NavM🌍rph: A Self-Evolving World Model for Vision-and-Language Navigation in Continuous Environments

Supplementary Material

§ 1 provides a detail lower bound derivation for the defined loss $\mathcal{L}_W$. § 2 presents more details about evaluation metrics. Implementation details for experiments are provided in § 3, and further comparisons against state-of-the-art methods are shown in § 4. Finally, we present several visualizations for qualitative analysis in § 5.

1. Lower Bound Derivation

Following the predictive network described in § 3 of our main paper, the joint probability distribution for our proposed world model can be factorized as:

$$
\begin{aligned}
&p(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}, \boldsymbol{x}_{1:T+T_p} \boldsymbol{a}_{1:T+T_p}) = \\
&\prod_{t=1}^{T} p(\boldsymbol{h}_t, \boldsymbol{s}_t | \boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1}, \boldsymbol{a}_{t-1}) p(\boldsymbol{x}_t, \boldsymbol{a}_t | \boldsymbol{h}_t, \boldsymbol{s}_t) \\
&\prod_{j=1}^{T_p} p(\boldsymbol{h}_{T+j}, \boldsymbol{s}_{T+j} | \boldsymbol{h}_T, \boldsymbol{s}_T, \boldsymbol{a}_{T+j-1}) p(\boldsymbol{x}_{T+j}, \boldsymbol{a}_{T+j} | \boldsymbol{h}_T, \boldsymbol{s}_T),
\end{aligned}
\tag{1}
$$

For the first item for step 1 to T, we have:

$$
\begin{aligned}
p(\boldsymbol{h}_t, \boldsymbol{s}_t | \boldsymbol{h}_{t-1}, &\boldsymbol{s}_{t-1}, \boldsymbol{a}_{t-1}) = \\
&p(\boldsymbol{h}_t | \boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1}) p(\boldsymbol{s}_t | \boldsymbol{h}_t, \boldsymbol{a}_{t-1}),
\end{aligned}
\tag{2}
$$

$$
p(\boldsymbol{x}_t, \boldsymbol{a}_t | \boldsymbol{h}_t, \boldsymbol{s}_t) = p(\boldsymbol{x}_t | \boldsymbol{h}_t, \boldsymbol{s}_t) p(\boldsymbol{a}_t | \boldsymbol{h}_t, \boldsymbol{s}_t), \tag{3}
$$

Given that $\boldsymbol{h}_t$ is deterministic as discussed earlier, we have $p(\boldsymbol{h}_t | \boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1}) = \delta\left(\boldsymbol{h}_t - f_\theta(\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})\right)$. Therefore, we need to infer the latent variables $\boldsymbol{s}_{1:T}$. Since no observations are available during the prediction phase $[T+1 : T+T_p]$, the inference process focuses on maximizing the marginal likelihood over the observed data $p(\boldsymbol{x}_{1:T}, \boldsymbol{a}_{1:T})$. Based on deep variational inference, we introduce a variational distribution $q_{H,S}$ and factorize as follows, for we assume that independence of $(\boldsymbol{x}_{1:T}, \boldsymbol{a}_{1:T})$ given $(\boldsymbol{o}_{1:T}, \boldsymbol{a}_{1:T-1})$:

$$
\begin{aligned}
q_{H,S} &\triangleq q(\boldsymbol{h}_{1:T+T_p}, \boldsymbol{s}_{1:T+T_p} | \boldsymbol{o}_{1:T}, \boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p}) \\
&\triangleq q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} | \boldsymbol{o}_{1:T}, \boldsymbol{x}_{1:T}, \boldsymbol{a}_{1:T}) \\
&\triangleq q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} | \boldsymbol{o}_{1:T}, \boldsymbol{a}_{1:T-1}) \\
&= \prod_{t=1}^{T} q(\boldsymbol{h}_t | \boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1}) q(\boldsymbol{s}_t | \boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1}),
\end{aligned}
\tag{4}
$$

with $q(\boldsymbol{h}_t | \boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1}) = p(h_t | \boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})$ and $q_1(\boldsymbol{h}_1) = \delta(\boldsymbol{0})$. The Kullback-Leibler (KL) divergence between the prior and posterior distributions can be calculated as:

$$
\begin{aligned}
&D_{\mathrm{KL}}(q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} | \boldsymbol{o}_{1:T}, \boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p}) \\
&\qquad \| \; p(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} | \boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p})) \\
&= \mathbb{E}_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} \sim q_{H,S}} \left[\log \frac{q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} | \boldsymbol{o}_{1:T}, \boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p})}{p(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} | \boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p})} \right] \\
&= \mathbb{E}_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} \sim q_{H,S}} \left[\log \frac{q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} | \boldsymbol{o}_{1:T}, \boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p})}{p(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} | \boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p})} \right] \\
&= \mathbb{E}_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} \sim q_{H,S}} \\
&\left[\log \frac{q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} | \boldsymbol{o}_{1:T}, \boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p}) p(\boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p})}{p(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} | \boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p}) p(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T})} \right] \\
&= \log p(\boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p}) \\
&- \mathbb{E}_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} \sim q_{H,S}} \left[\log p(\boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p} | \boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}) \right] \\
&+ D_{\mathrm{KL}} \left(q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} | \boldsymbol{o}_{1:T}, \boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p}) \, \| \, p(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}) \right).
\end{aligned}
\tag{5}
$$

Since $D_{KL} \geq 0$, the left side of Eq. (5) should be non-negative. Based on Jensen's inequality [22], a variational lower bound on the log evidence can be obtained as follows:

$$
\begin{aligned}
&\log p(\boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p}) \geq \\
&\quad \mathbb{E}_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} \sim q_{H,S}} \left[\log p(\boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p} | \boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}) \right] - \\
&\quad D_{\mathrm{KL}} \left(q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} | \boldsymbol{o}_{1:T}, \boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p}) \, \| \, p(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}) \right).
\end{aligned}
\tag{6}
$$

As for the first term of the lower bound in Eq. (6):

