAST NAVIGATOR method, with the ground truth as reference. Note how SMNA retraces its steps multiple times due to the lack of global information. This example is taken from Room-to-Room, path 2617, instruction set 3. You can view a video of this trajectory here: https://youtu.be/AD9TNohXoPA.

Instructions: Walk out of the bedroom through the open door into the hallway. Turn the corner and walk into the dining area. Pass the dining table and walk into the living room area towards the television. Stop near the chair and open sliding doors to outside.

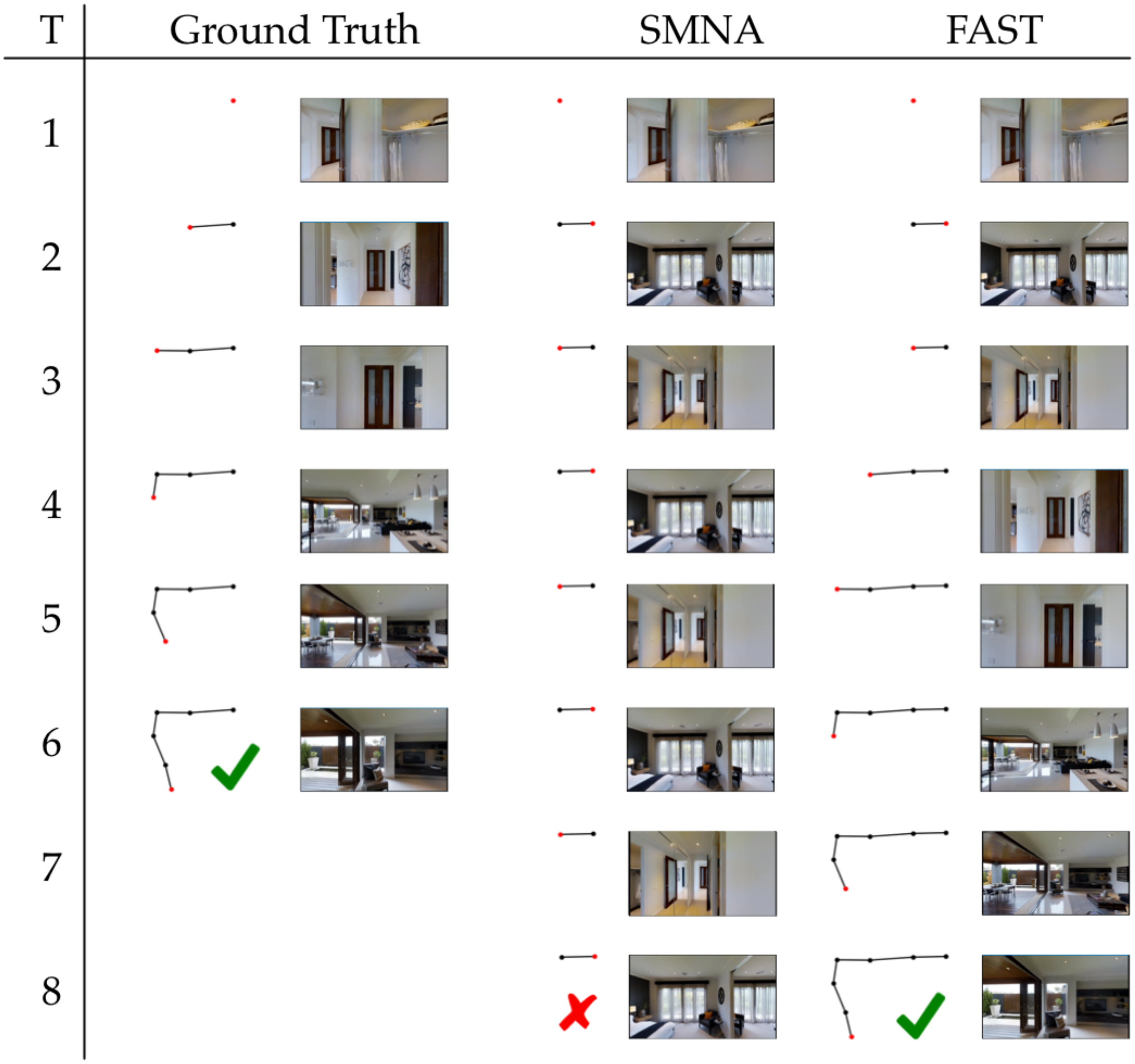

Figure A2. Identical to previous figure A1, except that this example is taken from Room-to-Room, path 15, instruction set 1.

