DEFT: Demystifying VLN Failures via a Unified Dual-View Explainability Framework for LLM-based Agents

Yawen Wang[1,2,3,4], **Yihan Dai**[1,2,3,4], **Jianming Chen**[1,2,3,4], **Junjie Wang**[1,2,3,4]*, **Qing Wang**[1,2,3,4]*

[1]Institute of Software, Chinese Academy of Sciences, Beijing, China
[2]State Key Laboratory of Complex System Modeling and Simulation Technology, Beijing, China
[3]Science & Technology on Integrated Information System Laboratory, Beijing, China
[4]University of Chinese Academy of Sciences, Beijing, China
{yawen2018, jianming2023, junjie, wq}@iscas.ac.cn, daiyihan25@mails.ucas.ac.cn

Abstract

Large Language Models (LLMs) have emerged as central planners in Vision-and-Language Navigation (VLN), yet their complexity increasingly obscures their internal decision-making. Existing interpretability methods typically isolate temporal criticality from feature salience, creating an alignment gap and failing to account for the behavioral instability of black-box agents. To address this, we propose DEFT, a unified dual-view framework that demystifies agent behavior by jointly analyzing *when* a decision is pivotal and *what* visual evidence grounds it. Featuring a dual-head architecture with a shared latent representation, DEFT employs a *Mask Head* for counterfactual-based criticality detection and an *Action Head* that leverages an ensemble of surrogates to recover robust visual cues. Extensive experiments on MatterPort3D across three LLM-based agents demonstrate that DEFT outperforms baselines in both temporal and feature fidelity. User studies further validate its utility, showing 78% alignment with human intuition.

1 Introduction

Integrating Large Language Models (LLMs) (OpenAI, 2023) into embodied agents has become a dominant paradigm for tackling complex physical tasks. In the field of Vision-and-Language Navigation (VLN) (Wu et al., 2024), LLMs are increasingly deployed as central planners, leveraging their extensive world knowledge and reasoning capabilities to guide agents through unseen environments based on natural language instructions.

To tackle complex VLN tasks, recent frameworks have evolved beyond simple prompting, integrating auxiliary mechanisms such as history modeling (Zhou et al., 2024b), topological memories (Chen et al., 2024a), and multi-agent deliberation (Long et al., 2024). While these designs enhance performance, they inevitably increase system opacity, making it difficult to determine the rationale behind specific actions, whether they stem from precise visual grounding or sound logical reasoning. Purely observing the final outcome offers little insight into the agent's internal decision process, limiting the transparency needed for both understanding and optimization.

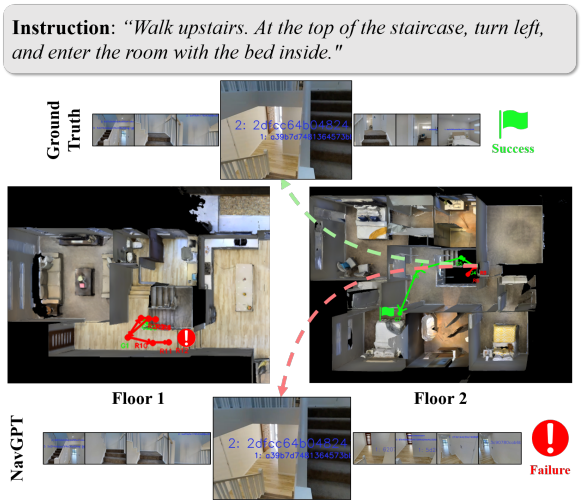


Figure 1: NavGPT failure case: A critical directional misinterpretation of the staircase occurs at step 5.

Fig.1 illustrates this diagnostic necessity. At a pivotal timestep, NavGPT(Zhou et al., 2024b) correctly detects the staircase but misinterprets its directional semantics, choosing to go downstairs rather than upstairs. This single error propagates into an irreversible failure, highlighting that identifying both critical timesteps (*when*) and grounding visual evidence (*what*) is fundamental to understanding and optimizing these agents.

However, interpreting these agents is non-trivial, as adapting general techniques from reinforcement learning or computer vision to the multimodal, sequential nature of VLN faces three primary limitations: (1) *Alignment Gap*: Existing approaches typically treat step-level (Cheng et al., 2023) and

*Corresponding authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 29541–29560

feature-level importance (Sundararajan et al., 2017) in isolation. This separation creates a gap where visual evidence explaining *what* guided an action may not correspond to cues determining *when* a decision was critical, leading to disjoint or even contradictory explanations. (2) *Inherent Unfaithfulness*: While Chain-of-Thought (CoT) offers a direct channel, it is often an unfaithful post-hoc rationalization rather than a reflection of true model internals (Turpin et al., 2023). Furthermore, textual CoT lacks the spatial granularity required to pinpoint specific visual regions responsible for grounding errors. (3) *Behavioral Instability*: LLM-based planners frequently exhibit instable behaviors. Standard interpretability methods are unreliable in this context, as they risk attributing significance to features that trigger random hallucinations rather than consistent reasoning patterns.

In this work, we propose **DEFT**, a unified **D**ual-view **E**xplainability framework designed to demystify black-box VLN agents through the joint analysis of salient **F**eatures and critical **T**imesteps. Unlike disjoint approaches that apply separate analyzers, DEFT is constructed as a unified interpreter with a dual-head architecture sharing a common latent representation. Specifically, the *Mask Head* functions as a criticality detector; it utilizes a counterfactual masking mechanism to simulate alternative actions and measure reward degradation, thereby pinpointing timesteps essential for navigation success. Complementarily, the *Action Head* serves as a differentiable proxy. By leveraging an ensemble of gradient-accessible surrogate models, this module recovers fine-grained visual features while mitigating the behavioral instability of LLMs through robust attribution aggregation.

Extensive experiments in the MatterPort3D simulator across three LLM-based agents (NavGPT, MapGPT, NavGPT-2) validate the effectiveness of DEFT. Quantitatively, it consistently outperforms baselines in timestep criticality, achieving RRD scores of 1.28–4.13 and Fidelity of 0.15–0.77. In feature attribution, DEFT surpasses both white- and black-box methods in Insertion and Deletion metrics while maintaining consistently positive μ_F scores. Qualitatively, user studies confirm its practical utility: DEFT aligns better with human intuition (78% preference) and more than doubles the success rate in diagnosing failure causes (64% vs. 30%) compared to strong composite baselines. The contributions of this work are as follows:

- **DEFT**, a unified dual-view framework for systematically explaining black-box LLM-based VLN agents by jointly analyzing temporal criticality and feature salience.
- An ensemble-surrogate strategy that aggregates gradient-based explanations to recover robust, pixel-level visual evidence.
- Extensive experiments on MatterPort3D across three LLM agents, validating the framework's superiority over strong baselines both quantitatively and qualitatively.
- Publicly available codebase[1] to ensure reproducibility and support further studies.

2 Related Work

2.1 LLM-Based VLN

VLN requires agents to navigate unseen environments guided by natural language. While traditional imitation (IL) and reinforcement learning (RL) established strong foundations (Chen et al., 2022), the field has pivoted toward leveraging LLMs for complex reasoning. Early zero-shot planners like NavGPT (Zhou et al., 2024b) and NaVid (Zhang et al., 2024) have been further enhanced by structured memories (e.g., MapGPT (Chen et al., 2024a)) and multi-agent deliberation (e.g., DiscussNav (Long et al., 2024)). Despite their success, these agents remain prone to hallucinations and compounding errors. Furthermore, the reliance on proprietary models (e.g., GPT-4) renders their internal logic opaque. Unlike previous works that prioritize architectural design, we focus on demystifying these black-box agents by identifying critical timesteps and salient visual features without internal parameter access.

2.2 Model Interpretability in Decision Making

Interpretability in decision-making generally branches into two dimensions: temporal criticality and feature attribution. To identify pivotal timesteps, RL-based methods such as EDGE (Guo et al., 2021) and AIRS (Yu et al., 2023) model trajectory dependencies, while others like StateMask (Cheng et al., 2023), Rice (Cheng et al., 2024) and UTILITY (Liu and Zhu, 2025) assess stepwise significance via auxiliary agents or inverse RL. To ground decisions in visual inputs, white-box techniques typically leverage gradients (Zahavy et al., 2016; Selvaraju et al., 2017) or attention weights (Mott et al., 2019), whereas black-box approaches rely on perturbation-based mask-

[1] https://github.com/yihandai/DEFT

ing (Greydanus et al., 2018). However, naively combining these disparate techniques introduces alignment gaps and fails to account for the inherent instability of LLM agents. In contrast, DEFT bridges these perspectives through a unified dual-head architecture, employing an ensemble surrogate strategy to ensure robust and consistent explanations under black-box constraints.

3 Problem Definition

Vision-and-Language Navigation. VLN is modeled as a sequential decision-making, typically a Markov Decision Process (MDP). At each timestep t, an agent receives a natural language instruction $\mathcal{I}$ and a panoramic visual observation $\mathcal{O}_t$ to select an action $a_t \in \mathcal{A}_t$ via the navigation policy $\pi(\mathcal{A}_t|\mathcal{I}, \mathcal{O}_t)$. In discrete environments, navigation occurs on a graph $G = (V, E)$, where nodes $v \in V$ represent navigable viewpoints. An action a_t involves transitioning to a node within a candidate set $\mathcal{A}_t \subseteq V$, defined by local graph connectivity. The episode terminates upon reaching the goal or exceeding a maximum horizon T. See Appx. A.1 for extended background.

Interpretability Objectives. Under a black-box setting, where the agent's internal parameters and gradients are inaccessible, we define interpretability for a navigation trajectory $\tau = \{(\mathcal{O}_t, \mathcal{A}_t)\}_{t=1}^{T}$ along two dimensions. *1) Timestep-level Criticality:* The goal is to identify a subset of pivotal timesteps $\mathcal{T}^* \subseteq \{1, \ldots, T\}$. A timestep $t \in \mathcal{T}^*$ is defined as critical if a counterfactual perturbation on action $\mathcal{A}_t$ leads to a significant cumulative reward degradation or even a failure. *2) Feature-level Salience:* For each step t, we aim to extract a salient pixel subset $P_t \subseteq \{1, \ldots, H\} \times \{1, \ldots, W\}$ from the $H \times W$ panoramic observation $\mathcal{O}_t$. This subset must represent the dominant visual evidence that grounds the agent's selected action $\mathcal{A}_t$.

4 Proposed Method: DEFT

As illustrated in Fig. 2, DEFT integrates temporal and visual interpretability for black-box LLM agents via a dual-head architecture. Specifically, a *Mask Head* identifies critical steps through counterfactual analysis, while an *Action Head* leverages an ensemble of differentiable surrogates to recover robust visual evidence.