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} \sim q_{H,S}} \left[\log p(\boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p} | \boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}) \right] \\
&= \mathbb{E}_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} \sim q_{H,S}} \Bigg[\log \prod_{t=1}^{T} p(\boldsymbol{x}_t | \boldsymbol{h}_t, \boldsymbol{s}_t) p(\boldsymbol{a}_t | \boldsymbol{h}_t, \boldsymbol{s}_t) \\
&\quad \prod_{j=1}^{T_p} p(\boldsymbol{h}_{T+j}, \boldsymbol{s}_{T+j} | \boldsymbol{h}_T, \boldsymbol{s}_T, \boldsymbol{a}_{T+j-1}) p(\boldsymbol{x}_{T+j}, \boldsymbol{a}_{T+j} | \boldsymbol{h}_T, \boldsymbol{s}_T) \Bigg] \\
&= \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{h}_{1:t}, \boldsymbol{s}_{1:t} \sim q(\boldsymbol{h}_t, \boldsymbol{s}_t | \boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})} [\log p(\boldsymbol{x}_t | \boldsymbol{h}_t, \boldsymbol{s}_t) \\
&\quad + \log p(\boldsymbol{a}_t | \boldsymbol{h}_t, \boldsymbol{s}_t)] + \sum_{j=1}^{T_p} \mathbb{E}_{\boldsymbol{h}_T, \boldsymbol{s}_T \sim q(\boldsymbol{h}_T, \boldsymbol{s}_T | \boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})} \\
&\quad [\log p(\boldsymbol{x}_{T+j} | \boldsymbol{h}_T, \boldsymbol{s}_T) + \log p(\boldsymbol{a}_{T+j} | \boldsymbol{h}_T, \boldsymbol{s}_T)],
\end{aligned}
\tag{7}
$$

where Eq. (7) is obtained by integrating over remaining latent variables $(\boldsymbol{h}_{t:1+T}, \boldsymbol{s}_{t:1+T})$.

Regarding the second term of the lower bound in Eq. (6), since there are no observations available during the prediction phase $[T+1 : T+T_p]$, the posterior distribution q is no longer updated with new input information. Consequently, it converges to the prior distribution, making the KL divergence between the posterior $q(\boldsymbol{h}_{T:T+T_p}, \boldsymbol{s}_{T:T+T_p})$ and the prior $p(\boldsymbol{h}_{T:T+T_p}, \boldsymbol{s}_{T:T+T_p})$ equal to zero. As a result, only the KL divergence for the observed phase $[1 : T]$ needs to be considered, which can be calculated according to Eq. (4):

$$
\begin{aligned}
&D_{\mathrm{KL}}\left(q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}|\boldsymbol{o}_{1:T}, \boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p}) \,\|\, p(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T})\right)\\
&\triangleq D_{\mathrm{KL}}\left(q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}|\boldsymbol{o}_{1:T}, \boldsymbol{a}_{1:T-1}) \,\|\, p(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T})\right)\\
&=\mathbb{E}_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T} \sim q_{H,S}}\left[\log \frac{q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}|\boldsymbol{o}_{1:T}, \boldsymbol{a}_{1:T-1})}{p(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T})}\right]\\
&=\int_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}} q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}|\boldsymbol{o}_{1:T}, \boldsymbol{a}_{1:T-1})\\
&\quad\left(\log \frac{q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}|\boldsymbol{o}_{1:T}, \boldsymbol{a}_{1:T-1})}{p(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T})}\right) \mathrm{d}\boldsymbol{h}_{1:T}\, \mathrm{d}\boldsymbol{s}_{1:T}\\
&=\int_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}} q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}|\boldsymbol{o}_{1:T}, \boldsymbol{a}_{1:T-1})\\
&\quad\log\left[\prod_{t=1}^{T} \frac{q(\boldsymbol{h}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})q(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})}{p(\boldsymbol{h}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})p(\boldsymbol{s}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})}\right] \mathrm{d}\boldsymbol{h}_{1:T}\, \mathrm{d}\boldsymbol{s}_{1:T}\\
&=\int_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}} \left(\prod_{t=1}^{T} q(\boldsymbol{h}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})q(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})\right)\\
&\quad\left(\sum_{t=1}^{T} \log \frac{q(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})}{p(\boldsymbol{s}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})}\right) \mathrm{d}\boldsymbol{h}_{1:T}\, \mathrm{d}\boldsymbol{s}_{1:T}.
\end{aligned}
\tag{8}
$$

Based on the above deduction, we iteratively integrate out each latent variable and, by recursively applying this process to the sum of logarithmic terms indexed by t, decompose the KL divergence into a summation over time steps.

$$
\begin{aligned}
&D_{\mathrm{KL}}\left(q(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}|\boldsymbol{o}_{1:T}, \boldsymbol{x}_{1:T}, \boldsymbol{a}_{1:T}) \,\|\, p(\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T})\right)\\
&=\int_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}} \left(\prod_{t=1}^{T} q(\boldsymbol{h}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})q(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})\right)\\
&\quad\left(\log \frac{q(\boldsymbol{s}_1|\boldsymbol{o}_1)}{p(\boldsymbol{s}_1)} + \sum_{t=2}^{T} \log \frac{q(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})}{p(\boldsymbol{s}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})}\right) \mathrm{d}\boldsymbol{h}_{1:T}\, \mathrm{d}\boldsymbol{s}_{1:T}\\
&=\mathbb{E}_{\boldsymbol{s}_1 \sim q(\boldsymbol{s}_1|\boldsymbol{o}_1)}\left[\log \frac{q(\boldsymbol{s}_1|\boldsymbol{o}_1)}{p(\boldsymbol{s}_1)}\right]\\
&\quad+\int_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}} \left(\prod_{t=1}^{T} q(\boldsymbol{h}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})q(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})\right)\\
&\quad\left(\sum_{t=2}^{T} \log \frac{q(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})}{p(\boldsymbol{s}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})}\right) \mathrm{d}\boldsymbol{h}_{1:T}\, \mathrm{d}\boldsymbol{s}_{1:T}\\
&=D_{\mathrm{KL}}\left(q(\boldsymbol{s}_1|\boldsymbol{o}_1) \,\|\, p(\boldsymbol{s}_1)\right)\\
&\quad+\int_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}} \left(\prod_{t=1}^{T} q(\boldsymbol{h}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})q(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})\right)\\
&\quad\left(\log \frac{q(\boldsymbol{s}_2|\boldsymbol{o}_{1:2}, \boldsymbol{a}_1)}{p(\boldsymbol{s}_2|\boldsymbol{h}_1, \boldsymbol{s}_1)} + \sum_{t=3}^{T} \log \frac{q(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})}{p(\boldsymbol{s}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})}\right) \mathrm{d}\boldsymbol{h}_{1:T}\, \mathrm{d}\boldsymbol{s}_{1:T}\\
&=D_{\mathrm{KL}}\left(q(\boldsymbol{s}_1|\boldsymbol{o}_1) \,\|\, p(\boldsymbol{s}_1)\right)+\\
&\quad\mathbb{E}_{\boldsymbol{h}_1, \boldsymbol{s}_1 \sim q(\boldsymbol{h}_1, \boldsymbol{s}_1|\boldsymbol{o}_1)}\left[D_{\mathrm{KL}}\left(q(\boldsymbol{s}_2|\boldsymbol{o}_{1:2}, \boldsymbol{a}_1) \,\|\, p(\boldsymbol{s}_2|\boldsymbol{h}_1, \boldsymbol{s}_1)\right)\right]\\
&\quad+\int_{\boldsymbol{h}_{1:T}, \boldsymbol{s}_{1:T}} \left(\prod_{t=1}^{T} q(\boldsymbol{h}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})q(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})\right)\\
&\quad\left(\sum_{t=3}^{T} \log \frac{q(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})}{p(\boldsymbol{s}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})}\right) \mathrm{d}\boldsymbol{h}_{1:T}\, \mathrm{d}\boldsymbol{s}_{1:T}\\
&=\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1} \sim q(\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1}|\boldsymbol{o}_{1:t-1}, \boldsymbol{a}_{1:t-2})}\\
&\qquad\left[D_{\mathrm{KL}}\left(q(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1}) \,\|\, p(\boldsymbol{s}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})\right)\right]
\end{aligned}
\tag{9}
$$