Instructions: Walk towards the television, turn around and walk down the stairs directly to your left. Stop direclty at the bottom of the stairs.

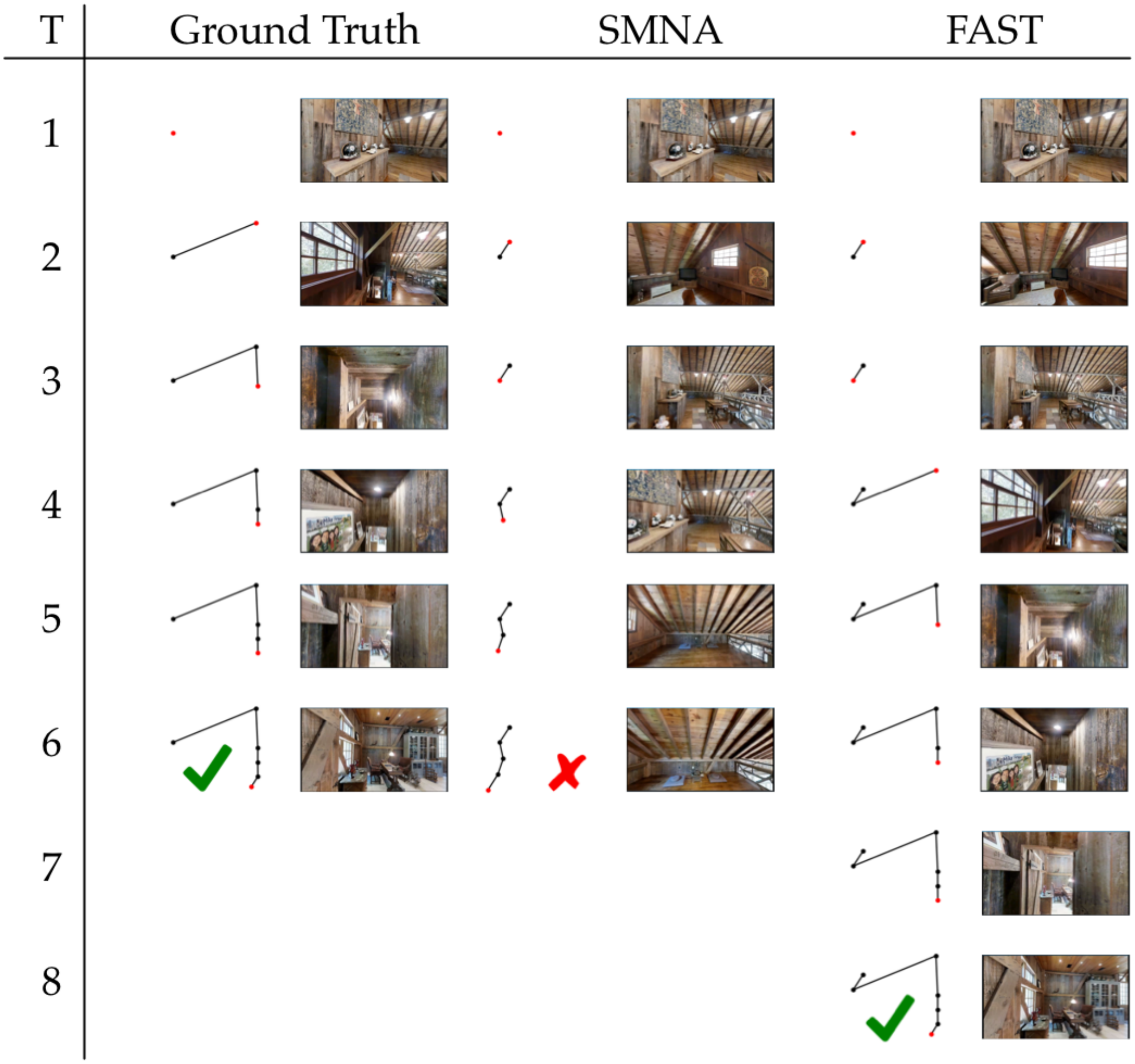

Figure A3. Identical to previous figure A1, except that this example is taken from Room-to-Room, path 1759, instruction set 1. The typo ’direclty’ comes from the dataset.