Active Test-time Vision-Language Navigation

Heeju Ko[1] **Sung June Kim**[1] **Gyeongrok Oh**[1] **Jeongyoon Yoon**[1]
Honglak Lee[2] **Sujin Jang**[3] **Seungryong Kim**[4] **Sangpil Kim**[1]*

[1] Korea University [2] University of Michigan [3] Samsung AI Center, DS Division [4] KAIST AI

Abstract

Vision-Language Navigation (VLN) policies trained on offline datasets often exhibit degraded task performance when deployed in unfamiliar navigation environments at test time, where agents are typically evaluated without access to external interaction or feedback. Entropy minimization has emerged as a practical solution for reducing prediction uncertainty at test time; however, it can suffer from accumulated errors, as agents may become overconfident in incorrect actions without sufficient contextual grounding. To tackle these challenges, we introduce ATENA (Active TEst-time Navigation Agent), a test-time active learning framework that enables a practical human-robot interaction via episodic feedback on uncertain navigation outcomes. In particular, ATENA learns to increase certainty in successful episodes and decrease it in failed ones, improving uncertainty calibration. Here, we propose *mixture entropy optimization*, where entropy is obtained from a combination of the action and pseudo-expert distributions—a hypothetical action distribution assuming the agent's selected action to be optimal—controlling both prediction confidence and action preference. In addition, we propose a *self-active learning* strategy that enables an agent to evaluate its navigation outcomes based on confident predictions. As a result, the agent stays actively engaged throughout all iterations, leading to well-grounded and adaptive decision-making. Extensive evaluations on challenging VLN benchmarks—REVERIE, R2R, and R2R-CE—demonstrate that ATENA successfully overcomes distributional shifts at test time, outperforming the compared baseline methods across various settings.

1 Introduction

Vision-Language Navigation (VLN) is a fundamental multimodal task in embodied AI systems, which requires an agent to interpret natural language instructions and navigate through complex visual environments [1]. Despite recent advancements in VLN, distributional shifts between offline training and online testing environments remain a critical challenge for robust and reliable deployment [2, 3]. To address this issue, many prior works focus on enhancing generalizability during offline training to better handle potential domain shifts [4, 5, 6]. However, these approaches are limited in addressing real-world variability, as collecting diverse expert demonstrations across environments is often impractical. Therefore, test-time adaptation (TTA)—the ability to directly adapt to test-time environments—is crucial for real-world robotic navigation.

Test-Time Adaptation (TTA) refines a pre-trained model during inference using unsupervised signals—such as prediction entropy [7], consistency [8], or pseudo-labels [9]—offering a practical yet challenging approach to improving robustness at test time. Entropy minimization is one of the widely accepted TTA strategies, based on the assumption that greater model certainty correlates with improved accuracy during inference [7, 10, 11]. However, applying entropy minimization uniformly

*Corresponding author: spk7@korea.ac.kr

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-1490

across all decision points in sequential tasks such as VLN may cause the policy to overfit to failure patterns, thereby increasing the likelihood of incorrect actions. Consequently, the policy accumulates errors throughout iterations and loses resilience on failure cases. Thus, blindly increasing prediction certainty without considering the navigation status leads to suboptimal behaviors.

How, then, can we provide the contextual cues necessary for a VLN agent to properly leverage entropy as a meaningful signal at test time? To answer this, we propose an active learning (AL) [12, 13] strategy, enabling the agent to query a human oracle for necessary contextual labels. Here, we must consider practical constraints in the online test-time navigation setting for utilizing human feedback: (1) Latency—human involvement should not introduce delays during navigation rollout; and (2) Accessibility—human input must be intuitive, requiring minimal expertise and effort. Therefore, it is unrealistic to expect human feedback at the same level of detail as stepwise expert demonstrations in VLN at test time.

To address these practical concerns, we define the active label as an episodic, binary evaluation indicating navigation success or failure, rather than requiring detailed stepwise supervision. Inspired by the uncertainty sampling paradigm in AL [14, 15, 16], we design the agents to inquire feedback whenever the average action uncertainty throughout each navigation task exceeds a predefined threshold. Given the sparsity of the feedback, we introduce a novel technique called *mixture entropy optimization* (MEO) to effectively leverage it. Specifically, based on the binary outcome, we guide entropy optimization by minimizing entropy for successful navigation and maximizing it for failed ones. Here, entropy is derived from a mixture of two distributions: an action distribution, representing the likelihood assigned to each possible action, and a pseudo-expert distribution, a one-hot probability distribution centered on the agent's chosen action, assuming this action as optimal. By combining these two distributions, MEO not only controls the certainty of the decisions but also explicitly suppresses incorrect actions and encourages actions that led to successful navigation.

Additionally, we introduce a novel paradigm of *self-active learning* (SAL), enabling the navigation agent to remain actively engaged throughout all iterations for continuous feedback. Traditional AL methods typically request human feedback only when the model prediction is uncertain, potentially overlooking the subtle errors that are hidden beneath high confidence in certain predictions. In contrast, our method allows the agent to determine the navigation outcome by itself in relatively certain predictions. This is achieved through a self-prediction head, initialized at test time and trained during streaming test episodes using both human-provided labels and the agent's own predicted outcomes. As a result, the agent receives continuous guidance for the direction of entropy optimization, which is crucial for precise adaptation. Ultimately, SAL reduces reliance on human intervention, thereby improving the agent's autonomy and robustness in online test-time environments.