Combining Eq. (8), Eq. (9) and Eq. (6), the final lower bound can be obtained as follows:

$$
\begin{aligned}
&\log p(\boldsymbol{x}_{1:T+T_p}, \boldsymbol{a}_{1:T+T_p}) \geq\\
&\quad\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{h}_{1:t}, \boldsymbol{s}_{1:t} \sim q(\boldsymbol{h}_t, \boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})}[\log p(\boldsymbol{x}_t|\boldsymbol{h}_t, \boldsymbol{s}_t)\\
&\quad+\log p(\boldsymbol{a}_t|\boldsymbol{h}_t, \boldsymbol{s}_t)] + \sum_{j=1}^{T_p} \mathbb{E}_{\boldsymbol{h}_T, \boldsymbol{s}_T \sim q(\boldsymbol{h}_T, \boldsymbol{s}_T|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})}\\
&\qquad[\log p(\boldsymbol{x}_{T+j}|\boldsymbol{h}_T, \boldsymbol{s}_T) + \log p(\boldsymbol{a}_{T+j}|\boldsymbol{h}_T, \boldsymbol{s}_T)],\\
&\quad-\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1} \sim q(\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1}|\boldsymbol{o}_{1:t-1}, \boldsymbol{a}_{1:t-2})}\\
&\qquad\left[D_{\mathrm{KL}}\left(q(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1}) \,\|\, p(\boldsymbol{s}_t|\boldsymbol{h}_{t-1}, \boldsymbol{s}_{t-1})\right)\right]
\end{aligned}
\tag{10}
$$

2. Evaluation Metrics for VLN-CE agents

We follow previous approaches [4, 5, 20] and adopt the standard metrics for evaluating VLN-CE agents:

- TL (Trajectory length) measures the average length of the predicted navigation trajectories.
- NE (Navigation Error) measures the average distance (in meter) between the agent's final position in the predicted trajectory and the target in the ground truth.
- SR (Success Rate) is the proportion of the agent stopping in the predicted route within a threshold distance (set as 3 meters) of the goal in the reference route.
- OSR (Oracle Success Rate) is the proportion of the closest point in the predicted trajectory to the target in the reference trajectory within a threshold distance.

- SPL (Success weighted by Path Length))is a comprehensive metric method integrating SR and TL that takes both effectiveness and efficiency into account.
- NDTW (Normalized Dynamic Time Warping) measures the normalized cumulative distance between reference path and agent position.
- SDTW (Success weighted by normalized Dynamic Time Warping) is a comprehensive metric method integrating NDTW and SR that takes both path efficiency and task completion into account.

3. Implementation Details

The Baseline Framework. In conventional panoramic VLN-CE frameworks [3, 36], the agent perceives its surroundings through multi-view RGB-D panoramas captured at 30-degree intervals at each timestep t. These panoramic observations are processed by a trained waypoint prediction module[18] to identify navigable waypoints. The VLN model then encodes both the visual features of these waypoints and their spatial information (relative direction and distance) to construct a topological map. This map is subsequently integrated with the navigation instruction via the Cross-Modal Graph Transformer[3, 11], which selects the optimal waypoint as the agent's next navigation goal.

Monocular VLN-CE settings rely on a single RGB-D camera, which presents challenges in waypoint estimation due to the lack of full panoramic coverage. To address this, an enhanced waypoint predictor [37] utilizes a semantic traversability map and 3D feature fields to infer viable waypoints, ensuring effective decision-making even with limited field-of-view.

Model Configuration. Following the previous baseline model [3, 37], we utilize CLIP-pretrained ViT-B/32 [13] for RGB feature extraction, while depth information is processed through a point-goal navigation pretrained ResNet-50 [17]. The framework maintains encoder depths of 2, 9, and 4 layers for panoramic, textual, and cross-modal graph components respectively, aligned with [15, 18]. Other hyperparameters are the same as LXMERT [31] on the R2R-CE dataset and pre-trained RoBerta [27] for the multilingual RxR-CE dataset. The camera's HFOV is set to 90° for R2R-CE and 79° for RxR-CE.

Experimental Details. NavMorph was trained over 10K episodes on the R2R-CE dataset and 20K episodes on the RxR-CE dataset, following the same initialization and training strategies as the pretrained baseline [3, 37]. The learning rate is set to 1×10^{-5}, while the weighting coefficient for loss function $\mathcal{L}$ is $\gamma = 10^{-3}$. Note that the weighting coefficients are heuristically adjusted to balance each loss term, ensuring they remain at the same order of magnitude based on initial values.

At each timestep of a navigation task, the model predicts actions for $T_p = 2$ consecutive future states, starting from $t = 1$ (*i.e.*, the next position after the agent's initial point). Accordingly, the observation window T dynamically expands throughout the navigation process, increasing until the agent selects 'stop' action or reaches the maximum step limit. For input image dimensions $N_o \times h \times w \times c$, we set $1 \times 224 \times 224 \times 3$ for monocular settings and $12 \times 224 \times 224 \times 3$ for panoramic settings. The encoded visual embedding has a dimension of $d_x = 512$, while both scene-contextual features (d_v) and action embedding (d_a) are set to 768. Our Contextual Evolution Memory (CEM) is initially randomized and progressively updated with informative scene-contextual features. These features are derived from panoramic visual representations extracted by the panoramic encoder during training, encapsulating comprehensive environmental information to enhance navigation. The memory size N_m is set to 1000. For the monocular setting, K is set to 16 for top-K retrieval with update factors $\alpha = \beta = 0.7$. For panoramic setting, K is set to 10 for top-K retrieval with update factors $\alpha = \beta = 0.9$.