4.1 Architecture of DEFT

DEFT adopts a unified architecture where a shared cross-modal backbone encodes instruction-conditioned visual features for two specialized heads: an *Action Head* for policy approximation and a *Mask Head* for step-level criticality estimation. This joint design enforces representational consistency between decision modeling and interpretability within a single framework.

Shared Backbone. VLN↻BERT (Hong et al., 2021) is employed as DEFT's cross-modal backbone. At each timestep t, it encodes the instruction $\mathcal{I}$, panoramic observation $\mathcal{O}_t$, and previous state s_{t-1} via cross-modal self-attention to derive a joint instruction-conditioned representation:

$$h_t = \text{VLN} \circlearrowright \text{BERT}(s_{t-1}, \mathcal{I}, \mathcal{O}_t) \quad (1)$$

The resulting h_t provides a shared latent representation for the subsequent dual-head analysis.

Mask Head. The *Mask Head* estimates temporal criticality by mapping the shared representation h_t through a feed-forward network (FFN) to yield a binary criticality distribution:

$$P_{\text{crit}} = \text{Softmax}(\text{FFN}_{\text{mask}}(h_t)) \quad (2)$$

where $P_{\text{crit}} \in \mathbb{R}^2$ denotes the probabilities of the current step being critical or non-critical.

Action Head. To enable feature attribution, the *Action Head* serves as a surrogate proxy that maps h_t to a probability distribution over next navigable viewpoints:

$$P_{\text{act}} = \text{Softmax}(\text{FFN}_{\text{act}}(h_t)) \quad (3)$$

where P_{act} denotes the action probabilities over the candidate navigable viewpoints.

4.2 Feature-level Interpretability

Our feature-level interpretation follows a two-stage process: first approximating the target agent with multiple gradient-accessible surrogates, then performing ensemble attribution via Integrated Gradients (IG) (Sundararajan et al., 2017). Aggregating explanations across surrogates ensures stable and robust visual grounding, mitigating the behavioral instability inherent in LLM agents.

Surrogate Models. To approximate the target agent's policy, we construct an ensemble of K differentiable surrogates. Each surrogate is a non-LLM VLN model trained on a distinct trajectory set, generated by executing the target agent multiple times. This multi-run data collection allows

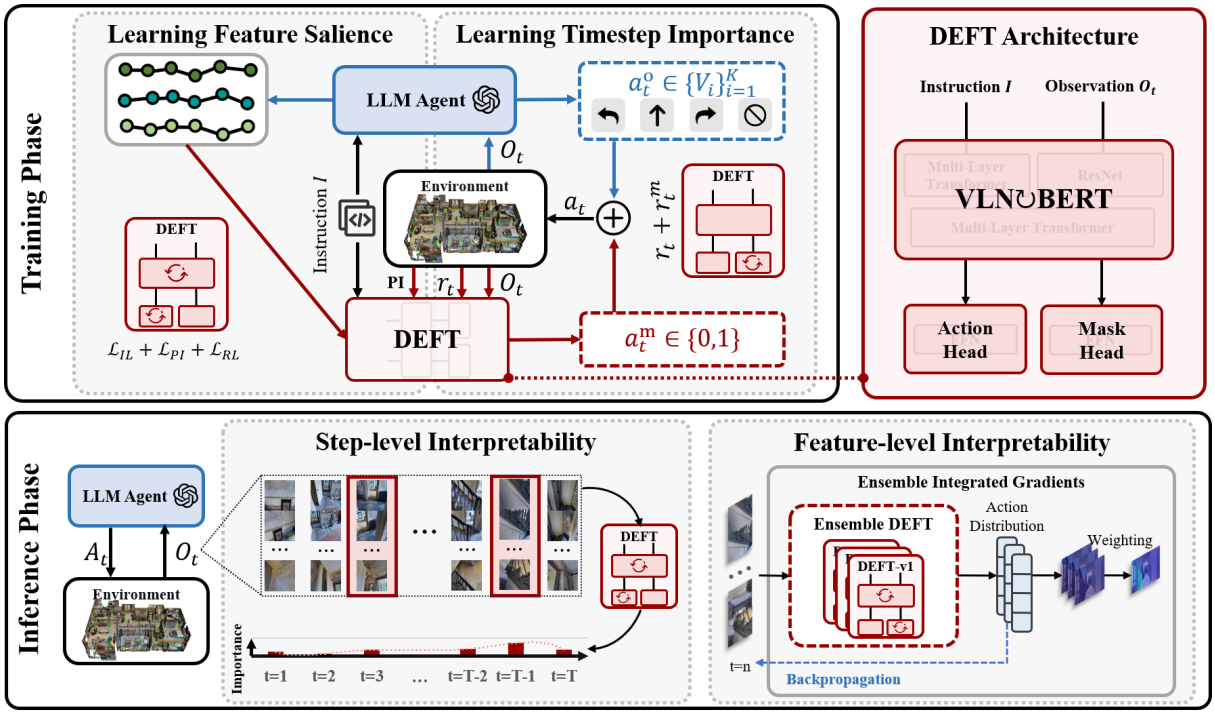


Figure 2: The overview of DEFT.

the ensemble to capture the target agent's behavioral variations. By diversifying the surrogates, we simultaneously characterize the agent's instability and provide the multiple distinct proxies required for robust ensemble attribution.

Training Surrogate Model. We train the shared backbone together with the *Action Head* as a surrogate model using a hybrid IL and RL objective. At each timestep t, given the state s_t (comprising the navigation instruction and the current observation), the IL objective encourages the surrogate policy π^θ to imitate the target VLN agent π:

$$\mathcal{L}_{\text{IL}} = -\log p_\theta(a_t^\pi \mid s_t) \tag{4}$$

where a_t^π is the action taken by the VLN agent.

To mitigate the convergence difficulties and distribution shifts arising from the target agent's inherent behavioral instability, we further incorporate pseudo-interactive supervision (Chen et al., 2022). A heuristic demonstrator π^* selects corrective actions based on shortest-path guidance in the environment graph, yielding an auxiliary loss:

$$\mathcal{L}_{\text{PI}} = \sum_{t=1}^{T} -\log p_\theta(a_t^{\pi^*} \mid s_t) \tag{5}$$

where $a_t^{\pi^*}$ is the pseudo-interactive action.

Then we employ RL to provide trajectory-level feedback via an A2C (Mnih et al., 2016) objective:

$$\mathcal{L}_{\text{RL}} = -A_t \log p_\theta(a_t^{\pi^\theta} \mid s_t) \tag{6}$$

where $a_t^{\pi^\theta}$ is sampled from π^θ, A_t is the advantage estimate. Since the navigation environment does not provide a built-in reward function $\mathcal{R}$, we adopt the composite design from VLN↻BERT (Hong et al., 2021), comprising a progress reward, a path-fidelity reward, and a negative stop reward. Specifically, the path-fidelity term encourages the surrogate to mirror the target agent's trajectory while maintaining goal-oriented optimality. Detailed analyses of each reward component are provided in Appx. A.7.

Finally, the *Action Head* is trained by jointly minimizing the combined loss:

$$\mathcal{L} = \mathcal{L}_{\text{IL}} + \mathcal{L}_{\text{PI}} + \mathcal{L}_{\text{RL}} \tag{7}$$

This hybrid training enables the surrogate to closely match the target agent's local decisions while remaining a functional and robust navigator.

Ensemble Integrated Gradients. Given the trained surrogate, we employ IG (Sundararajan et al., 2017) to attribute the *Action Head*'s navigation policy to individual input pixels at each timestep t. For a specific observation $\mathcal{O}_t$ and its baseline $\mathcal{O}_t'$, the attribution along the i-th dimen-

sion is formulated as:

$$IG_i(\mathcal{O}_t) = (\mathcal{O}_{i,t} - \mathcal{O}'_{i,t}) \int_{\alpha=0}^{1} \frac{\partial \pi^\theta(\mathcal{O}'_t + \alpha(\mathcal{O}_t - \mathcal{O}'_t), \mathcal{I})}{\partial \mathcal{O}_{i,t}} \, \mathrm{d}\alpha \tag{8}$$

where π^θ denotes the surrogate policy, $\mathcal{I}$ is the navigation instruction, and $\frac{\partial \pi^\theta(\cdot)}{\partial \mathcal{O}_{i,t}}$ represents the gradient with respect to the i-th input dimension. The baseline $\mathcal{O}'_t$ is typically chosen as an all-black or all-white image with the same resolution as $\mathcal{O}_t$. The interpolation coefficient $\alpha \in [0, 1]$ smoothly transitions from the baseline input ($\alpha = 0$) to the original observation ($\alpha = 1$).

To improve robustness against surrogate variance, we adopt an ensemble-based IG (EIG) strategy. Rather than relying on a single surrogate, we compute IG explanations across an ensemble of K surrogate models, $\{\pi^{\theta_k}\}_{k=1}^{K}$, and aggregate their pixel-wise attribution maps $\mathrm{IG}^{(k)}(\mathcal{O}_t)$ for a given observation $\mathcal{O}_t$. A general weighted aggregation can be expressed as:

$$\mathrm{EIG}(\mathcal{O}_t) = \sum_{k=1}^{K} w_k \, \mathrm{IG}^{(k)}(\mathcal{O}_t), \quad \text{s.t.} \sum_{k=1}^{K} w_k = 1 \tag{9}$$

where w_k reflects the reliability of surrogate π^{θ_k}, e.g., measured via action agreement or rollout similarity with the target VLN agent. In practice, uniform weighting (i.e., averaging) is sufficiently effective and is adopted by default. This ensemble mechanism reduces sensitivity to individual surrogate errors, yielding more stable and robust feature-level explanations.

4.3 Timestep-Level Interpretability

To identify critical steps, DEFT operates alongside the target VLN agent, observing the same environment state $\mathcal{O}_t$ and instruction $\mathcal{I}$, and predicting the likelihood that the current step is critical. To assess the influence of a timestep, we mask the agent's action by sampling a random alternative from the action space, yielding a counterfactual trajectory whose drop in cumulative reward reflects the step's importance. By learning to associate mask-induced reward degradation with specific timesteps, DEFT effectively pinpoints decision points that strongly determine navigation outcomes (Cheng et al., 2023; Chen et al., 2025).

Problem Modeling. We formulate the above process as a MDP, defined as $G = \langle \mathcal{O}, \mathcal{A}^m, \mathcal{P}, \mathcal{R}, \gamma \rangle$. The observation space $\mathcal{O}$, state-transition dynamics $\mathcal{P}$ follow the same definitions as in the original VLN environment. The reward function $\mathcal{R}$ adopts the composite reward design from VLN⟳BERT (Hong et al., 2021). DEFT's *Mask Head* receives an observation from the environment and outputs a masking action according to the policy $\pi^\theta : o \mapsto a^m$, parameterized by θ. The masking action is selected from the binary action space $\mathcal{A}^m = \{0, 1\}$, where $a^m = 0$ denotes no masking and $a^m = 1$ denotes applying a mask. The discount factor γ is used to weight future rewards.