We name our overall framework as ATENA (Active TEst-time Navigation Agent), and validate its effectiveness through comprehensive evaluation on challenging VLN benchmarks: REVERIE [17], R2R [1], and R2R-CE [18]. ATENA achieves significant gains over the underlying target policies and outperforms strong TTA baselines. Our empirical results and in-depth analysis indicate that ATENA effectively addresses test-time distribution shifts and provides a strong foundation for future research on active human-robot interaction in vision-and-language navigation.

The contributions of this work are summarized as follows:

- We introduce ATENA, the first active learning framework for online VLN that leverages human input to guide entropy-based optimization.
- Mixture entropy optimization enhances confidence calibration by explicitly suppressing incorrect actions and encouraging desired actions.
- The self-active learning phase provides a strategic solution to provide continuous active labels, with reduced burden of human labeling in online environment.

2 Related Work

2.1 Vision-Language Navigation

Vision-Language Navigation (VLN) is a pivotal task of bridging human communications with embodied AI system [19, 20, 21]. The sequential natures of the decision making process in VLN led early research to adopt recurrent neural network-based architectures [1, 22, 23]. Following works

utilized the Transformer network [24] to capture complex multimodal dependencies and achieved substantial performance gains [5, 4, 25, 26, 27, 28, 29]. However, these offline training methods merely anticipate domain shifts and suffer a performance degradation when the online navigation environment deviates from the training distribution. To overcome the discrepancy, large-language models came into play as zero-shot navigation agents, but their reasoning capabilities without fine-tuning have yet to yield reliable performances [30, 31, 32, 33, 34, 35, 36]. Recently, another line of research has focused on online test-time adaptation of offline-trained policies, using either unsupervised entropy minimization [37] or feedback-based reinforcement learning [38]. In this work, we explore how the core principle of VLN—*human-robot interaction*—can be effectively leveraged to facilitate online test-time adaptation.

2.2 Entropy-based Test-time Adaptation

Entropy minimization is a widely adopted learning objective in domain adaptation [39, 40, 41] and semi-supervised learning [42, 43, 44]. Recently, entropy minimization has emerged as a foundational technique in test-time adaptation (TTA) due to its simplicity and effectiveness in the absence of labeled target data. [45, 46]. The core idea is to encourage the model to make confident predictions by minimizing the entropy of output distributions at test time, assuming that well-adapted models should be confident on in-distribution samples. Tent [7] introduced a lightweight yet effective approach that minimizes prediction entropy by updating only batch normalization parameters during test time. Building upon this, numerous studies across various fields began integrating entropy minimization into their TTA strategies [47, 48, 10, 11]. In VLN, FSTTA [37] extends entropy minimization by accounting for the sequential and episodic nature of the task. Although effective in many settings, blindly minimizing the entropy can lead to the propagation of overconfident mistakes, making it particularly problematic in sequential tasks like VLN where early errors can cascade [49, 50]. This work addresses the problem by enabling agents to actively query oracles for feedback on uncertain navigation outcomes, providing crucial guidance for entropy-based test-time adaptation.

2.3 Active Learning

Active Learning (AL) is a machine learning strategy designed to efficiently reduce labeling costs by selectively querying labels for the most uncertain or informative data points [12, 51, 52, 13]. Traditional AL methods primarily utilize uncertainty sampling, prioritizing data points when the model's predictive confidence is low; typical metrics for quantifying uncertainty include entropy, margin sampling, and least-confidence measures [14, 15, 16]. Initially focused on relatively simple classification tasks, AL techniques have progressively evolved to tackle increasingly complex, real-world scenarios [53, 54, 55, 56]. Recent advancements further extend AL concepts into TTA, enabling models to dynamically adapt during inference by utilizing uncertainty estimates, thus reducing the reliance on extensive retraining or large amounts of labeled data [57, 58]. Inspired by these developments, our research pioneers the integration of Active Test-Time Adaptation into VLN, effectively overcoming practical constraints in real-world navigation.

3 Method

3.1 Task Description

Vision-Language Navigation (VLN) tasks an agent with interpreting natural language instructions I to navigate through a visual environment. Starting from an initial visual observation o_0, at each timestep t, the agent perceives a visual observation o_t, selects an action a_t according to its policy π_θ, and transitions into the next state. Repeating this process until the agent selects a stopping action produces a trajectory $\tau = \{(o_t, a_t)\}_{t=0}^{T-1}$, where T is the total number of steps taken. In this work, we specifically consider an online VLN scenario where the navigation policy encounters a stream of test instructions and environments during deployment.

3.2 Overview

Our proposed framework, ATENA (Active TEst-time Navigation Agent), enables active learning in online VLN by integrating human guidance into entropy-based optimization. ATENA consists of two core components:

 https://doi.org/10.52202/085713-1490

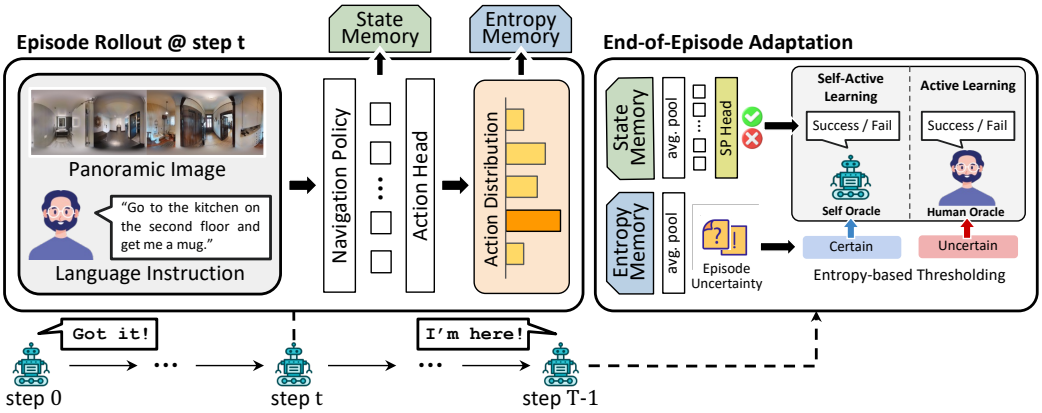

Figure 1: **Overview of the ATENA adaptation framework.** At each navigation step, the agent stores state and entropy information in its memory. Once the episode ends, the stored entropy is used to determine the feedback source: human oracle for uncertain episodes, and self oracle for certain episodes. Self oracle utilizes a self-prediction head, trained during online test-time, enabling the agent to autonomously predict navigation success or failure by itself.