Working Modes. During the ***training*** phase, NavMorph operates through two core components: the world-aware navigator, which executes navigation actions for VLN-CE tasks, and the foresight action planner, which performs imaginative rollouts for future T_p steps. This collaborative framework enables the model to learn effective navigation strategies while simultaneously refining its latent state representation capabilities. During the ***online testing*** phase, the world-aware navigator performs navigation planning by leveraging the predicted future actions generated by the foresight action planner as guidance. Specifically, the navigator evaluates each candidate waypoint by assigning navigation scores based on the learned policy, which are subsequently refined according to their proximity to the predicted trajectory points. This weighting strategy prioritizes candidates closer to the predicted path, seamlessly integrating the foresight planner's predictions into final navigation decision-making process.

4. Complementary Experiments

4.1. Full Results

In our main paper, we provide representative comparison results on the R2R-CE [5, 24] and RxR-CE [24, 25] benchmarks due to space constraints. Here, we present the complete results across the 'validation seen', 'validation unseen', and 'test unseen' splits of these benchmarks, including comparisons with a broader range of state-of-the-art methods, as detailed in Table 1 and Table 2. Our self-evolving world model enhances its ability to anticipate future states based on current observations and cross-episodic experiences, effectively handling complex navigation tasks even with monocular input.

Performance Improvement on Seen/Unseen sets. Based

Table 1. Experimental results on R2R-CE dataset. Results better than base model are shown in blue. Best results for the panoramic and monocular settings are each highlighted in **bold**. * indicates experimental results that we have reproduced in this work.

Camera	Methods	Val Seen					Val Unseen					Test Unseen				
		TL ↓	NE ↓	OSR	SR	SPL	TL ↓	NE ↓	OSR	SR	SPL	TL ↓	NE ↓	OSR	SR	SPL
Monocular	LAW [30]	**9.34**	6.35	49	40	37	**8.89**	6.83	44	35	31	**9.67**	7.69	28	38	25
	CM^2 [15]	12.05	6.10	50.7	42.9	34.8	11.54	7.02	41.5	34.3	27.6	13.90	7.70	39	31	24
	WS-MGMap [9]	10.12	5.65	51.7	46.9	**43.4**	10.00	6.28	47.6	38.9	34.3	12.30	7.11	45	35	28
	NaVid [39]	-	-	-	-	-	-	**5.47**	49.1	37.4	**35.9**	-	-	-	-	-
	ETPNav/p [37]	-	-	-	-	-	-	6.81	42.4	32.9	23.1	-	-	-	-	-
	VLN-3DFF [37]	-	-	-	-	-	-	5.95	55.8	44.9	30.4	-	6.24	54.4	43.7	28.9
	VLN-3DFF*	22.90	4.92	62.1	52.7	36.7	26.16	6.05	54.9	43.8	29.4	26.02	6.22	**54.7**	43.8	28.6
	NavMorph	20.03	**4.58**	**62.7**	**55.8**	38.9	22.54	5.75	**56.9**	**47.9**	33.2	24.75	**6.01**	54.5	**45.7**	**30.2**
Panoramic	Seq2Seq [5]	9.26	7.12	46	37	35	8.64	7.37	40	32	30	8.85	7.91	36	28	25
	SASRA [21]	8.89	7.71	-	36	34	7.89	8.32	-	24	22	-	-	-	-	-
	CWTP [8]	-	7.10	56	36	31	-	7.90	38	26	23	-	-	-	-	-
	AG-CMTP [6]	-	6.60	56	36	31	-	7.90	39	23	19	-	-	-	-	-
	R2R-CMTP [6]	-	7.10	45	36	31	-	7.90	38	26	23	-	-	-	-	-
	WPN [34]	**8.54**	5.48	53	46	43	**7.62**	6.31	40	36	34	**8.02**	6.65	37	32	30
	Sim2Sim [23]	11.18	4.67	61	52	44	10.69	6.07	52	43	36	11.43	6.17	52	44	37
	CWP-CMA [18]	11.47	5.20	61	51	45	10.90	6.20	52	41	36	11.85	6.30	49	38	33
	CWP-BERT [18]	12.50	5.02	59	50	44	12.23	5.74	53	44	39	13.51	5.89	51	42	36
	ERG [34]	11.80	5.04	61	46	42	9.96	6.20	52	41	36	-	-	-	-	-
	DUET [11]	12.62	4.13	67	57	49	11.86	5.13	55	46	40	13.13	5.82	50	42	36
	DREAMW [33]	11.60	4.09	59	66	48	11.30	5.53	49	59	44	11.80	5.48	49	57	44
	Ego^2-Map [19]	-	-	-	-	-	-	4.93	-	52	46	-	5.54	56	47	41
	ScaleVLN [35]	-	-	-	-	-	-	4.80	-	55	51	-	5.11	-	55	50
	GridMM [36]	12.69	4.21	69	59	51	13.36	5.11	61	49	41	13.31	5.64	56	46	39
	BEVBert [1]	13.98	3.77	73	68	60	13.27	4.57	67	59	50	15.31	4.70	67	59	50
	FSTTA [14]	12.39	4.25	69	58	50	11.58	5.27	58	48	42	13.17	5.84	55	46	38
	ETPNav [3]	11.78	3.95	72	66	59	11.99	4.71	65	57	49	12.87	5.12	63	55	48
	ETPNav*	11.35	3.93	72	66	59	11.40	4.69	64	57	49	12.72	5.10	63	55	48
	NavMorph	11.43	3.86	73	67	60	11.55	4.62	66	59	50	12.88	4.91	64	57	49
	HNR [38]	11.79	3.67	76	69	61	12.64	4.42	67	61	51	13.03	4.81	67	58	50
	HNR*	11.84	3.73	76	69	61	12.76	4.57	67	61	51	12.92	4.85	67	58	50
	NavMorph	11.76	**3.66**	**78**	**70**	**62**	12.68	**4.37**	**68**	**64**	**53**	12.69	**4.69**	**68**	**60**	**52**

Note: Following established conventions in prior works, we report experimental results with different precision formats across camera configurations: integers for panoramic settings and two decimal places for monocular settings.

on the experimental results in Table 1 and Table 2, our proposed NavMorph consistently achieves notable performance improvements across different datasets. While performance gains varies between seen and unseen environments, we analyze relative improvements to better quantify the effectiveness of our self-evolving world model across different settings.

Taking the monocular setup as an example, NavMorph improves the success rate (SR) by 6.85% in unseen environments, compared to 5.88% in seen environments on R2R-CE dataset. The improvement in SPL is even more pronounced, reaching 9.26% in unseen environments versus 5.99% in seen environments. A similar trend is observed in RxR-CE, where unseen SR improves by 10.94%, while seen SR increases by 7.54%. Likewise, SPL improves 11.29% in unseen settings, compared to 12.71% in seen ones. These results indicate that NavMorph achieves higher or comparable performance gains in unseen environments (average of val/test unseen) compared to seen ones, demonstrating its capacity to generalize across novel tasks.