According to the above formulation, the VLN agent produces an action a^o according to its fixed policy π, while DEFT trains the masking policy π^θ to determine whether the action should be masked at each step. The final action a executed in the environment is given by:

$$a = a^m \oplus a^o = \begin{cases} a^o, & \text{if } a^m = 0, \\ \text{RndAct}, & \text{if } a^m = 1. \end{cases} \tag{10}$$

where operator $\oplus$ denotes a conditional selection: the original action is retained when $a^m = 0$, and replaced by a random alternative when $a^m = 1$.

The objective of DEFT's *Mask Head* π^θ is to learn a masking policy that perturbs the VLN agent while minimizing the difference in expected rewards (Chen et al., 2025):

$$obj(\pi^\theta) = \arg\min_{\theta} \left| J(\pi) - J(\pi^\theta) \right| \tag{11}$$

where $J(\pi)$ denotes the expected reward obtained by the VLN agent under its fixed policy π, and $J(\pi^\theta)$ denotes the expected reward when the agent's actions are perturbed by DEFT's masking policy π^θ. To encourage exploration by masking more target actions during training, we introduce a sparse masking reward defined as $r_t^m = \beta \cdot a_t^m$. The coefficient β is a hyperparameter controlling the strength of this constraint. Combining this with the original task reward r_t (i.e., the composite reward described above), The final expected discounted reward is therefore given by:

$$J(\pi^\theta) = \mathbb{E}_{s_t, a_t} \left[\sum_t \gamma_t \left(r_t + r_t^m \right) \right] \tag{12}$$

Training Mask Head. Since the surrogate (i.e., the shared backbone together with the *Action Head*) is trained, we freeze its backbone and optimize only the *Mask Head* via RL. This strategy effectively

reduces interaction overhead with LLM agent while boosting overall performance (See Sec. 6.1).

To optimize the reward-difference objective in Eq. 11, we adopt the PPO-style constrained optimization method of StateMask (Cheng et al., 2023). Let π denote the fixed VLN agent policy, and let $\tilde{\pi}_\theta$ denote the masked policy induced by the *Mask Head* with parameters θ. The *Mask Head* is trained to minimize the behavioral and reward deviation caused by masking, formulated as:

$$\begin{aligned} &\max_{\theta} \ \mathcal{L}_{\text{PPO}}(\tilde{\pi}_\theta) \ - \ w\,\mathcal{L}_{\text{dist}}(\tilde{\pi}_\theta, \pi) \\ &\text{s.t.} \quad \text{KL}(\tilde{\pi}_{\theta_{\text{old}}} \,\|\, \tilde{\pi}_\theta) \le \delta \end{aligned} \tag{13}$$

where $\mathcal{L}_{\text{PPO}}$ is the standard PPO objective computed on the masked policy, and $\mathcal{L}_{\text{dist}}$ penalizes deviations between the masked policy and the original VLN agent. The KL constraint stabilizes optimization by limiting the update magnitude of the masked policy between iterations.

5 Experiment Setup

5.1 Environment and Dataset

We evaluate DEFT on the Room-to-Room (R2R) dataset (Anderson et al., 2018) hosted on the MatterPort3D simulator. The simulator encompasses 90 diverse indoor environments modeled as undirected connectivity graphs, where the agent perceives egocentric RGB panoramas at each discrete node (Fig. 4). The dataset contains approximately 22K instructions (avg. 29 words) split into *Train* (14,025), *Val Seen* (1,020), *Val Unseen* (2,349), and *Test* (4,173). In this work, we train our interpretability modules using a randomly sampled 10% subset of the *Train* split (See training convergence and efficiency in Appx. A.5). For evaluation, following NavGPT (Zhou et al., 2024b) and MapGPT (Chen et al., 2024a), we test on the standard subset of 216 trajectories from *Val Unseen* split. A detailed computational breakdown of DEFT's training and inference phases is provided in Appx. A.6.

5.2 Target Models

We validate DEFT on three agents covering distinct modalities and training strategies:
(1) **NavGPT** (Zhou et al., 2024b) is a text-mediated zero-shot agent driven by GPT-4. It utilizes visual foundation models to translate observations into text, performing navigation via explicit CoT reasoning without direct visual perception.
(2) **MapGPT** (Chen et al., 2024a) is a multimodal agent based on GPT-4V. It processes visual inputs directly and maintains a "linguistic topological map" in its prompt. This mechanism acts as a global memory to support adaptive global planning and systematic exploration.
(3) **NavGPT-2** (Zhou et al., 2024a) represents the open-source, fine-tuned paradigm. Unlike prompt-based methods, it employs visual instruction tuning to align visual-language model latent representations with navigational contexts, balancing general language skills with task-specific reasoning.

5.3 Baselines

We evaluate DEFT against state-of-the-art interpretability methods. Under our black-box setting, white-box baselines are applied to the gradient-accessible surrogate models as proxies (Sec. 4.2). Detailed descriptions are provided in Appx. A.2.

5.3.1 Timestep Criticality

(1) **StateMask (SM)** (Cheng et al., 2023), which measures reward degradation via a masking policy.
(2) **Value-Based (VB)** (Huang et al., 2018), defining importance by the divergence of surrogate action-values.
(3) **Gradient-Based (GB)** (Srinivas and Fleuret, 2021), using logit gradients of executed actions.

5.3.2 Feature Attribution

(1) **IG** (Sundararajan et al., 2017) and (2) **Guided-IG** (Kapishnikov et al., 2021), using linear and adaptive path integration respectively.
(3) **SMDL** (Chen et al., 2024b), formulated as submodular subset selection for sparse attribution.
(4) **HSIC** (Novello et al., 2022), a statistical black-box baseline that estimates non-linear feature dependencies.

5.4 Evaluation Metrics

We assess DEFT on two levels, with detailed definitions and equations provided in Appx. A.3. All results are averaged over three independent runs.

5.4.1 Timestep-Level Metrics

(1) **Relative Reward Difference (RRD)** (Yu et al., 2023), which normalizes the reward degradation caused by masking critical steps against random perturbations.
(2) **Fidelity** (Guo et al., 2021), measuring the alignment between predicted importance and actual performance impact, with a sequence penalty to discourage redundant explanations.

Table 1: Performance comparison of RRD and Fidelity for timestep criticality across models and methods.

Model	Metric	SM	VB	GB	DEFT
NavGPT	RRD↑	3.65	3.97	3.79	**4.13**
	Fidelity↑	-0.45	0.13	-0.46	**0.23**
MapGPT	RRD↑	1.01	1.16	1.06	**1.28**
	Fidelity↑	-0.24	0.04	-0.96	**0.15**
NavGPT-2	RRD↑	2.48	**3.26**	2.80	3.15
	Fidelity↑	-0.40	0.66	-0.41	**0.77**

5.4.2 Feature-Level Metrics

In black-box settings, we evaluate feature attribution via *Action Consistency* (i.e., whether the agent retains its original decision under perturbation): (1) **Insertion (Ins)** and (2) **Deletion (Del)** assess the average action consistency as pixels are progressively added or removed based on their importance. (3) μ**Fidelity** (μ_F) quantifies the correlation between accumulated attribution scores and the model's behavioral deviation under random subset perturbation.

5.4.3 Implementation Details

We implement DEFT using Python 3.10. For the target LLM VLN agents and all baseline methods, we adhere to their official implementations and default hyperparameter configurations to ensure a fair comparison. We specifically set the DEFT's ensemble size to $K = 4$ (See Appx. A.8 for its impact). All experiments were conducted on a server equipped with an Intel i7-10700 CPU, an NVIDIA TITAN RTX GPU and 32GB RAM.

6 Results

6.1 Timestep-Level Fidelity Evaluation

We first evaluate the effectiveness of DEFT in identifying pivotal decision steps, with results summarized in Table 1. Across all target agents, DEFT consistently demonstrates superior performance compared to both black-box and white-box baselines. Notably, baselines such as SM and GB frequently yield negative *Fidelity* scores, indicating a significant misalignment between their predicted importance and the agent's actual navigation impact. In contrast, DEFT achieves positive and substantially higher *Fidelity* across all settings. This confirms that our counterfactual masking mechanism more accurately pinpoints steps that are truly critical to navigation success by directly measuring reward degradation under intervention. Regarding the RRD metric, while the VB method achieves a marginally higher score on NavGPT-2 (3.26 vs. 3.15), due to its direct reliance on the surrogate's critic network for value estimation. Nevertheless, DEFT offers a superior overall diagnostic trade-off, maintaining a highly competitive RRD while delivering higher *Fidelity* (0.77 vs. 0.66). Furthermore, DEFT consistently outperforms SM, attributed to our shared representation and two-stage optimization strategy. Freezing the shared backbone allows the *Mask Head* to exploit robust, task-aligned visual features from surrogate training to effectively boost performance.

6.2 Feature-Level Fidelity Evaluation

We evaluate the quality of visual explanations generated by DEFT against standard white-box (e.g., IG, SMDL) and black-box (HSIC) baselines, with results summarized in Table 2 (See Ins/Del ratio impact in Appx. 7). DEFT consistently outperforms these methods across metrics and agents, validating the robustness of our ensemble-based strategy.

First, regarding faithfulness, DEFT achieves higher μ_F scores than all baselines. Standard gradient-based methods often yield negative values, indicating misalignment due to agent instability, while DEFT consistently maintains positive scores. This confirms that our ensemble mechanism effectively recovers visual evidence truly faithful to the black-box agent. Second, DEFT demonstrates precise saliency localization. It achieves the highest Ins scores across all three agents, indicating that the highlighted pixels contribute most to agent confidence. Similarly, for Del, DEFT remains highly competitive, achieving the best scores on MapGPT and tying for best on NavGPT-2 (0.75), further validating its visual grounding precision. Third, comparison on the open-source NavGPT-2 reveals that DEFT surpasses the statistical black-box baseline, HSIC, in both Ins (0.78 vs. 0.74) and μ_F (1.27 vs. -5.50). Crucially, HSIC is computationally prohibitive due to thousands of inference queries, while DEFT achieves superior fidelity efficiently via its gradient-accessible surrogates, offering a practical solution for analyzing LLM VLN agents.

6.3 Qualitative Evaluation

Quantitative metrics alone cannot capture holistic interpretability, where the strength of our unified framework lies. For qualitative evaluation, we conduct a user study and a case analysis to demonstrate the explanatory power of DEFT.