- **Mixture Entropy Optimization (MEO):** A method that refines the agent's policy using outcome-conditioned entropy signals, leveraging a pseudo-expert-guided action distribution to more effectively amplify correct behavior and penalize failure.
- **Self-Active Learning (SAL):** A strategy that enables the agent to autonomously request or replace feedback based on internal uncertainty and self-assessment, allowing robust adaptation even when explicit feedback is sparse or unavailable.

Together, these solutions allow the agent to adapt online by jointly leveraging episodic outcomes and self-predicted performance, without relying on ground-truth trajectories or dense human supervision.

3.3 Mixture Entropy Optimization (MEO)

A traditional entropy minimization method in VLN [37] aim to reduce uncertainty by decreasing the entropy of the predicted action distribution. However, indiscriminately minimizing entropy can reinforce confidence even in incorrect actions, leading to compounding errors during navigation. To address this, we optimize the entropy based on success or failure of the navigation episode. Specifically, we minimize it for successful episodes to reinforce the selected action, and maximize it for failed episodes to penalize incorrect decisions. Furthermore, this entropy-based test-time adaptation is facilitated by our novel mixture entropy optimization.

3.3.1 Mixture Action Distribution

First, we define the *Mixture Action Distribution* as a convex combination of the predicted action distribution π_θ and a pseudo-expert distribution q_{pseudo}. The pseudo-expert distribution is a one-hot probability distribution that assigns full probability (*i.e.* 1.0) to the selected action a_t^{sel}, which refers to the action with the highest predicted probability under the current policy, i.e., $a_t^{\text{sel}} = \arg\max_a \pi_\theta(a \mid o_t, I)$. In other words, the pseudo-expert distribution treats as if a_t^{sel} is the optimal expert action. The mixture action distribution is formalized as:

$$q_{\text{mix}}(a \mid o_t, I) = \lambda q_{\text{pseudo}}(a \mid a_t^{\text{sel}}) + (1-\lambda)\pi_\theta(a \mid o_t, I), \quad 0 \le \lambda \le 1. \tag{1}$$

This mixture formulation sharpens the distribution around the selected action, with the combination weight λ controlling how strongly the pseudo-expert guides the distribution. Accordingly, the entropy of the mixture action distribution at timestep t is defined as:

$$\mathcal{H}(q_{\text{mix}}(\cdot \mid o_t, I)) = -\sum_{a \in \mathcal{A}_t} q_{\text{mix}}(a \mid o_t, I) \log q_{\text{mix}}(a \mid o_t, I), \tag{2}$$

https://doi.org/10.52202/085713-1490

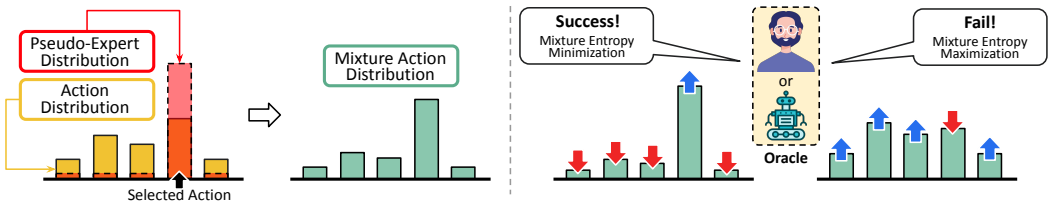

Figure 2: **llustration of Mixture Entropy Optimization (MEO).** (a) The Mixture Action Distribution is constructed by combining the action distribution (yellow) with a pseudo-expert distribution (red). (b) Mixture entropy is minimized for successful episodes to encourage the correct actions, and maximized for failures to penalize incorrect ones.

where $\mathcal{A}_t$ is the set of all possible actions at step t. We average the entropy over all steps and obtain $\mathcal{H}'(q_{\text{mix}})$ as the optimization signal of the episode. Then, the mixture entropy loss function can be formulated as:

$$\mathcal{L}_{\text{mix}} = \mathbb{I}_{\text{success}} \cdot \mathcal{H}'(q_{\text{mix}}) - (1 - \mathbb{I}_{\text{success}}) \cdot \mathcal{H}'(q_{\text{mix}}), \tag{3}$$

where $\mathbb{I}_{\text{success}}$ is a binary indicator that is 1 if the navigation was successful, 0 otherwise.

3.3.2 Effect on Policy Adaptation

Since the mixture action distribution inherently sharpens the original predicted action distribution, this strategy amplifies the feedback signal—further boosting correct actions when successful, and more strongly suppressing incorrect ones when failed (see Figure 2). This is quantitatively evident from the selected action's probability:

$$q_{\text{mix}}(a_t^{\text{sel}} \mid o_t, I) = \lambda + (1 - \lambda)\pi_\theta(a_t^{\text{sel}} \mid o_t, I), \tag{4}$$

which is strictly greater than $\pi_\theta(a_t^{\text{sel}} \mid o_t, I)$ when $\lambda > 0$. As a result, entropy-based optimization applied to q_{mix} exerts stronger influence on the selected action compared to directly using π_θ. Specifically, when minimizing this entropy in successful episodes, the gradient increases $q_{\text{mix}}(a_t^{\text{sel}})$ more sharply than optimizing π_θ alone would. Conversely, maximizing entropy in failed episodes suppresses $q_{\text{mix}}(a_t^{\text{sel}})$ more aggressively, allowing MEO to drive stronger and more directional updates to the policy, improving sample efficiency and reducing the reliance on active learning during test time as demonstrated in Table 4.