A key factor contributing to this generalization ability is self-evolution, which enhances adaptation uniformly across both seen and unseen data rather than specifically optimizing for new scenarios. The observed gains in seen settings further suggest that the model effectively adapts to novel instructions within familiar scenes, rather than merely overfitting to training data.

4.2. Extended Results for Self-Evolution

Detailed Ablation Study on Self-Evolution Strategy. In our main paper (*Table 3*), we conducted an ablation study on the effect of self-evolution, in which the proposed Contextual Evolution Memory (CEM) module was entirely prevented from updating. The results demonstrated the effectiveness of self-evolution in enhancing model performance and learning dynamics. To further investigate its role only in online adaptation, we introduce 'NavMorph *w/o* SE*', a variant where CEM undergoes self-evolution following *Eq. (3) in main paper* during training, progressively refining its stored representations. Once training is complete,

Table 2. Experimental results on RxR-CE datasets. Results better than the base model are shown in blue. Best results for the panoramic and monocular settings are each highlighted in bold.

Camera	Methods	Val Seen							Val Unseen							Test Unseen						
		TL ↓	NE ↓	OSR	SR	SPL	NDTW	SDTW	TL ↓	NE ↓	OSR	SR	SPL	NDTW	SDTW	TL ↓	NE ↓	OSR	SR	SPL	NDTW	SDTW
Monocular	LAW [30]	**7.92**	11.94	20.0	7.0	6.0	-	-	**4.01**	10.87	21.0	8.0	8.0	-	-	-	-	-	-	-	-	-
	CM^2 [15]	-	-	-	-	-	-	-	12.29	8.98	25.3	14.4	9.2	-	-	-	-	-	-	-	-	-
	WS-MGMap [9]	10.37	10.19	27.7	14.0	12.3	-	-	10.80	9.83	29.8	15.0	12.1	-	-	-	-	-	-	-	-	-
	NaVid [39]	-	-	-	-	-	-	-	10.59	**8.41**	34.5	23.8	32.2	-	-	-	-	-	-	-	-	-
	A^2-Nav [10]	-	-	-	-	-	-	-	-	-	-	16.8	6.3	-	-	-	-	-	-	-	-	-
	VLN-3DFF [37]	-	-	-	-	-	-	-	-	8.79	36.7	25.5	18.1	-	-	-	-	-	-	-	-	-
	VLN-3DFF*	18.91	9.87	40.54	27.72	20.61	42.37	20.94	16.21	9.41	38.40	26.66	20.11	42.91	20.36	**20.85**	10.19	-	23.41	15.43	32.38	14.75
	NavMorph	21.61	**9.80**	**41.27**	**29.81**	**23.23**	**44.51**	**22.68**	20.28	8.85	**43.05**	**30.76**	**22.84**	**44.19**	**23.30**	21.13	**9.81**	-	**24.93**	**16.82**	**33.71**	**15.64**
Panoramic	Seq2Seq [5]	-	-	-	-	-	-	-	7.33	12.1	-	13.93	11.96	30.86	11.01	-	12.10	-	13.93	11.96	30.86	11.01
	Reborn [2]	-	5.69	-	52.43	45.46	66.27	44.47	-	5.98	-	48.60	42.05	63.35	41.82	-	7.10	-	45.82	38.82	55.43	38.42
	CWP-CMA [18]	-	-	-	-	-	-	-	-	8.76	-	26.59	22.16	47.05	-	**20.04**	10.4	-	24.08	19.07	37.39	18.65
	CWP-RecBERT [18]	-	-	-	-	-	-	-	-	8.98	-	27.08	22.65	46.71	-	20.09	10.4	-	24.85	19.61	37.30	19.05
	AO-Planner [7]	-	-	-	-	-	-	-	-	7.06	-	43.3	30.5	50.1	-	-	-	-	-	-	-	-
	LAW-Pano [30]	**6.27**	12.07	17.0	9.0	9.0	-	-	**4.62**	11.04	16.0	10.0	9.0	-	-	-	-	-	-	-	-	-
	UnitedVLN [12]	-	**4.74**	-	**65.1**	52.9	69.4	53.6	-	**5.48**	-	57.9	45.9	63.9	48.1	-	-	-	-	-	-	-
	ETPNav [3]	-	5.03	-	61.46	50.83	66.41	51.28	-	5.64	-	54.79	44.89	61.90	45.33	-	6.99	-	51.21	39.86	54.11	41.30
	ETPNav*	18.16	5.06	64.06	62.09	50.64	66.06	51.17	18.92	5.96	63.66	54.83	44.62	61.36	44.87	21.83	6.92	-	51.38	39.90	53.85	40.91
	NavMorph	18.97	5.08	65.86	63.88	52.28	67.94	52.54	19.93	5.80	64.83	56.23	46.39	63.23	46.98	21.29	6.90	-	51.97	41.56	55.01	42.60
	HNR [38]	-	4.85	-	63.72	53.17	68.81	52.78	-	5.51	-	56.39	46.73	63.56	47.24	-	6.81	-	53.22	41.14	55.61	42.89
	HNR*	19.74	4.93	66.01	63.55	53.37	69.02	52.66	20.41	5.75	64.93	56.48	46.62	63.43	47.38	23.02	6.88	-	53.33	41.18	55.47	42.95
	NavMorph	20.80	5.10	**67.88**	64.95	**54.17**	**70.94**	**54.82**	21.33	5.67	**66.02**	**58.02**	**48.98**	**64.77**	**48.85**	23.36	**6.67**	-	**54.98**	**43.02**	**57.31**	**44.76**

Note: Official evaluation on the Test Unseen split of RxR-CE dataset only provides TL, NE, SR, SPL, NDTW and SDTW metrics, thus OSR metric is not reported for the test split in this table.

the finalized memory is used as the initial state for deployment and remains unchanged throughout online testing.

As shown in Table 3, enabling self-evolution during online testing improves performance in online unseen environments, highlighting its crucial role in real-time adaptation. Moreover, since the self-evolution process benefits from prolonged environmental interaction—where unsupervised learning progressively refines the model's dynamic latent state—we extend our analysis to a larger, more diverse dataset, RxR-CE, to examine its influence on generalization. The results indicate a notable improvement in SPL (21.46→22.84), further validating the effectiveness of self-evolution in enhancing adaptability in unseen environments.

Different Steps of Predictive Future States T_p. As shown in Table 4, we further investigate the impact of varying predictive steps in our foresight action planner on navigational performance. Notably, predicting two steps ($T_p = 2$) achieves the optimal balance across key metrics, offering sufficient foresight for reliable decision-making without introducing excessive uncertainty. As T_p increases beyond 2, we observe a slight decline in SPL, SR, and NDTW, possibly due to compounding errors (accumulation of inaccuracies over multiple predictive steps) or over-commitment to future predictions (focuses too heavily on long-term predictions), which reduces the agent's flexibility to adapt to changing environments. These results demonstrate the need of striking a balance between foresight and adaptability. Predicting too few steps may limit the agent's strategic planning, while predicting too many steps introduces unnecessary complexity, diminishing trajectory efficiency.