Table 2: Comparison of feature attribution methods using Ins, Del (at 25% perturbation), and μ_F ($\times 10^{-3}$). Due to high computational overhead, HSIC is evaluated only on the open-source NavGPT-2. '-' denotes omitted entries.

Model	Metric	IG	Guided-IG	SMDL	HSIC	DEFT
NavGPT	Ins↑	0.74	0.71	0.69	-	**0.75**
	Del↓	0.76	0.81	**0.72**	-	0.76
	μ_F↑	-101.45	-29.31	-10.29	-	**12.52**
MapGPT	Ins↑	0.67	0.69	0.69	-	**0.70**
	Del↓	0.72	0.73	0.75	-	**0.69**
	μ_F↑	23.52	38.13	30.00	-	**69.20**
NavGPT-2	Ins↑	0.76	0.77	0.76	0.74	**0.78**
	Del↓	0.80	0.80	0.79	**0.75**	**0.75**
	μ_F↑	-5.94	-7.64	0.40	-5.50	**1.27**

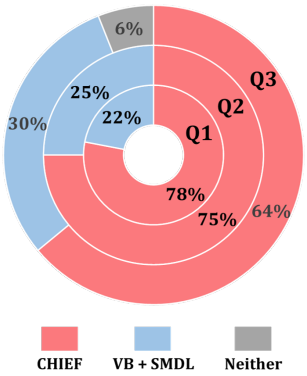


Figure 3: The user study results.

User Study. To validate the interpretability of DEFT from a human perspective, we conducted a user study with 20 participants. We benchmarked DEFT against a strong composite baseline formed by combining top-performing disjoint methods: VB (for timesteps) and SMDL (for features). Detailed experimental protocols and participant demographics are provided in Appx. A.4.

Participants evaluated the explanations across three dimensions: Q1 *Intuitive Alignment*, Q2 *Reasoning Understanding*, and Q3 *Failure Diagnosis*. As shown in Fig. 3, DEFT consistently outperforms the baseline: DEFT was favored in 78% of trials for intuitive alignment and 75% for reasoning understanding. For failure diagnosis, participants were asked to identify the root causes of navigation failures. Users equipped with DEFT successfully pinpointed the exact error cause in 64% of cases, more than doubling the success rate of the baseline (30%). Results are statistically significant: DEFT achieves $78\% \pm 18\%$ for intuitive alignment and $64\% \pm 21\%$ for failure diagnosis. Notably, even the lower bounds exceed baseline performance (22% and 30%, respectively). The advantage confirms by jointly modeling temporal and visual explanations, DEFT avoids unaligned explanations and reduces the cognitive load required to diagnose complex agent failures.

To intuitively validate the effectiveness of DEFT, we visualize a navigation episode in Fig. 5, and analyze the resulting attention maps to illustrate how DEFT aligns visual grounding with temporal criticality. At routine steps ($T = 0, 2$), DEFT reveals that the agent's focus aligns perfectly with instruction semantics, highlighting the geometry of the "hallway" and the "railing" respectively. This confirms that the agent effectively captures the semantic correspondence between the textual instruction and the visual scene. The diagnostic power is most evident at the critical decision point $T = 4$, when the agent faces visual ambiguity with "plants" appearing in multiple directions (one on the path, one in a side room, as seen in the top-down view). Here, DEFT elucidates the rationale behind the agent's correct decision: the generated saliency map shows that the agent effectively ignored the distractor plant, focusing exclusively on the correct "plant" near the stairs. This qualitative visualization confirms that DEFT accurately uncovers the specific visual cues driving the agent's successful navigation at pivotal moments.

6.4 Surrogate Fidelity Evaluation

Given the inherent behavioral instability of LLM-based agents, assessing surrogate fidelity is challenging. To address this, we propose a relative fidelity evaluation protocol. First, we measure the target agent's self-consistency by averaging trajectory similarities across K independent runs, establishing a relative baseline that reflects its intrinsic variance. We then measure the surrogate-target consistency by computing the average similarities between our K surrogate models and the target agent.

Table 3: Surrogate Fidelity: Comparison on nDTW and CLS metrics between target agents and our surrogate.

Model	nDTW↑	CLS↑
NavGPT	0.60	0.60
DEFT	0.55	0.53
MapGPT	0.58	0.61
DEFT	0.61	0.60
NavGPT-2	0.99	0.99
DEFT	0.80	0.78

Trajectory similarity is quantified using nDTW and CLS (Detailed in Appx. A.3).

As shown in Table 3, for high-variance agents (NavGPT, MapGPT), our surrogate-target alignment closely approaches the targets' self-consistency scores. Notably, DEFT even surpasses the self-consistency baseline on MapGPT, suggesting that the surrogate effectively captures the agent's central behavioral tendency. For the highly stable NavGPT-2 (self-consistency ≈ 0.99), while a gap exists due to the difficulty of approximating a near-deterministic policy, DEFT maintains a high absolute alignment score (0.80). These results show that the surrogates closely approximate the black-box agents and learn robust representations, enabling IG-based attribution to capture stable, task-aligned reasoning patterns even under divergent actions.

7 Conclusion

This paper presents DEFT, a unified dual-view framework that demystifies black-box LLM VLN agents by jointly analyzing temporal criticality and feature salience. By integrating counterfactual masking with an ensemble-based surrogate strategy, DEFT successfully bridges the gap between when decisions are pivotal and what evidence supports them. Quantitative evaluations and user studies demonstrate its superior performance in providing faithful, human-aligned explanations, offering a robust diagnostic tool to facilitate the development of more transparent and reliable navigation systems.

Limitations

Despite its effectiveness, DEFT has several limitations that remain to be addressed in future work.

Surrogate Fidelity. Although DEFT uses an ensemble of surrogate models to approximate black-box LLMs, a surrogate is inherently an approximation. While our fidelity experiments and superior performance demonstrate that these proxies capture the decision logic effectively, there may still be a gap in representing highly non-linear reasoning in extreme corner cases. We provide additional qualitative analyses of success and failure cases in Appx. A.10.

Dependency on Reward Signals. The criticality detection in the mask head relies on measuring reward degradation. In scenarios with sparse or poorly defined rewards, the precision of identifying critical timesteps may be affected (See detailed analyses of each reward component in Appx. A.7). Future work could explore self-supervised signals or goal-reaching metrics to reduce this threat.

Environment Scope. While this study focuses on the discrete MatterPort3D, we emphasize that DEFT is designed as a modular framework with interchangeable backbones, offering strong potential for generalization to continuous-control and other embodied tasks. In continuous VLN settings where actions are defined by orientation and displacement, the surrogate model only needs to be replaced with a continuous-controller backbone, such as ETPNav (An et al., 2025). Our proposed Ensemble-based Integrated Gradients method remains fully compatible, as it only requires gradient accessibility, a property shared by mainstream architectures including CNNs, RNNs, and Transformers. For tasks like robotic manipulation or exploration, the *Mask Head*'s counterfactual analysis can still work based on task-specific reward degradation, which can be well-defined by successful grasp or incremental map coverage. Fundamentally, as long as a differentiable surrogate can be optimized to approximate the target agent's policy, DEFT can work regardless of the underlying environment settings.

Acknowledgments

This work was supported by the National Natural Science Foundation of China Grant No. 62232016, Basic Research Program of ISCAS Grant No. ISCAS-JCZD-202405 and No. ISCAS-JCZD-202304, Major Program of ISCAS Grant No. ISCAS-ZD-202401 and No. ISCAS-ZD-202302, Innovation Team 2024 ISCAS (No. 2024-66).

References

Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. 2025. Etpnav: Evolving topological planning for vision-language

navigation in continuous environments. IEEE Trans. Pattern Anal. Mach. Intell., 47(7):5130–5145.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In CVPR, pages 3674–3683. Computer Vision Foundation / IEEE Computer Society.

Jianming Chen, Yawen Wang, Junjie Wang, Xiaofei Xie, Jun Hu, Qing Wang, and Fanjiang Xu. 2025. Understanding individual agent importance in multi-agent system via counterfactual reasoning. In AAAI, pages 15785–15794. AAAI Press.

Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee Kenneth Wong. 2024a. Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation. In ACL (1), pages 9796–9810. Association for Computational Linguistics.

Ruoyu Chen, Hua Zhang, Siyuan Liang, Jingzhi Li, and Xiaochun Cao. 2024b. Less is more: Fewer interpretable region via submodular subset selection. In ICLR. OpenReview.net.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. 2022. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In CVPR, pages 16516–16526. IEEE.

Zelei Cheng, Xian Wu, Jiahao Yu, Wenhai Sun, Wenbo Guo, and Xinyu Xing. 2023. Statemask: Explaining deep reinforcement learning through state mask. In NeurIPS.

Zelei Cheng, Xian Wu, Jiahao Yu, Sabrina Yang, Gang Wang, and Xinyu Xing. 2024. RICE: breaking through the training bottlenecks of reinforcement learning with explanation. In ICML. OpenReview.net.

Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. 2018. Visualizing and understanding atari agents. In ICML, volume 80 of Proceedings of Machine Learning Research, pages 1787–1796. PMLR.

Wenbo Guo, Xian Wu, Usmann Khan, and Xinyu Xing. 2021. EDGE: explaining deep reinforcement learning policies. In NeurIPS, pages 12222–12236.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez Opazo, and Stephen Gould. 2021. VLN BERT: A recurrent vision-and-language BERT for navigation. In CVPR, pages 1643–1653. Computer Vision Foundation / IEEE.

Sandy H. Huang, Kush Bhatia, Pieter Abbeel, and Anca D. Dragan. 2018. Establishing appropriate trust via critical states. In IROS, pages 3929–3936. IEEE.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. 2019. General evaluation for instruction conditioned navigation using dynamic time warping. In ViGIL@NeurIPS.

Vihan Jain, Gabriel Magalhães, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. 2019. Stay on the path: Instruction fidelity in vision-and-language navigation. In ACL (1), pages 1862–1872. Association for Computational Linguistics.

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. 2021. Guided integrated gradients: An adaptive path method for removing noise. In CVPR, pages 5050–5058. Computer Vision Foundation / IEEE.

Shicheng Liu and Minghui Zhu. 2025. UTILITY: utilizing explainable reinforcement learning to improve reinforcement learning. In ICLR. OpenReview.net.

Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. 2024. Discuss before moving: Visual language navigation via multi-expert discussions. In ICRA, pages 17380–17387. IEEE.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In ICML, volume 48 of JMLR Workshop and Conference Proceedings, pages 1928–1937. JMLR.org.

Alexander Mott, Daniel Zoran, Mike Chrzanowski, Daan Wierstra, and Danilo Jimenez Rezende. 2019. Towards interpretable reinforcement learning using attention augmented agents. In NeurIPS, pages 12329–12338.