3.4 Self-Active Learning (SAL)

Mixture Entropy Optimization enables policy refinement based on navigation outcomes, requiring episodic feedback during test-time adaptation. In practice, however, acquiring feedback—especially from human annotators—can be costly or delayed. Moreover, uncertainty alone may fail to capture subtle but critical navigation errors in seemingly confident predictions. To address these challenges, we propose *Self-Active Learning* (SAL), where the agent selectively queries human feedback or uses its own predictions on navigation outcomes to self-supervise, allowing for more robust and autonomous adaptation.

3.4.1 Uncertainty-Guided Query Strategy

At each timestep, the agent computes the entropy of its action distribution, $\mathcal{H}(\pi_\theta(\cdot \mid o_t, I))$, and stores it in the entropy memory. At the end of an episode, the agent determines the source of supervision $\mathcal{O}$—either Human (human-provided feedback) or Agent (self-generated feedback)—based on the average entropy. Technically, we consider $\mathcal{O}$ as a function of τ to predict $\mathbb{I}_{\text{success}}$:

$$\mathcal{O} = \begin{cases} \text{Human}, & \text{if } \frac{1}{T}\sum_{t=1}^{T} \mathcal{H}(\pi_\theta(\cdot \mid o_t, I)) > \delta \\ \text{Agent}, & \text{otherwise}, \end{cases} \tag{5}$$

where δ is a pre-defined uncertainty threshold. In other words, the agent requests supervision from human in uncertain navigation and self-supervise in relatively certain navigation.

https://doi.org/10.52202/085713-1490

Algorithm 1 Self-Active Learning for Online Adaptation

Require: Pre-trained policy π_θ, entropy threshold δ, learning rate η, loss weight γ

Initialize parameters θ, ϕ
for each episode **do**
  Follow instruction I and collect trajectory $\tau = \{(o_t, a_t)\}_{t=0}^{T-1}$
  Compute average entropy $\bar{\mathcal{H}} = \frac{1}{T}\sum_{t=0}^{T-1} \mathcal{H}(\pi_\theta(\cdot \mid o_t, I))$ (Eq. 5)
  if $\bar{\mathcal{H}} > \delta$ **then**
    Receive human feedback: $\mathbb{I}_{\text{success}} \leftarrow \mathcal{O}_{\text{Human}}(\tau)$
  else
    Receive agent feedback: $\mathbb{I}_{\text{success}} \leftarrow \mathcal{O}_{\text{Agent}}(\tau)$ (Eq. 6)
  Compute mixture entropy loss $\mathcal{L}_{\text{mix}}$ (Eq. 3)
  Compute self-prediction loss $\mathcal{L}_{\text{self}}$ (Eq. 7)
  $\mathcal{L} = \mathcal{L}_{\text{mix}} + \gamma\mathcal{L}_{\text{self}}$ (Eq. 8)
  Update: $(\theta, \phi) \leftarrow (\theta, \phi) - \eta\nabla_{\theta,\phi}\mathcal{L}$
return Adapted policy parameters (θ^*, ϕ^*)

3.4.2 Self-Prediction Head

Predicting Navigation Outcome. To enable autonomous self-supervision, we incorporate a self-prediction head f_ϕ into the pre-trained policy π_θ, trained online to predict the episodic outcome (*i.e.*, success or failure) from its internal states. Specifically, at step t, the D-dimensional hidden state vector $s_t \in \mathbb{R}^D$ is stored in the state memory and averaged over the episode as s_{avg}. This is then fed into the self-prediction head to determine the navigation outcome $\mathbb{I}_{\text{success}}$:

$$\mathbb{I}_{\text{success}} = \begin{cases} 1, & \text{if } \sigma(f_\phi(s_{\text{avg}})) > 0.5 \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where σ is the sigmoid activation function.

Training Self-Prediction Head. To train the self-prediction head, we use a binary cross-entropy between $f_\phi(s_{\text{avg}})$ and the binary episodic outcome $\mathbb{I}_{\text{success}} \in \{0, 1\}$:

$$\mathcal{L}_{\text{self}} = -\Big[\mathbb{I}_{\text{success}} \log(\sigma(f_\phi(s_{\text{avg}})) + (1 - \mathbb{I}_{\text{success}}) \log(1 - \sigma(f_\phi(s_{\text{avg}})))\Big] \tag{7}$$

This loss is used during test-time adaptation regardless of the label source. If the feedback oracle is human, we assume that $\mathbb{I}_{\text{success}}$ is mostly accurate. Alternatively, if the feedback oracle is the agent itself, this can be interpreted as a self-training paradigm with pseudo label derived from the agent's own assessment of task completion, enabling continual self-improvement without external supervision as shown in Table 5. Algorithm 1 summarizes the full adaptation process of SAL.

Total Adaptation Objective of ATENA. We combine the mixture entropy loss from Eq. 3 and the self-prediction loss into a unified test-time adaptation objective:

$$\mathcal{L} = \mathcal{L}_{\text{mix}} + \gamma\mathcal{L}_{\text{self}}, \tag{8}$$

where γ balances the influence of self-assessment. This joint objective reinforces correct decisions, penalizes errors, and improves the agent's ability to assess its own performance during deployment.