4.3. Other Ablation Studies

Comparison with Representative TTA Strategies. In our main paper, we discussed how our evolving world model accumulates scene-specific information from test environments as memory knowledge during online testing. This mechanism allows the model to refine its predictions in dynamically changing environments without requiring ground truth actions. A related class of approaches, referred to as Test-Time Adaptation (TTA), also aims to improve model generalization by dynamically adjusting model parameters during testing, often through gradient-based updates or statistical alignment methods (*e.g.*, batch normalization adaptation, entropy minimization).

Given the comparison with the most related method, FSTTA [14], in our main paper, we further evaluate representative TTA approaches under the same test conditions as FSTTA (*i.e.*, updating the same set of parameters) and compare them with our NavMorph. To ensure a fair and robust evaluation, we conduct experiments under three different random seeds, reporting both the mean and standard deviation of the results. As demonstrated in Table 5, our proposed NavMorph achieves the best overall performance while exhibiting more stable results (with lower standard

Table 3. Ablation Study on Self-Evolution.

Dataset	Methods	TL ↓	NE ↓	OSR	SR	SPL	NDTW	SDTW
R2R-CE Val Unseen	Base Model	26.16	6.05	54.92	43.77	29.39	40.94	29.30
	NavMorph *w/o* SE*	23.33	5.77	56.12	46.87	32.56	44.42	32.16
	NavMorph	22.54	5.75	56.88	47.91	33.22	44.86	32.73
RxR-CE Val Unseen	Base Model	16.21	9.41	38.40	26.66	20.11	42.91	20.36
	NavMorph *w/o* SE*	20.83	9.08	41.49	28.78	21.46	43.26	21.52
	NavMorph	20.28	8.85	43.05	30.76	22.84	44.19	23.30

Note: Results better than 'NavMorph *w/o* SE*' are shown in blue.

Table 4. Experimental Results for Different Predictive Steps.

Methods	Predictive Steps T_p	R2R-CE Val Unseen						
		TL ↓	NE ↓	OSR	SR	SPL	NDTW	SDTW
Base model	-	26.16	6.05	54.92	43.77	29.39	40.94	29.30
NavMorph	1	22.05	5.99	55.57	46.06	32.78	**44.89**	32.36
	2	22.54	5.75	**56.88**	**47.91**	**33.22**	44.86	**32.73**
	3	25.36	5.99	56.50	44.97	31.30	42.99	30.57
	4	**20.91**	5.81	55.52	46.82	32.04	44.61	32.20
	5	25.94	**5.69**	56.66	47.18	31.79	43.72	31.92

Table 5. Comparison with Representative TTA Strategies.

Methods	R2R-CE Val Unseen						
	TL ↓	NE ↓	OSR	SR	SPL	NDTW	SDTW
Base Model	26.16	6.05	54.92	43.77	29.39	40.94	29.30
+ Tent [32]	28.56± 1.59	7.21± 1.01	52.13± 1.98	40.97± 1.77	27.46± 0.94	37.90± 1.53	27.65± 1.60
+ NOTE [16]	26.88± 1.82	6.71± 0.63	53.87± 1.71	42.85± 0.88	28.43± 0.56	39.02± 0.93	28.37± 0.88
+ SAR [29]	27.15± 1.40	6.57± 0.83	53.50± 1.30	43.02± 0.91	27.98± 0.72	38.77± 0.95	27.92± 0.75
+ ViDA [26]	26.74± 1.26	6.88± 0.75	55.26± 0.98	43.58± 0.86	28.29± 0.53	40.89± 0.94	28.73± 0.56
+ FSTTA [14]	28.25± 0.72	6.67± 0.34	55.41± 0.91	43.94± 0.32	29.63± 0.47	42.76± 0.65	29.34± 0.49
NavMorph	**22.54**± **0.07**	**5.75**± **0.03**	**56.88**± **0.05**	**47.91**± **0.04**	**33.22**± **0.02**	**44.86**± **0.07**	**32.73**± **0.04**

Note: The reported values represent the mean results, with the standard deviation provided in a reduced font size. Best results are shown in bold.

Table 6. Ablation Study of Action Embedding.

Methods	R2R-CE Val Unseen						
	TL ↓	NE ↓	OSR	SR	SPL	NDTW	SDTW
Base Model	26.16	6.05	54.92	43.77	29.39	40.94	29.30
Base-AE	26.01	6.09	55.1	43.4	29.3	41.34	29.42
NavMorph	**22.54**	**5.75**	**56.88**	**47.91**	**33.22**	**44.86**	**32.73**

Note: 'Base Model' denotes the chosen baseline under monocular setting, VLN-3DFF. 'Base-AE' denotes the baseline incorporating action information into the input without the world model. Best results are shown in bold.

deviation). These results highlight the effectiveness of incorporating a world model in VLN-CE tasks, as conventional TTA methods alone yield limited improvements, underscoring the necessity of structured world modeling for online adaptation to novel tasks.

Additionally, the configurations of these TTA strategies for VLN are detailed as follows:

- **Tent** [32]. We adopt all hyperparameter settings as specified in Tent. Specifically, the optimizer is AdamW [28], and for a batch size of 1, the learning rate is set to $0.001/64$.
- **NOTE** [16]. The hyperparameter configurations strictly follow those defined in NOTE. In particular, the soft-shrinkage width is set to 4, and the EMA momentum is 0.01. The optimization is performed using AdamW with a learning rate of 0.0001.
- **SAR** [29]. We adhere to the default hyperparameters in SAR. Specifically, the entropy constant E_0 (for reliable sample identification) is set to $0.4 \times \ln 1000$, while the neighborhood size for sharpness-aware minimization is configured as 0.05. For model recovery, the moving average factor is 0.9, and the reset threshold is 0.2.
- **ViDA** [26]. The experimental setup follows the original ViDA configuration. Random augmentation compositions, including Gaussian noise and dropout, are incorporated. The AdamW optimizer, identical to that in Tent, is utilized. The threshold value is set to 0.2, and the updating weight is 0.999.

Ablation Study of Action Embedding. In our proposed world model, action embedding $\boldsymbol{a}_t$ plays a crucial role in modeling state transitions and long-term predictive reasoning within RSSM, enabling the model to generate plausible future trajectories based on past observations and actions (*Section 3.1 in main paper*). Some readers may wonder whether the model's performance improvement arises from learning an action-state mapping through action embedding. To further investigate this, we conduct an ablation study to assess its role in latent representation learning. Specifically, we introduce a variant, 'Base-AE', which retains the same backbone as the baseline model (VLN-3DFF [37]) without world modeling but includes an additional action embedding input.