Paul Novello, Thomas Fel, and David Vigouroux. 2022. Making sense of dependence: Efficient black-box explanations using dependence measure. In NeurIPS.

OpenAI. 2023. GPT-4 technical report. CoRR, abs/2303.08774.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In ICCV, pages 618–626. IEEE Computer Society.

Suraj Srinivas and François Fleuret. 2021. Rethinking the role of gradient-based attribution methods for model interpretability. In ICLR. OpenReview.net.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In ICML, volume 70 of Proceedings of Machine Learning Research, pages 3319–3328. PMLR.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In NeurIPS.

Yuchen Wu, Pengcheng Zhang, Meiying Gu, Jin Zheng, and Xiao Bai. 2024. Embodied navigation with multi-modal information: A survey from tasks to methodology. Inf. Fusion, 112:102532.

Jiahao Yu, Wenbo Guo, Qi Qin, Gang Wang, Ting Wang, and Xinyu Xing. 2023. AIRS: explanation for deep reinforcement learning based security applications. In USENIX Security Symposium, pages 7375–7392. USENIX Association.

Tom Zahavy, Nir Ben-Zrihem, and Shie Mannor. 2016. Graying the black box: Understanding dqns. In ICML, volume 48 of JMLR Workshop and Conference Proceedings, pages 1899–1908. JMLR.org.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. 2024. Navid: Video-based VLM plans the next step for vision-and-language navigation. In Robotics: Science and Systems.

Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. 2024a. Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In ECCV (7), volume 15065 of Lecture Notes in Computer Science, pages 260–278. Springer.

Gengze Zhou, Yicong Hong, and Qi Wu. 2024b. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In AAAI, pages 7641–7649. AAAI Press.

A Appendix

A.1 Background

A.1.1 Vision-and-Language Navigation

In the Vision-and-Language Navigation (VLN) task, an agent is required to interact with the environment and navigate to a target location following a given natural language instruction. This task can naturally be formulated as a Markov Decision Process (MDP), which is defined by a five-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, representing the state space, action space, state transition function, and reward function, respectively.

In the context of navigation tasks, $\mathcal{S}$ denotes the set of all RGB images within the environment as well as the associated navigation instructions. $\mathcal{A}$ represents the actions available to the navigation agent, which may involve selecting a navigation waypoint in discrete environments or determining both the direction and distance in continuous environments. The transition function $\mathcal{P}$ governs how the agent moves to a new state based on the current position and selected action, while also receiving a new environmental observation. The reward function $r(s, a)$ provides immediate feedback for taking action a in state s, typically defined based on changes in distance to the target location or task completion status. This MDP process repeats until the agent successfully completes the task, fails, or reaches a predefined maximum number of steps.

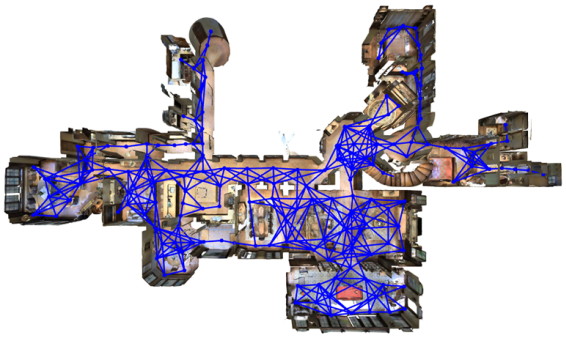


Figure 4: The undirected graph representation in the MatterPort3D simulator.

VLN tasks can be categorized into discrete and continuous environments based on how the environment is represented. In discrete environments, the space is modeled as an undirected graph $G = (V, E)$, where V represents navigable waypoints and E denotes the connections between them. At each timestep t, the agent receives an instruction I and an RGB panoramic observation O_t, infers the next action a_t, and selects from the navigable waypoints within its action space $\mathcal{A}_t$. The agent continues to execute a sequence of actions until reaching the target location or exceeding the step limit.

In contrast, continuous navigation environments remove the graph structure, allowing the agent to move freely within the space. The agent receives an instruction I and panoramic RGB images O_t, with an action space $\mathcal{A}_t = (\theta, d)$, where θ indicates the angle between the current heading and the target direction, and d represents the distance to move in that direction. The agent continuously selects actions from this continuous space until the goal is reached or the step limit is exceeded.

This work focuses primarily on the interpretability of LLM-based VLN agents in discrete environments.

A.1.2 Interpretability of Black-Box Navigation Agents

Under the black-box assumption, researchers do not have access to the internal structures, parameters, or gradient information of the navigation agent. Instead, only the instruction I, observation O_t, and

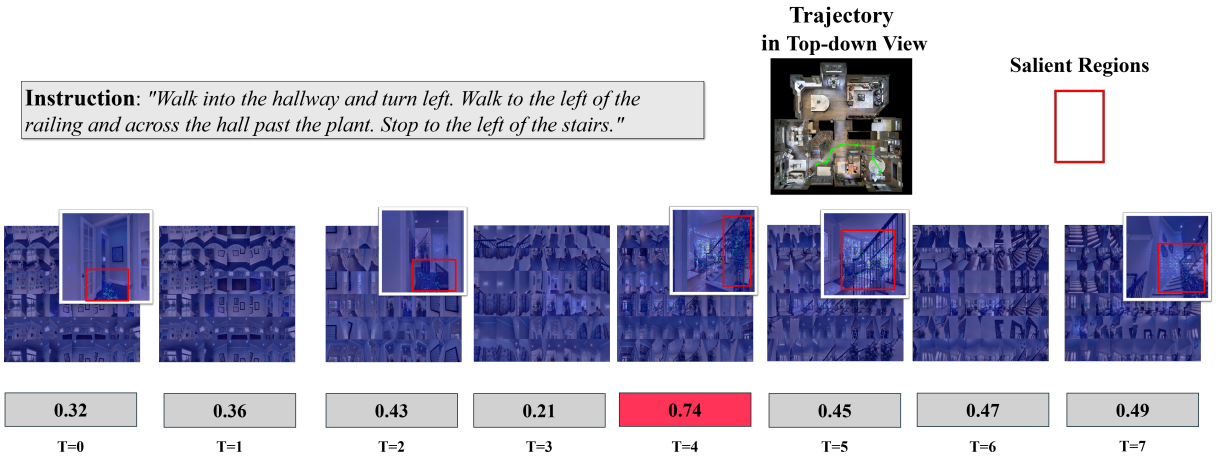


Figure 5: An example explanation of DEFT.

the predicted action a_t at each timestep are observable. The interpretability analysis of navigation models is conducted from two complementary perspectives: time-step level interpretability and feature-level interpretability.

Time-Step Level Interpretability. This aspect aims to identify the top-K critical timesteps along the navigation trajectory. These key decision points are those at which failure to make a correct decision would highly likely prevent the agent from successfully completing the navigation task according to the instruction.

Feature-Level Interpretability. This component seeks to understand which elements of the input information most influence the agent's decision-making process. Specifically, in the VLN task, given a navigation instruction I, it is possible to analyze, at the feature level, how much attention the agent allocates to different parts of the observation O_t. In particular, this work aims to identify, at each timestep t, either the most influential set of pixels $P_t \subseteq \{1, \ldots, H\} \times \{1, \ldots, W\}$ within the observation that most significantly affect the agent's action decision. Here, H and W represent the height and width of the input image, respectively, and N denotes the total number of segmented objects in the scene.

A.2 Baseline Models

To validate the effectiveness of DEFT, we compare it against state-of-the-art interpretability methods, ranging from white- to black-box baselines. Note that, given the black-box setting of our study, the white-box baselines described below are applied to the gradient-accessible surrogate models (described in Sec. 4) acting as proxies.

A.2.1 Timestep Criticality Baselines

StateMask (SM) (Cheng et al., 2023) employs a masking policy to measure importance via reward degradation in general RL. It provides the theoretical basis for our mask head but is evaluated here in its standard form, notably lacking the specific pre-training mechanism.

Value-Based (VB) (Huang et al., 2018) is a white-box metric that defines criticality based on action-value divergence. We explicitly utilize the trained surrogate's critic network to output the action values. A step is deemed critical if the value of the optimal action significantly exceeds the average value of all possible actions.

Gradient-Based (GB) (Srinivas and Fleuret, 2021) is a white-box approach applied to the surrogate model. It utilizes the gradients of the output logits, i.e., the log probability $\log p(a_t)$ of the executed action, to explain the importance of each step.

A.2.2 Feature Attribution Baselines

We compare our approach against a diverse set of attribution methods, ranging from white- to black-box baselines.

Integrated Gradients (IG) (Sundararajan et al., 2017) is an axiomatic white-box method applied to surrogate models to mitigate gradient saturation. It integrates gradients along a linear path, satisfying sensitivity and implementation invariance. We employ this standard IG as the base estimator within our ensemble framework.

Guided-IG (Kapishnikov et al., 2021) enhances IG by introducing an adaptive path strategy. Unlike

the fixed linear integration, it greedily mitigates noise by avoiding high-gradient artifacts, yielding cleaner attribution maps that focus on semantically relevant features.

SMDL (Chen et al., 2024b) formulates interpretation as a combinatorial optimization problem via submodular subset selection. It identifies sparse, robust image regions by maximizing a unified objective of model confidence, semantic effectiveness, and consistency, effectively filtering out dense noise.

HSIC (Novello et al., 2022) is a statistical black-box method leveraging the Hilbert-Schmidt Independence Criterion. It estimates feature importance by measuring non-linear dependencies between inputs and outputs in Reproducing Kernel Hilbert Spaces, eliminating the need for gradient access.

A.3 Evaluation Metrics

We employ comprehensive metrics to assess interpretability at both timestep and feature levels. All results represent the average of three independent runs to mitigate randomness. Moreover, we also include the trajectory similarity metrics here, which used in surrogate fidelity evaluation.

A.3.1 Timestep-Level Metrics

We adopt *Relative Reward Difference (RRD)* (Yu et al., 2023) and *Fidelity* (Guo et al., 2021) to validate the criticality of identified steps.

RRD normalizes the reward degradation caused by masking critical steps against random perturbations. For N episodes:

$$\text{RRD} = \frac{1}{N}\sum_{i=1}^{N}\frac{|R^{(i)}_{mask} - R^{(i)}_{org}|}{|R^{(i)}_{rand} - R^{(i)}_{org}|} \tag{14}$$

where R_{org}, R_{mask}, and R_{rand} denote the cumulative rewards of the original episode, the episode with critical steps masked, and the episode with random steps masked, respectively. An RRD > 1 indicates that the identified steps are more critical than random ones.