4 Experiments

4.1 Datasets & Metrics

We conduct experiments on three challenging VLN benchmarks—REVERIE [17], R2R [1], and R2R-CE [18]. REVERIE evaluates agents' ability to follow high-level, goal-oriented instructions to locate remote objects in indoor environments; a navigation episode is considered successful if the agent stops within 3 meters of the target. For REVERIE, the performance is measured with Success Rate (SR), Oracle Success Rate (OSR), Success penalize by Path Length (SPL) and Remote Grounding SPL (RGSPL). R2R, in contrast, emphasizes fine-grained instruction following, providing detailed step-by-step guidance and using the same 3-meter success criterion. R2R-CE extends R2R by

Table 1: Experimental results on the REVERIE dataset. † implies that the results are obtained from our re-implementation (same for Table 2 and Table 3).

Methods	Val Seen				Val Unseen				Test Unseen			
	OSR ↑	SR ↑	SPL ↑	RGSPL ↑	OSR ↑	SR ↑	SPL ↑	RGSPL ↑	OSR ↑	SR ↑	SPL ↑	RGSPL ↑
HAMT [5]	47.65	43.29	40.19	25.18	36.84	32.95	30.20	17.28	33.41	30.40	26.67	13.08
w/ TENT† [7]	46.03	43.43	40.78	25.81	32.60	30.56	28.23	14.48	25.06	23.73	21.78	10.82
w/ FSTTA† [37]	48.21	42.87	39.56	24.58	36.78	32.89	30.51	17.20	33.39	30.39	26.65	13.61
w/ ATENA (**Ours**)	52.92	57.34	48.08	29.60	38.85	34.00	30.96	17.51	38.19	32.55	28.38	14.32
DUET [4]	73.86	71.75	63.94	51.14	51.07	46.98	33.73	23.03	56.91	52.51	36.06	22.06
w/ TENT	73.72	71.89	64.06	50.41	51.43	47.55	33.99	23.32	57.12	52.61	36.17	22.16
w/ FSTTA	75.59	75.48	65.84	52.23	56.26	54.15	36.41	23.56	58.44	53.40	36.43	22.40
w/ ATENA (**Ours**)	85.52	84.33	74.31	59.99	71.88	68.11	45.82	31.26	57.74	54.28	40.70	25.01
GOAT† [60]	82.36	80.74	73.44	58.82	57.97	53.82	37.52	27.00	61.44	57.72	40.53	26.70
w/ TENT†	82.43	80.74	73.47	58.75	57.68	53.51	37.49	26.99	62.00	57.28	39.82	26.97
w/ FSTTA†	82.36	80.74	73.42	58.82	57.94	53.79	37.50	26.95	62.35	57.52	39.49	26.82
w/ ATENA (**Ours**)	85.03	83.35	76.45	61.60	70.29	67.66	53.15	39.80	64.26	62.03	46.82	31.54

Table 2: Experimental results on the R2R dataset.

Methods	Val Seen				Val Unseen			
	TL ↓	NE ↓	SR ↑	SPL ↑	TL ↓	NE ↓	SR ↑	SPL ↑
DUET [4]	12.33	2.28	79	73	13.94	3.31	72	60
w/ FSTTA [37]	13.39	2.25	79	73	14.64	3.03	75	62
w/ ATENA (**Ours**)	11.27	2.18	80	75	12.31	2.90	75	66
BEVBert [59]	13.56	2.17	81	74	14.55	2.81	75	64
w/ FSTTA†	12.28	2.31	80	75	13.96	2.89	74	63
w/ ATENA (**Ours**)	10.79	2.26	82	78	12.22	2.78	76	68
GOAT† [60]	11.87	1.70	84.52	79.60	13.43	2.33	77.91	67.34
w/ FSTTA†	11.67	1.65	84.92	80.08	13.26	2.32	77.99	67.48
w/ ATENA (**Ours**)	11.66	1.64	85.01	80.13	12.52	2.27	79.01	69.30

Table 3: Experimental results on the R2R-CE dataset.

Methods	Val Seen					Val Unseen				
	TL ↓	NE ↓	OSR ↑	SR ↑	SPL ↑	TL ↓	NE ↓	OSR ↑	SR ↑	SPL ↑
ETPNav [6]	11.78	3.95	72	66	59	11.99	4.71	65	57	49
w/ FSTTA† [37]	11.35	3.93	72	66	59	11.57	4.77	64	57	49
w/ ATENA (**Ours**)	10.81	3.86	72	67	61	12.89	4.53	66	58	49
BEVBert [59]	13.98	3.77	73	68	60	13.27	4.57	67	59	50
w/ FSTTA	14.07	4.11	74	69	60	13.11	4.39	65	60	51
w/ ATENA (**Ours**)	11.31	3.24	75	71	64	13.48	4.50	67	60	51

replacing the discrete action space with a continuous one, increasing the difficulty of low-level control and decision-making. For R2R variants, we use Trajectory Length (TL), Navigation Error (NE), SR and SPL as evaluation metrics.

4.2 Baselines

For experiments, we apply our ATENA on pre-trained HAMT [5], DUET [4] BEVBert [59], ETPNav [6] and GOAT [60]. HAMT is an end-to-end transformer-based VLN network trained via reinforcement learning. DUET exploits both global topology and local visual information for decision-making. BEVBert enhances spatial understanding by encoding the environment into Bird's-Eye-View representation. EPTNav emphasizes long-range planning for agents operating in continuous environments. Lastly, GOAT is a unified structural causal model for VLN. We compare our method against Tent [7] and FSTTA [37]. Tent is a TTA method that minimizes entropy to adjust normalization statistics. FSTTA further applies the concept of entropy minimization to the sequential VLN task. However, due to a reported issue in the official codebase [2], we re-implement the method to ensure accurate evaluation. Throughout our experiments, a † indicates results obtained from our version.