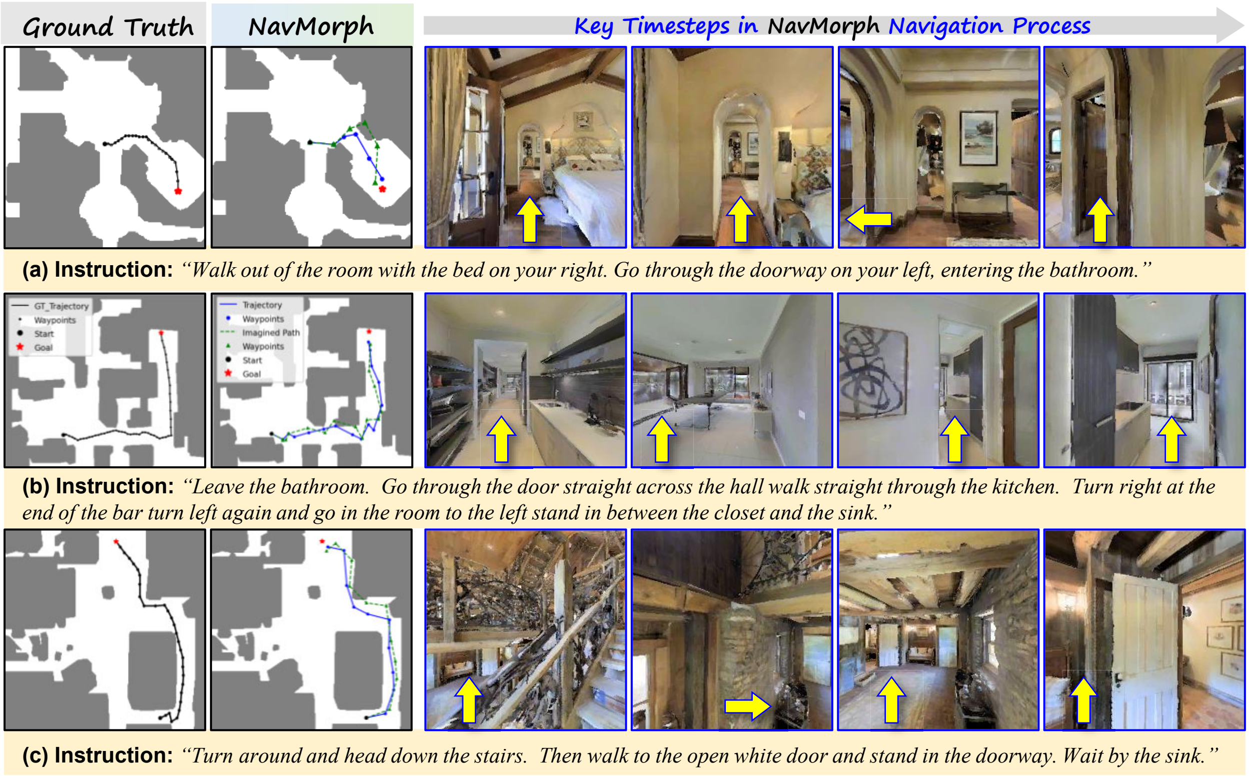
As shown in Table 6, the results indicate only marginal differences (OSR: 54.9→55.1, SR: 43.8→43.4, SPL: 29.4→29.3), indicating that explicitly encoding actions has a negligible effect on performance. This finding highlights that the observed improvements in our method stem primarily from the self-evolving world model's ability to model environmental dynamics, rather than the mere inclusion of action embedding.

Computational Analysis. *Table 4 in main paper* demonstrates that NavMorph maintains comparable inference efficiency to the base model, with an average episode time of 21.22s *vs* 20.53s, while achieving nearly 4% improvements in both SR and SPL. In terms of parameter overhead, our CEM introduces only a marginal increase—adding 2.30M parameters compared to the base model's 228.96M, accounting for merely 1.0% of the total model size. Given the consistent performance improvements, the computational and memory overheads are lightweight and acceptable.

4.4. Extensive Discussion

While our method shares similarities with test-time adaptation (TTA), NavMorph fundamentally extends beyond this paradigm. Traditional TTA typically applies gradient-based parameter updates during inference for static classification tasks. In contrast, our approach incorporates a contextual evolution mechanism (CEM) within the RSSM framework, explicitly modeling environment state transitions during both training and inference phases. This mechanism enables the agent to adapt proactively to dynamic environments—not just during inference—by selectively integrating new scene observations while retaining historically relevant information.

At the core of this design lies the Contextual Evolution Memory (CEM), which enhances long-term reasoning by dynamically maintaining latent scene representations. Rather than accumulating all past experiences, CEM performs top-K scene retrieval based on visual similarity, maintaining memory entries that are most pertinent to the current context. This suppresses noisy or outdated trajectories and enhances the agent's ability to infer plausible future transitions, particularly in out-of-distribution or evolving scenes, as often encountered in VLN-CE tasks. We provide ablations (Table 3) comparing variants of NavMorph with and without online evolution (NavMorph w/o SE*),



(a) Instruction: *"Walk out of the room with the bed on your right. Go through the doorway on your left, entering the bathroom."*

(b) Instruction: *"Leave the bathroom. Go through the door straight across the hall walk straight through the kitchen. Turn right at the end of the bar turn left again and go in the room to the left stand in between the closet and the sink."*

(c) Instruction: *"Turn around and head down the stairs. Then walk to the open white door and stand in the doorway. Wait by the sink."*

Figure 1. Qualitative results of NavMorph on the R2R-CE dataset are presented, showcasing a comparison between ground truth paths (GT_Trajectory), NavMorph's executed navigation routes (Trajectory), and the predictive paths generated by the Foresight Action Planner (Imagined Path). These visualizations highlight NavMorph's ability to perform effective navigation. Additionally, key input observations at critical timesteps during NavMorph's navigation are provided to illustrate its decision-making process.

confirming that the model remains effective even without runtime adaptation (SR: Base Model 43.77 → NavMorph w/o SE* 46.87 → NavMorph 47.91).

Importantly, our contribution lies not in the memory design itself, but in showing that adaptive evolution—when tightly integrated into a world model—can effectively improve navigation performance in VLN-CE literature.

5. Qualitative Analysis

To evaluate the predictive performance of NavMorph, we conduct qualitative analysis by comparing trajectories generated through our Foresight Action Planner with executed paths and ground truth sequences. Since our world model encodes high-level features instead of raw images, direct visualization of latent states remains non-trivial. Therefore, to effectively illustrate NavMorph's reasoning process, we resort to trajectory-based evaluations, where the predicted and executed navigation sequences serve as an implicit reflection of the model's latent space dynamics. Figure 1 presents trajectory visualizations across diverse navigational scenarios, where the coherence between predicted and executed paths demonstrates the model's capacity for environmental dynamics modeling and anticipatory planning. Furthermore, we visualize key observational inputs at critical navigation timesteps to provide insights into NavMorph's decision-making process.