Fidelity measures the alignment between predicted importance and actual impact, incorporating a sequence penalty:

$$\text{Fidelity} = \frac{1}{N}\sum_{i=1}^{N}(\log p^{(i)}_d - \log p^{(i)}_l) \tag{15}$$

Here, $p^{(i)}_d$ is the normalized reward drop. $p^{(i)}_l = l_i/L_i$ acts as a penalty, where L_i is the original trajectory length, and l_i is the length of the longest continuous sub-sequence within the top-K critical steps. High fidelity implies accurate identification of critical steps.

A.3.2 Feature-Level Metrics

Since LLMs do not expose output logits, we strictly evaluate *Action Consistency*, i.e., whether the agent retains its original decision under perturbation.

Insertion (Ins) & Deletion (Del). We report the *Average Action Consistency* across varying pixel perturbation ratios (25%, 50%, 75%, see ratio impact in Appx. A.9). (1) *Ins* measures the agent's ability to recover the original action as pixels are added to a blank canvas based on importance. Higher Ins indicates better feature prioritization. (2) *Del* measures the decline in action consistency as important pixels are removed (blurred). Lower Del indicates accurate identification of vital features.

μFidelity (μ_F) assesses the correlation between accumulated attribution scores and the model's behavioral deviation under perturbation:

$$\mu_F = \underset{S\subseteq\mathcal{X},|S|=k}{\text{Corr}}\left(\sum_{j\in S} g(\mathcal{O})_j,\ \Delta(f(\mathcal{O}), f(\mathcal{O}_{\text{m}}))\right) \tag{16}$$

where $\mathcal{X}$ is the total input pixels, S is a random subset of size k, and $g(\mathcal{O})_j$ denotes the importance score of pixel j. The term $\Delta(\cdot)$ quantifies the agent's output divergence induced by masking subset S. Higher correlation implies superior faithfulness.

A.3.3 Trajectory Similarity Metrics

We use normalized Dynamic Time Warping (nDTW) (Ilharco et al., 2019) and Coverage weighted by Length Score (CLS) (Jain et al., 2019) to quantify the trajectory similarity in surrogate fidelity evaluation (See Sec. 6.4 for details).

nDTW is a metric that evaluates the fidelity of the agent's trajectory by measuring the similarity between the predicted path P and the reference path R, while strictly respecting the temporal order of the nodes. It first calculates the DTW cost, which finds the optimal alignment (warping) W that minimizes the cumulative distance between the ordered nodes of P and R:

$$\text{DTW}(P,R) = \min_{W\in\mathcal{W}}\sum_{(i_k,j_k)\in W}\delta(p_{i_k}, r_{j_k}) \tag{17}$$

where p_i and r_j are nodes in the predicted and reference paths, respectively, and $\delta(\cdot)$ denotes the

distance metric (e.g., Euclidean distance). To make the metric comparable across different path lengths, nDTW normalizes the cost by the length of the reference path $|R|$ and a success threshold constant d_{th}:

$$\text{nDTW}(P, R) = \exp\left(-\frac{\text{DTW}(P, R)}{|R| \cdot d_{th}}\right) \quad (18)$$

This formulation yields a score in the range $[0, 1]$, where a higher score indicates that the agent followed the instructions and the reference path more closely.

CLS assesses fidelity by combining how well the agent covered the reference path with how efficiently it did so. It is defined as the product of Path Coverage (PC) and Length Score (LS):

$$\text{CLS}(P, R) = \text{PC}(P, R) \cdot \text{LS}(P, R) \quad (19)$$

Path Coverage measures the extent to which the nodes in the reference path R are visited by the agent's path P. Unlike nDTW, it is order-invariant for the coverage calculation:

$$\text{PC}(P, R) = \frac{1}{|R|} \sum_{r \in R} \exp\left(-\frac{\min_{p \in P} d(r, p)}{d_{th}}\right) \quad (20)$$

Length Score penalizes the agent if the length of the predicted path, $\text{PL}(P)$, deviates from the expected optimal length, $\text{EPL}(P, R) = \text{PC}(P, R) \cdot \text{PL}(R)$. This ensures the agent is not rewarded for wandering:

$$\text{LS}(P, R) = \frac{\text{EPL}(P, R)}{\text{EPL}(P, R) + |\text{EPL}(P, R) - \text{PL}(P)|} \quad (21)$$

A.4 User Study

To rigorously evaluate the interpretability of DEFT from a human perspective, we conducted a user study to benchmark it against existing baselines. This study aims to verify whether our dual-view explanations align with human cognition and effectively assist users in understanding agent behaviors.

A.4.1 Participants and Setup

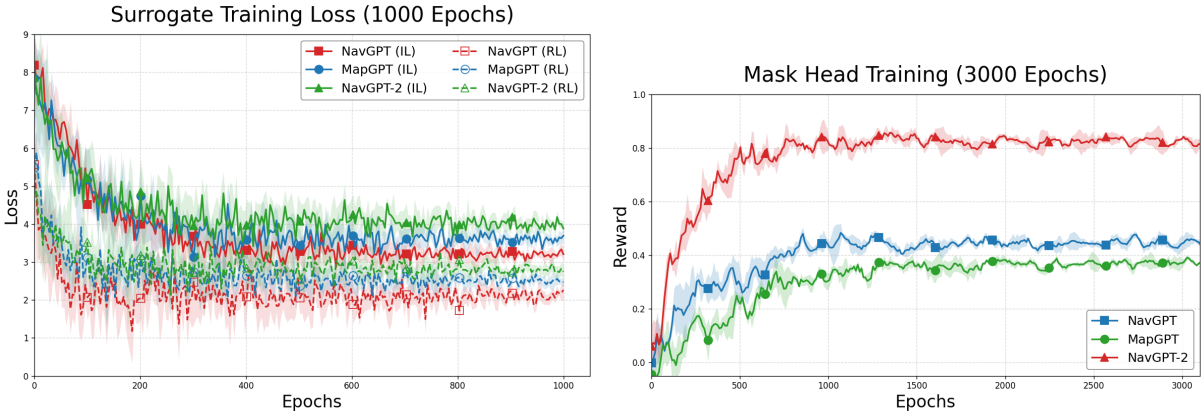
To mitigate bias, we recruited 20 participants with diverse backgrounds in embodied AI and interpretable machine learning: 25% domain experts, 50% students/practitioners with some familiarity, and 25% laypersons with no prior exposure to the field. This composition ensures both expert relevance and general intuitiveness. The evaluated episodes cover diverse failure modes (e.g., visual grounding errors, directional misinterpretations, and instruction skipping) across varied scene types, including stairs, hallways, and rooms. Prior to the study, we provided a brief tutorial explaining the VLN task—where an agent navigates based on natural language instructions—and how to interpret the visual comparison interfaces (i.e., criticality scores for timesteps and saliency maps for visual features). All participants confirmed their understanding of the task before proceeding. Given the high cognitive demands of VLN interpretation, which requires detailed visual and instruction-level reasoning, we prioritized response quality over large-scale but noisy crowd-sourcing. Accordingly, each participant was assigned 8 episodes. We designed the survey to evaluate the explanation approaches from three distinct perspectives:

- **Intuitive Alignment:** Which explanation aligns more closely with human intuition regarding navigation decisions?
- **Reasoning Understanding:** Which explanation best aids in understanding why the agent made specific decisions to accomplish the instruction?
- **Failure Diagnosis:** Which explanation is most helpful in identifying the root cause of a navigation failure?

A.4.2 Evaluation on Intuition and Understanding

For the first two perspectives, we presented participants with four distinct navigation episodes from the R2R validation set. Since DEFT provides a dual-view explanation, for a fair comparison, we constructed a composite baseline by combining the top-performing disjoint methods identified in our experiments: Value-Based + SMDL. Specifically, this baseline uses Value-Based to identify critical steps and then applies SMDL to generate visual heatmaps for steps. For each episode, participants reviewed the navigation instruction and trajectory, accompanied by explanations from both DEFT and the composite baseline. They were asked to select the approach that better aligned with their intuition (Q1) and better clarified the agent's reasoning (Q2).

The results, averaged across the four episodes, are illustrated in Fig. 3. For intuitive alignment, DEFT was favored in 78% of the trials. Participants cited that DEFT's visual explanations appeared more coherent and focused compared to the baseline, where the feature attribution (SMDL) often highlighted irrelevant regions even when the



(a) Surrogate training convergence and efficiency.

(b) *Mask Head* training convergence and efficiency.

Figure 6: Training convergence and efficiency.

timestep (Value-Based) was correct—a symptom of the alignment gap discussed in Sec. 1. Similarly, for reasoning understanding, DEFT was chosen in 75% of the cases. Participants reported that the unified view effectively let the navigation logic transparent and easier to follow.

A.4.3 Evaluation on Failure Diagnosis

To address the third perspective, we designed a "debugging" task involving four distinct failure cases (e.g., the staircase failure). We provided participants with replay videos where the agent deviated from the correct path. For each case, participants analyzed explanations from DEFT and the composite baseline to identify the specific cause of the error (e.g., "misinterpreted the staircase direction" vs. "ignored the instruction").

As shown in Fig. 3, DEFT demonstrated superior utility in diagnostics. Averaged across the failure cases, participants successfully pinpointed the exact cause of error in 64% of trials using DEFT. In contrast, when relying on the disjoint method (Value-Based + SMDL), the success rate dropped to 30%. Participants noted that while the baseline often flagged the correct step, the accompanying visual heatmap was frequently noisy or contradictory due to the agent's instability, making it difficult to correlate the timing of the error with specific visual evidence. These results confirm that DEFT's unified framework significantly lowers the cognitive load for humans when diagnosing complex long-horizon failures compared to simple combinations of existing methods. Finally, 6% of trials were deemed *unhelpful* for both methods. Post-hoc analysis reveals that these cases mainly arise from surrogate limitations in ambiguous scenarios: (1) diffuse saliency (∼4%) in repetitive environments (e.g., a long corridor with identical doors), leading to diffuse heatmaps; (2) mask head misses (∼2%) under subtle instruction dependencies (e.g., "stop after the second door").

A.5 Training Convergence and Efficiency

In DEFT, the surrogate (*Action Head*) and *Mask Head* are trained in a two-stage, sequential manner.

- Stage 1 (Feature-level interpretability). We train the surrogate to approximate the target policy, optimizing the shared backbone and *Action Head* to capture high-fidelity visual–linguistic representations and action decisions.
- Stage 2 (Timestep-level interpretability). We freeze the backbone and optimize only the *Mask Head* by minimizing reward degradation under counterfactual perturbations, enabling identification of temporally critical steps.

This design has two advantages: (1) Freezing the backbone allows the *Mask Head* to rely on task-relevant latent features learned in Stage 1 rather than raw inputs. (2) Decoupled training reduces trainable parameters in Stage 2, simplifying optimization and improving convergence stability.