4.3 Main Navigation Results

REVERIE. Table 1 reports the comparisons of the navigation results on the REVERIE dataset, where the TTA methods including ATENA is applied to HAMT [5], DUET [4] and GOAT [19]. Unlike previous methods that utilize entropy minimization as a test-time adaptation signal, we notice a substantial performance increase from ATENA. Specifically, TENT and FSTTA brings minimal performance gains, or rather hinders the navigation performances in several metrics when applied to HAMT and GOAT. However, ATENA improves the SR metric in the validation unseen split up to 3.19%, 44.98% and 25.72% in HAMT, DUET and GOAT, respectively. Furthermore, ATENA also excels in the test unseen split, improving GOAT by 4.59%, 7.47%, 15.52% and 18.13% in OSR, SR, SPL and RGSPL respectively.

R2R & R2R-CE. In Table 2, we present the experimental results on the R2R dataset. Consistent with the findings from the REVERIE dataset, ATENA demonstrates superior effectiveness compared to FSTTA. While FSTTA improves the SPL metric of GOAT by 0.21% on the validation unseen split, ATENA achieves a 2.91% improvement. Moreover, for the SR metric on the validation unseen split of DUET, although the success rate is the same as FSTTA, ATENA achieves this with an 11.69%

[2] https://github.com/Feliciaxyao/ICML2024-FSTTA/issues/1

https://doi.org/10.52202/085713-1490

Table 4: Comparison of Test-Time Adaptation (TTA) methods with Active Learning (AL). Asterisks ($*$) indicate methods integrated with Active Learning (AL), meaning they receive episodic feedback (success or failure) at uncertain navigation to guide entropy minimization or maximization. Active (%) denotes the ratio of navigation steps where feedback is requested.

Methods	Val Seen				Val Unseen			
	SR↑	SPL↑	RGSPL↑	Active (%)	SR↑	SPL↑	RGSPL↑	Active (%)
DUET + TENT*	75.69	67.20	54.67	65.92	55.69	38.76	26.16	90.34
DUET + FSTTA*	71.47	64.19	51.28	89.39	46.95	33.75	23.03	85.25
DUET + MEO* (**Ours**)	**80.53**	**72.84**	**58.78**	**25.44**	**63.70**	**42.49**	**27.83**	**55.52**

reduction in trajectory length, highlighting its high navigation efficiency. We observe similar results in the R2R-CE dataset, which is reported in Table 3. Specifically, ATENA increases 6.7% of SPL for BEVBert in the validation seen split. Lastly, we observe that given that the R2R variants rely on dense, step-wise guidance during training, the episodic feedback is relatively sparse to drive significant performance enhancements compared to that of the REVERIE's.

4.4 Comparison of TTA Methods with Active Learning

Since the compared baseline methods do not employ active learning (AL) in their framework, we integrate AL into the baselines to highlight the impact of MEO. Specifically, we apply TENT and FSTTA to the pre-trained DUET policy, and allow the agent to update the parameters based on the human evaluation of navigation success or failure at uncertain episodes. Similar to ATENA, these baselines also minimize entropy for successful navigation, and maximize for failed ones. Furthermore, we evaluate a variant of ATENA without Self-Active Learning to assess the individual contribution of MEO with AL. For this experiment, the entropy threshold is equally set as $\delta = 0.1$ and we evaluate on the REVERIE dataset. The results are reported in Table 4, from which we draw the following observations. First, compared to the result in Table 1, TENT shows substantial performance increase when guided by human interactions. However, AL provides minimal benefit to FSTTA, which we attribute to its internal mechanism for modifying gradient directions—potentially conflicting with the human-provided guidance on entropy optimization. MEO demonstrates strong synergy with AL, leading to superior navigation performance across SR, SPL, and RGSPL metrics. Moreover, MEO achieves these improvements with significantly fewer human interventions, suggesting that the model progressively gains confidence as navigation proceeds.

4.5 Effect of Self-Active Learning

Table 5 demonstrates the effectiveness of our propose Self-Active Learning (SAL). We compare the performance of ATENA with and without SAL, applied to the DUET navigation policy. The evaluation is done using the REVERIE dataset. Even without SAL, as previously observed in Table 4, ATENA shows a solid performance increase from the base policy. However, enabling the agents to train online from episodic labels and autonomously evaluate its navigation outcome brings notable improvements. Specifically, we observe in the validation unseen split 6.92%, 7.84% and 12.32% enhancements in SR, SPL, and RGSPL, respectively. This clearly demonstrates the effectiveness of remaining continuously active throughout the adaptation process.

Table 5: Comparison of performance on the REVERIE dataset demonstrating the effectiveness of Self-Active Learning (SAL).