We observe that in simple scenarios with clear navigation paths (Figure 1(a)), the predicted trajectories closely align with the actual execution, demonstrating robust state modeling capabilities. For complex environments involving multiple room transitions (Figure 1(b)), NavMorph maintains trajectory consistency with only minor deviations at critical decision points. Notably, in challenging multi-level scenarios (Figure 1(c)), where descending stairs introduces visual discontinuities and occlusions, the model exhibits resilient prediction performance.

Notably, the imagined paths predicted by the foresight action planner (indicated by green dashed trajectories) occasionally traverse through physical obstacles such as walls or furniture. This limitation stems from the world model's imaginative rollouts lacking immediate environmental feedback, particularly navigation rewards that would typically penalize obstacle intersections. Nevertheless, the foresight action planner proves effective as an approximation mechanism, generating target-oriented trajectories that the world-aware navigator can dynamically adjust during execution to satisfy environmental constraints. These visualization results validate our world model's proficiency in capturing environmental dynamics and spatial-temporal relationships, facilitating effective predictive planning in navigation tasks.

References

[1] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Topo-metric map pre-training for language-guided navigation. *arXiv preprint arXiv:2212.04385*, 2022. 4

[2] Dong An, Zun Wang, Yangguang Li, Yi Wang, Yicong Hong, Yan Huang, Liang Wang, and Jing Shao. 1st place solutions for rxr-habitat vision-and-language navigation competition. *arXiv preprint arXiv:2206.11610*, 2022. 5

[3] Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE TPAMI*, 2024. 3, 4, 5

[4] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 2

[5] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683, 2018. 2, 3, 4, 5

[6] Jinyu Chen, Chen Gao, Erli Meng, Qiong Zhang, and Si Liu. Reinforced structured state-evolution for vision-language navigation. In *CVPR*, pages 15429–15438, 2022. 4

[7] Jiaqi Chen, Bingqian Lin, Xinmin Liu, Xiaodan Liang, and Kwan-Yee K Wong. Affordances-oriented planning using foundation models for continuous vision-language navigation. *arXiv preprint arXiv:2407.05890*, 2024. 5

[8] Kevin Chen, Junshen Chen, Jo Chuang, Marynel V'azquez, and Silvio Savarese. Topological planning with transformers for vision-and-language navigation. In *CVPR*, pages 11271–11281, 2020. 4

[9] Peihao Chen, Dongyu Ji, Kun-Li Channing Lin, Runhao Zeng, Thomas H. Li, Mingkui Tan, and Chuang Gan. Weakly-supervised multi-granularity map learning for vision-and-language navigation. In *NeurIPS*, pages 38149–38161, 2022. 4, 5

[10] Peihao Chen, Xinyu Sun, Hongyan Zhi, Runhao Zeng, Thomas H Li, Gaowen Liu, Mingkui Tan, and Chuang Gan. A^2 nav: Action-aware zero-shot robot navigation by exploiting vision-and-language ability of foundation models. *arXiv preprint arXiv:2308.07997*, 2023. 5

[11] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *CVPR*, pages 16537–16547, 2022. 3, 4

[12] Guangzhao Dai, Jian Zhao, Yuantao Chen, Yusen Qin, Hao Zhao, Guosen Xie, Yazhou Yao, Xiangbo Shu, and Xuelong Li. Unitedvln: Generalizable gaussian splatting for continuous vision-language navigation. *ArXiv*, abs/2411.16053, 2024. 5

[13] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[14] Junyu Gao, Xuan Yao, and Changsheng Xu. Fast-slow test-time adaptation for online vision-and-language navigation. In *ICML*, pages 14902–14919, 2024. 4, 5, 6

[15] Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *CVPR*, pages 15439–15449, 2022. 3, 4, 5

[16] Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. In *NeurIPS*, pages 27253–27266, 2022. 6

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[18] Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *CVPR*, pages 15418–15428, 2022. 3, 4, 5

[19] Yicong Hong, Yang Zhou, Ruiyi Zhang, Franck Dernoncourt, Trung Bui, Stephen Gould, and Hao Tan. Learning navigational visual representations with semantic map supervision. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3032–3044, 2023. 4

[20] Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446*, 2019. 2

[21] Muhammad Zubair Irshad, Niluthpol Chowdhury Mithun, Zachary Seymour, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. Semantically-aware spatio-temporal reasoning agent for vision-and-language navigation in continuous environments. In *ICPR*, pages 4065–4071, 2021. 4

[22] Johan Ludwig William Valdemar Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*, 30(1):175–193, 1906. 1

[23] Jacob Krantz and Stefan Lee. Sim-2-sim transfer for vision-and-language navigation in continuous environments. In *ECCV*, pages 588–603, 2022. 4

[24] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*, pages 104–120, 2020. 3

[25] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *EMNLP*, pages 4392–4412, 2020. 3

[26] Jiaming Liu, Senqiao Yang, Peidong Jia, Ming Lu, Yandong Guo, Wei Xue, and Shanghang Zhang. Vida: Homeostatic visual domain adapter for continual test time adaptation. In *ICLR*, 2024. 6

[27] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019. 3

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2017. 6

[29] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *ICLR*, 2023. 6

[30] Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel X. Chang. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. In *EMNLP*, pages 4018–4028, 2021. 4, 5

[31] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 3

[32] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 6

[33] Hanqing Wang, Wei Liang, Luc Van Gool, and Wenguan Wang. Dreamwalker: Mental planning for continuous vision-language navigation. In *ICCV*, pages 10873–10883, 2023. 4

[34] Ting Wang, Zongkai Wu, Feiyu Yao, and Donglin Wang. Graph-based environment representation for vision-and-language navigation in continuous environments. In *ICASSP*, pages 8331–8335, 2024. 4

[35] Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12009–12020, 2023. 4

[36] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *ICCV*, pages 15625–15636, 2023. 3, 4

[37] Zihan Wang, Xiangyang Li, Jiahao Yang, and Shuqiang Jiang. Sim-to-real transfer via 3d feature fields for vision-and-language navigation. In *CoRL*, 2024. 3, 4, 5, 6

[38] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, Junjie Hu, Ming Jiang, and Shuqiang Jiang. Lookahead exploration with neural radiance representation for continuous vision-language navigation. In *CVPR*, pages 13753–13762, 2024. 4, 5

[39] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and Wang He. Navid: Video-based vlm plans the next step for vision-and-language navigation. In *RSS*, 2024. 4, 5