To validate the optimization stability and computational efficiency of DEFT, we visualize the training dynamics of both the surrogate models (*Action Head*) and the temporal critic (*Mask Head*). Fig. 6 presents the loss and reward curves across three target agents: NavGPT, MapGPT, and NavGPT-2.

A.5.1 Surrogate Convergence

Fig. 6a illustrates the training loss of the surrogate models over 1,000 epochs. Both the IL loss (solid lines) and RL loss (dashed lines) exhibit a sharp

Table 4: Computational cost of the training and inference phases for MapGPT.

Phase	Component	Mem.	Time
Training	Ensemble Surrogates	5.18GB	2.4h×4
	Mask Head	1.20GB	1.14h
Inference	Ensemble IG	4.86GB	101.93s
	Mask Head	0.70GB	3.03s

Table 5: Performance comparison for MapGPT under different reward settings.

Reward	RRD↑	Fidelity↑
r_{stop}	1.03	-0.74
$r_{stop} + r_{prog}$	1.22	0.08
$r_{stop} + r_{fid}$	1.15	-0.08
$r_{stop} + r_{prog} + r_{fid}$	1.28	0.15

decline in the initial phase and achieve asymptotic stability at approximately 600 epochs. Notably, despite the inherent behavioral differences among the three target agents, the training curves for all surrogates show consistent convergence patterns with low variance (indicated by the narrow shaded regions). This confirms that our surrogates can efficiently approximate diverse black-box behaviors.

A.5.2 *Mask Head* Optimization

Fig. 6b depicts the reward evolution of the Mask Head over 3,000 epochs. The reward steadily increases, reflecting the model's improving ability to balance masking sparsity with reward preservation. We observe that the training stabilizes around 1,500 epochs for all agents. Interestingly, NavGPT-2 achieves a significantly higher converged reward (≈ 0.8) compared to NavGPT and MapGPT (≈ 0.4). This discrepancy likely stems from NavGPT-2's higher stability (as noted in Sec. 6.4), which makes its critical steps easier to identify. Overall, these results demonstrate that DEFT can be trained efficiently with stable convergence, making it a practical solution for interpreting LLM-based VLN agents.

A.6 Computational Cost

Although DEFT incurs additional training cost for the surrogate and *Mask Head*, it provides substantial efficiency gains at inference compared to black-box perturbation-based methods, which require repeated re-querying of the target LLM.

Table 4 reports a detailed computational breakdown, using MapGPT as the target agent with $K = 4$ surrogates (as in our main experiments). Training cost is measured by peak GPU memory and total runtime (1,000 iterations for the surrogate and 3,000 for the *Mask Head*), while inference cost is reported per sample, including one feature-level heatmap and a full trajectory for timestep-level interpretation.

Under the same setup (MapGPT as the target), each HSIC perturbation takes ∼10s since it requires re-querying the (multimodal) LLM rather than a learnable target. With the default setting of over 750 perturbations per attribution map, generating a single heatmap takes ∼125 minutes, significantly slower than our EIG method (101.93s). HSIC also consumes 1,229,250 tokens and costs $3.73 per heatmap, whereas DEFT introduces no additional inference overhead. Even accounting for training, DEFT becomes more efficient after interpreting 5 samples. Since training is a one-time offline cost, transparent to end users, the method remains efficient in practice while delivering higher-fidelity explanations.

A.7 Impact of Reward Design

Our reward formulation is grounded in the fundamental objectives of the VLN task, ensuring alignment with both agent performance and human intuition. Specifically, our composite reward consists of three complementary components:

- Progress Reward: It measures the reduction in geodesic distance to the target, reflecting the most basic goal of navigation, whick is getting closer to the goal.
- Path-fidelity Reward: By utilizing the nDTW metric between the agent's trajectory and the ground-truth path, it encourages the agent to follow the linguistically prescribed route, aligning the explanation with the instruction-following nature of the task.
- Negative Stop Reward: A sparse reward that penalizes failure and rewards correct termination, providing a global signal for task completion.

The synergy of these components ensures that the criticality identified by the *Mask Head* corresponds to steps that are essential for approaching the goal, following instructions, and successful termination–all of which are core to human judgment of a pivotal decision.

To assess robustness under different reward settings, we conduct an ablation using MapGPT as the target agent, comparing the proposed dense re-

Table 6: The impact of ensemble size K on Ins/Del metrics (at 25% perturbation).

Model	Metric	K=1	K=2	K=3	K=4	K=5
NavGPT	Ins ↑	0.74	0.73	0.72	0.75	0.75
	Del ↓	0.76	0.76	0.77	0.76	0.77
MapGPT	Ins ↑	0.67	0.69	0.69	0.70	0.68
	Del ↓	0.72	0.70	0.71	0.69	0.69
NavGPT-2	Ins ↑	0.76	0.74	0.78	0.78	0.78
	Del ↓	0.74	0.77	0.77	0.75	0.77

ward with a strictly sparse variant, e.g., r_{stop} only, which reflects only the final success or failure. Results in Table 5 show that DEFT remains functional under sparse rewards, but its ability to capture fine-grained temporal criticality degrades, approaching gradient-based baselines. In contrast, incorporating r_{prog} and r_{fid} provides dense, task-aligned supervision, leading to more accurate and interpretable critical step identification.

Under sparse rewards, performance can be further improved by introducing auxiliary evaluators (e.g., path-ranking models) as pseudo-reward signals or by distilling criticality from state-aware teacher policies, reducing reliance on manual reward design.

A.8 Impact of Ensemble Size

We investigate the influence of the ensemble size (K) on attribution quality across three models. Table 6 summarizes the *Insertion* and *Deletion* performance at the strictest 25% perturbation ratio as K increases from 1 to 5.

A.8.1 Insertion Analysis

Increasing the ensemble size K generally yields a clear performance gain in the *Insertion* metric. For instance, NavGPT-2 improves from 0.76 with a single surrogate ($K = 1$) to a stable peak of 0.78 at $K \geq 3$, while MapGPT rises from 0.67 to 0.70 ($K = 4$). This trend confirms that aggregating gradients from multiple surrogates effectively mitigates the bias of individual proxies, recovering more robust and information-dense visual features. However, performance tends to saturate or slightly fluctuate beyond $K = 4$ (e.g., MapGPT drops to 0.68 at $K = 5$). This suggests that a moderate ensemble size (e.g., $K = 3$ or 4) achieves the optimal trade-off between interpretation fidelity and computational cost, as the marginal information gain diminishes with further expansion.

A.8.2 Deletion Analysis

For the *Deletion* metric (lower is better), the ensemble strategy proves particularly effective for agents with higher behavioral variance. Taking MapGPT as an example, the score improves notably from 0.72 ($K = 1$) to 0.69 ($K = 4$), demonstrating that leveraging consensus from diverse surrogates helps refine attribution and remove critical pixels more precisely. In contrast, for the highly deterministic NavGPT-2, the single surrogate ($K = 1$) yields a marginally better score (0.74) compared to the ensemble average (≈ 0.76). This discrepancy indicates that our method offers the most substantial improvements for unstable agents. For highly accurate and stable agents like NavGPT-2, the diversity among surrogates is naturally lower, limiting the additional benefits of ensemble aggregation.

Based on the overall stability and performance saturation observed across metrics, we empirically set the default ensemble size to $K = 4$. This choice strikes an optimal balance between attribution fidelity and computational efficiency.

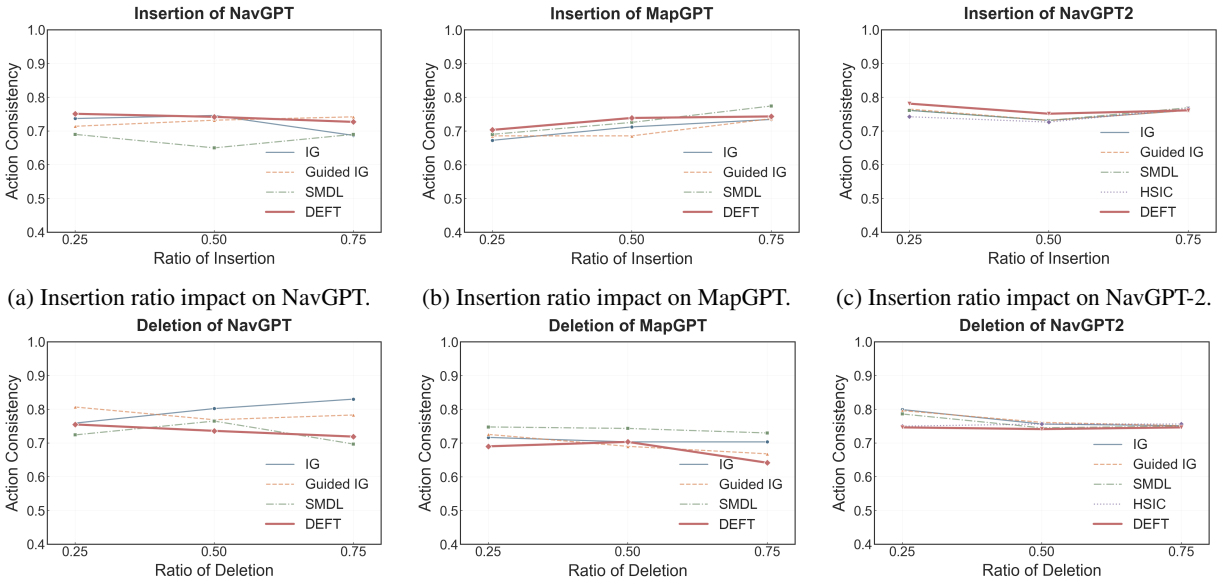
A.9 Impact of Insertion/Deletion Ratios

To assess the robustness of our visual explanations, we analyze the effect of varying pixel perturbation ratios (25%, 50%, and 75%) for both *Insertion* and *Deletion* metrics. Fig. 7 illustrates the *Action Consistency* with the original decision across three agents (NavGPT, MapGPT, NavGPT-2).

A.9.1 Insertion Analysis

As shown in Fig. 7a–7c, a faithful attribution method is expected to yield increasing action consistency as more pixels are inserted. This trend is clearly observed for MapGPT. In contrast, NavGPT exhibits irregular fluctuations, which we attribute to its inherent behavioral instability. As discussed in Sec. 6.4, NavGPT fails to maintain high self-consistency ($\sim$0.6) even with full visual input, making its responses to partial observations highly stochastic. For NavGPT-2, the absence of a rising trend suggests susceptibility to visual distraction. While initial salient pixels suffice for reasoning, inserting subsequent non-salient regions introduces noise that interferes with judgment, leading to decision inconsistency.

Notably, DEFT achieves the strongest performance at the most stringent 25% insertion ratio, indicating that it successfully identifies the most information-dense pixels that suffice to recover the agent's decision. As the insertion ratio increases



(a) Insertion ratio impact on NavGPT. (b) Insertion ratio impact on MapGPT. (c) Insertion ratio impact on NavGPT-2.