Methods	Val Seen			Val Unseen		
	SR↑	SPL↑	RGSPL↑	SR↑	SPL↑	RGSPL↑
w/o SAL	80.53	72.84	58.78	63.70	42.49	27.83
w/ SAL	**84.33**	**74.31**	**59.99**	**68.11**	**45.82**	**31.26**

4.6 Combination Weight of Mixture Entropy Optimization

We vary the combination weight λ in Eq. 1 within $\lambda \in \{0.0, 0.1, 0.2, \ldots, 1.0\}$ and evaluate its effect on adaptation performance. $\lambda = 0.0$ implies that the agent relies solely on its original action distribution, serving as the baseline in Figure 3. We omit $\lambda = 1.0$ from our results because, in this case, the policy relies exclusively on the entropy of the pseudo-expert distribution, which is identically zero and thus provides no informative signal. As the weight of the pseudo-expert distribution increases, we observe consistent improvements across SR, SPL, and RGSPL, demonstrating the

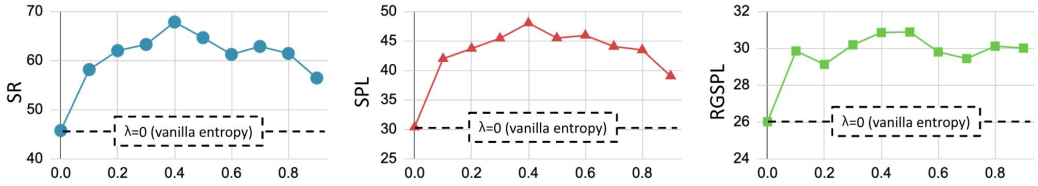

Figure 3: **Effect of the combination weight λ.** Performance comparison with different combination weight λ in Mixture Entropy Optimization. $\lambda = 0$ corresponds to vanilla entropy without distribution mix. The results are averaged across three experiments with different seeds.

benefit of distribution sharpening. The performance peaks at $\lambda = 0.4$, where the balance between the predicted distribution and the pseudo-expert distribution appears to be optimal. As λ increases further, we observe decrease in SR and SPL, yet the benefits compared to vanilla entropy remains solid. These results collectively suggest that while the optimal performance depends on a careful balance between the predicted and pseudo-expert distributions, incorporating the mixture itself is consistently beneficial for adaptation.

4.7 Sampling Strategies for Feedback Episodes

We compare the uncertainty-based active learning strategy with two different sampling baselines: (1) Random Episodes, where episodes that receive feedback are selected randomly; and (2) Consecutive Episodes, where feedback is provided on a contiguous block of episodes starting from the beginning of the dataset. The number of samples of the baselines are equally set to match that of our method's—60% in this experiment—as the portion of uncertainty-based selection cannot be approximated heuristically. In Figure 4, our uncertainty-based sampling outperforms the baselines, indicating that optimization guided by informative uncertainty signals is more effective than relying on simple rule-based selection. Furthermore, when compared to the setting where feedback is provided for all episodes, our method achieves higher performance in SR and RGSPL, while maintaining competitive results in SPL. These results suggest that our uncertainty-based strategy achieves superior performance with reduced supervision by selectively focusing on the most informative episodes, outperforming both heuristic baselines and full feedback in terms of efficiency and effectiveness.

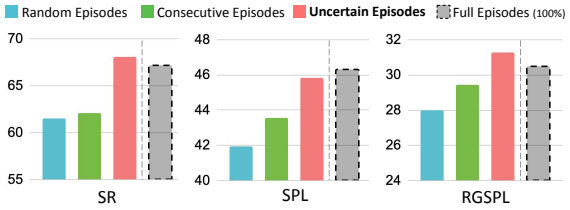

Figure 4: **Different Episode Sampling Strategies.** Our uncertainty-based sampling outperforms baselines and remains competitive against full-feedback settings.

5 Conclusion

We introduce ATENA, a novel TTA framework that leverages active human-robot interaction to enhance online vision-language navigation. Specifically, we propose Mixture Entropy Optimization, explicitly reinforcing correct actions and penalizing incorrect ones based on episodic outcomes. Additionally, through Self-Active Learning, we enable the agent to autonomously predict navigation outcomes during episodes where it has relatively high confidence. Extensive experiments demonstrate that ATENA substantially outperforms baseline approaches, effectively addressing distribution shifts between training and testing environments. By integrating human-guided and self-guided active learning mechanisms, ATENA allows the agent to handle uncertainty through continuous adaptation and self-refinement. Ultimately, our approach opens promising avenues for future research by integrating human-robot interaction with automated self-assessment to support robust and efficient online adaptation across diverse interactive embodied AI tasks.

Limitations and Future Work. A limitation of ATENA is that the policy is updated only after the episode ends, rather than during navigation. This design minimizes external interaction latency but prevents the agent from receiving fine-grained, step-level feedback during execution. As a result, the agent cannot immediately adapt its behavior or identify which specific steps contributed to performance degradation. Future work could explore lightweight mechanisms for incorporating intermediate, segment- or step-level signals to enable more timely and precise adaptation while maintaining low-latency operation.

https://doi.org/10.52202/085713-1490

6 Acknowledgment

This work was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism (International Collaborative Research and Global Talent Development for the Development of Copyright Management and Protection Technologies for Generative AI, RS-2024-00345025, 25%; Research on neural watermark technology for copyright protection of generative AI 3D content, RS-2024-00348469, 25%), the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(RS-2025-00521602, 48%), Institute of Information & communications Technology Planning & Evaluation (IITP) & ITRC(Information Technology Research Center) grant funded by the Korea government(MSIT) (No.RS-2019-II190079, Artificial Intelligence Graduate School Program(Korea University), 1%; IITP-2025-RS-2024-00436857, 1%), and Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT(MSIT, Korea)&Gwangju Metropolitan City.

References

[1] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.

[2] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Javen Qinfeng Shi, and Anton Van den Hengel. Counterfactual vision-and-language navigation: Unravelling the unseen. *Advances in neural information processing systems*, 33:5296–5307, 2020.

[3] Jialu Li, Hao Tan, and Mohit Bansal. Envedit: Environment editing for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15407–15417, 2022.

[4] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16537–16547, 2022.

[5] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Advances in neural information processing systems*, 34:5834–5847, 2021.