(d) Deletion ratio impact on NavGPT. (e) Deletion ratio impact on MapGPT. (f) Deletion ratio impact on NavGPT-2.

Figure 7: Ratio impact for Insertion/Deletion across models.

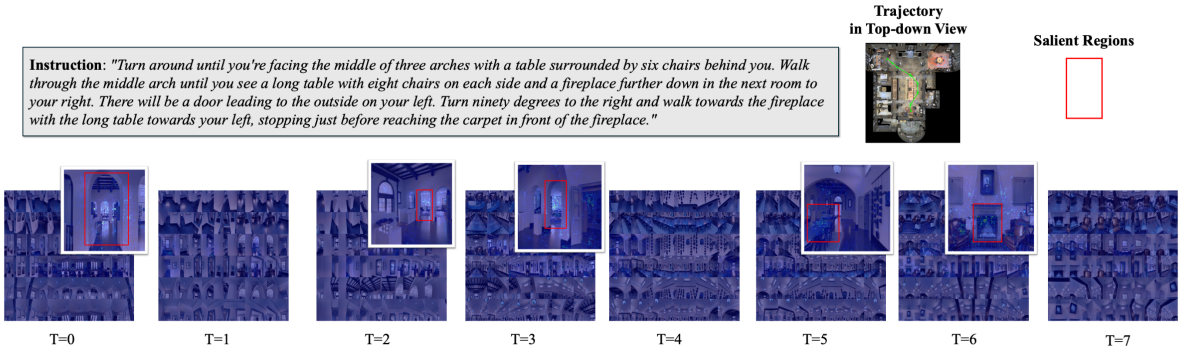


Figure 8: **ID: 2539_0 (well-explained, NavGPT)**. $nDTW = 0.19, CLS = 0.26$; $Ins = 0.80\uparrow, Del = 0.20\downarrow$.

to 75%, performance differences across methods naturally diminish, since most visual information is restored and pixel prioritization becomes less critical.

A.9.2 Deletion Analysis

For the *Deletion* metric (Fig. 7d– 7f), lower consistency indicates stronger attribution quality, as removing the identified pixels should significantly disrupt the agent's decision. Our method exhibit a clear downward trend as the deletion ratio increases from 25% to 75% for all the target models, whereas other methods suffer from varying degrees of fluctuation.

Importantly, DEFT attains competitive or lower consistency at the 25% deletion ratio compared to baselines. This sharp performance drop after removing only a small fraction of pixels confirms that DEFT precisely localizes decision-critical visual evidence. As the deletion ratio approaches 75%, distinctions among methods become less pronounced, since removing most of the image inevitably degrades performance for all approaches.

A.10 Additional Qualitative Analysis of Success and Failure Cases

The approximation gap between surrogate models and the target agent is inevitable due to differences in architecture, scale, and pre-training, especially under non-linear reasoning in LLM-based policies. DEFT mitigates this issue through its ensemble-surrogate design, which aggregates multiple surrogates to reduce individual bias and improve robustness. As shown in our experiments, this strategy achieves consistently higher fidelity than single-surrogate baselines, indicating a closer approximation of the target agent's decision behavior. In challenging cases, the ensemble further

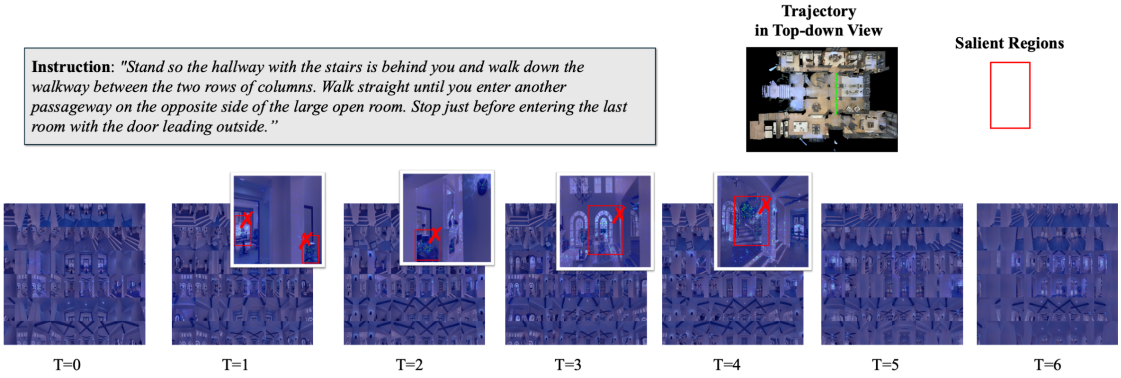


Figure 9: **ID: 6565_2 (poorly-explained, NavGPT)**. $nDTW = 0.06, CLS = 0.46$; $Ins = 0.29\uparrow, Del = 0.71\downarrow$.

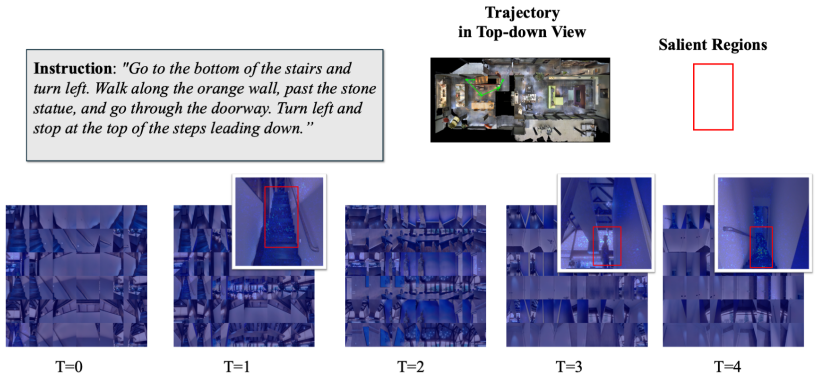


Figure 10: **ID: 7238_0 (well-explained, NavGPT-2)**. $nDTW = 0.24, CLS = 0.20$; $Ins = 0.8\uparrow, Del = 0.2\downarrow$.

stabilizes attribution, allowing DEFT to localize key landmarks even when actions slightly deviate. Moreover, the surrogates learn robust internal representations during training, enabling IG-based attribution to trace gradients through a stable latent space and uncover task-aligned reasoning patterns beyond surface-level action matching.

To further analyze corner cases, we provide additional qualitative examples showing that behavioral imitation and attribution consistency are partially decoupled. Specifically, we present four cases from both NavGPT and NavGPT-2 where surrogate trajectories deviate from the target agent, yet the resulting attribution maps vary in quality.

Case 1. On NavGPT, as shown in Fig. 8, DEFT can produce robust explanations despite low trajectory similarity. The heatmaps consistently highlight key landmarks, e.g., "middle arch" in early steps and the "long table" and "fireplace" in later steps. While individual surrogates may generate noisy attributions due to imperfect trajectory alignment, the ensemble mechanism reinforces consensus regions and suppresses outliers. This collaborative effect yields a more stable and faithful attribution map, showing that the surrogates capture the LLM's perception of key landmarks even without perfectly replicating its path.

Case 2. In contrast, a failure case with particularly poor surrogate imitation is illustrated in Fig. 9. This example yields Ins = 0.29 and Del = 0.71, where the low Insertion and high Deletion scores indicate weak causal faithfulness and mislocalized evidence. Although the instruction requires walking along a "walkway", the heatmap incorrectly emphasizes irrelevant objects such as "vases", "wall decorations", and "stairs" in Steps 1–4. This suggests that when surrogate alignment substantially degrades, the attribution may reflect surrogate-specific biases rather than the LLM's true reasoning process.

Case 3. A similar pattern is observed across other targets (e.g., NavGPT-2). As shown in Fig. 10, the surrogate's trajectory poorly matched NavGPT-2 (nDTW = 0.24, CLS = 0.20), reflecting a significant path imitation failure. Despite this, DEFT's heatmaps show robust visual grounding, with Ins = 0.8 and Del = 0.2, indicating faithful capture of the LLM's internal logic. Step 1 highlights the "stairs" from the first sub-instruction; Steps 2–3 emphasize the "stone statue" landmark for the "past the stone statue" instruction; and the final step focuses

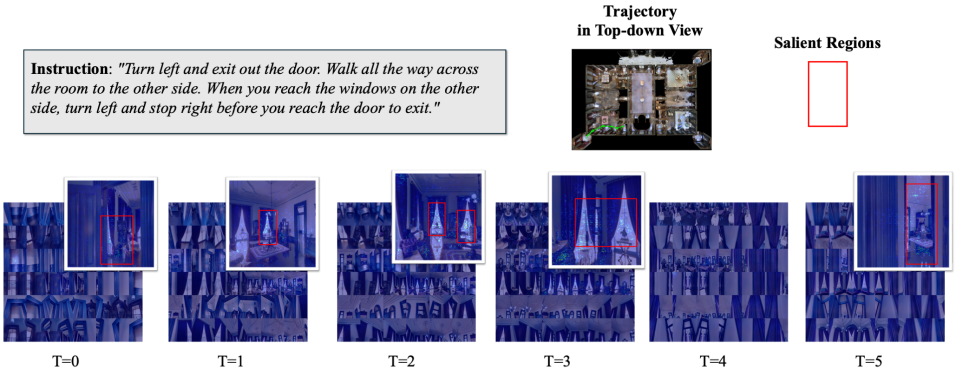


Figure 11: **ID: 5166_1 (well-explained, NavGPT-2)**. $nDTW = 0.36, CLS = 0.74$; $Ins = 0.67\uparrow, Del = 0.33\downarrow$.

on the "steps leading down", the explicit stopping condition.

Case 4. A similar trend is observed in Fig. 11, where the trajectory exhibits a movement mismatch (nDTW = 0.36). Despite this, it achieves high Ins and low Del scores (0.67 and 0.33), reflecting faithful capture of the LLM's reasoning. DEFT accurately tracks the reasoning chain: Step 0 highlights the "door" for the "exit out the door" instruction; Steps 1–3 focus on the "windows", which serve as the pivot for the "turn left" maneuver; and Step 5 peaks at the "exit door", marking the stopping point.

Using NavGPT and NavGPT-2 as the targets, we observe that in well-explained cases, DEFT accurately localizes the agent's perceptual focus despite low trajectory fidelity, with insertion and deletion metrics confirming the causal relevance of highlighted regions. In contrast, failure cases reveal misleading attributions, highlighting the limitations of the approach. Overall, these examples demonstrate that attribution maps can remain spatially consistent under moderate divergence, while also exposing scenarios where surrogate approximation errors degrade explanation quality.