[6] 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 Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[7] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[8] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7201–7211, June 2022.

[9] Sachin Goyal, Mingjie Sun, Aditi Raghunathan, and J Zico Kolter. Test time adaptation via conjugate pseudo-labels. *Advances in Neural Information Processing Systems*, 35:6204–6218, 2022.

[10] Hao Yang, Min Wang, Jinshen Jiang, and Yun Zhou. Towards test time adaptation via calibrated entropy minimization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3736–3746, 2024.

[11] Xingzhi Zhou, Zhiliang Tian, Boyang Zhang, Yibo Zhang, Ka Chun Cheung, Simon See, Hao Yang, Yun Zhou, and Nevin L Zhang. Test-time adaptation on noisy data via model-pruning-based filtering and flatness-aware entropy minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10852–10860, 2025.

[12] Burr Settles. Active learning literature survey. 2009.

[13] Dongyuan Li, Zhen Wang, Yankai Chen, Renhe Jiang, Weiping Ding, and Manabu Okumura. A survey on deep active learning: Recent advances and new frontiers. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[14] Jiaxi Wu, Jiaxin Chen, and Di Huang. Entropy-based active learning for object detection with progressive diversity constraint. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9397–9406, 2022.

[15] Alex Tamkin, Dat Nguyen, Salil Deshpande, Jesse Mu, and Noah Goodman. Active learning helps pretrained models learn the intended task. *Advances in Neural Information Processing Systems*, 35:28140–28153, 2022.

[16] Alexandru Tifrea, Jacob Clarysse, and Fanny Yang. Margin-based sampling in high dimensions: When being active is less efficient than staying passive. In *International Conference on Machine Learning*, pages 34222–34262. PMLR, 2023.

[17] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991, 2020.

[18] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 104–120. Springer, 2020.

[19] Peng Gao, Peng Wang, Feng Gao, Fei Wang, and Ruyue Yuan. Vision-language navigation with embodied intelligence: A survey. *arXiv preprint arXiv:2402.14304*, 2024.

[20] Wansen Wu, Tao Chang, Xinmeng Li, Quanjun Yin, and Yue Hu. Vision-language navigation: a survey and taxonomy. *Neural Computing and Applications*, 36(7):3291–3316, 2024.

[21] Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Eric Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. *arXiv preprint arXiv:2203.12667*, 2022.

[22] Dong An, Yuankai Qi, Yan Huang, Qi Wu, Liang Wang, and Tieniu Tan. Neighbor-view enhanced model for vision and language navigation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5101–5109, 2021.

[23] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Vision-language navigation policy learning and adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4205–4216, 2020.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[25] Rui Liu, Wenguan Wang, and Yi Yang. Volumetric environment representation for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16317–16328, 2024.

[26] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13137–13146, 2020.

[27] Liuyi Wang, Zongtao He, Jiagui Tang, Ronghao Dang, Naijia Wang, Chengju Liu, and Qijun Chen. A dual semantic-aware recurrent global-adaptive network for vision-and-language navigation. *arXiv preprint arXiv:2305.03602*, 2023.

[28] Federico Landi, Lorenzo Baraldi, Marcella Cornia, Massimiliano Corsini, and Rita Cucchiara. Multimodal attention networks for low-level vision-and-language navigation. *Computer vision and image understanding*, 210:103255, 2021.

 https://doi.org/10.52202/085713-1490

[29] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah Smith, and Yejin Choi. Robust navigation with language pretraining and stochastic sampling. *arXiv preprint arXiv:1909.02244*, 2019.

[30] Kehan Chen, Dong An, Yan Huang, Rongtao Xu, Yifei Su, Yonggen Ling, Ian Reid, and Liang Wang. Constraint-aware zero-shot vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[31] Yanyuan Qiao, Wenqi Lyu, Hui Wang, Zixu Wang, Zerui Li, Yuan Zhang, Mingkui Tan, and Qi Wu. Open-nav: Exploring zero-shot vision-and-language navigation in continuous environment with open-source llms. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6710–6717. IEEE, 2025.

[32] Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee K Wong. Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation. *arXiv preprint arXiv:2401.07314*, 2024.

[33] Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024.

[34] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7641–7649, 2024.

[35] Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. Discuss before moving: Visual language navigation via multi-expert discussions. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17380–17387. IEEE, 2024.

[36] Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*, pages 260–278. Springer, 2024.

[37] Junyu Gao, Xuan Yao, and Changsheng Xu. Fast-slow test-time adaptation for online vision-and-language navigation. In *International Conference on Machine Learning*, pages 14902–14919. PMLR, 2024.

[38] Sungjune Kim, Gyeongrok Oh, Heeju Ko, Daehyun Ji, Dongwook Lee, Byung-Jun Lee, Sujin Jang, and Sangpil Kim. Test-time adaptation for online vision-language navigation with feedback-based reinforcement learning. In *Forty-second International Conference on Machine Learning*.

[39] Ning Ma, Jiajun Bu, Lixian Lu, Jun Wen, Sheng Zhou, Zhen Zhang, Jingjun Gu, Haifeng Li, and Xifeng Yan. Context-guided entropy minimization for semi-supervised domain adaptation. *Neural Networks*, 154:270–282, 2022.

[40] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2517–2526, 2019.

[41] Xiaofu Wu, Suofei Zhang, Quan Zhou, Zhen Yang, Chunming Zhao, and Longin Jan Latecki. Entropy minimization versus diversity maximization for domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 34(6):2896–2907, 2021.

[42] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004.

[43] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

[44] Jiawei Wu, Haoyi Fan, Xiaoqing Zhang, Shouying Lin, and Zuoyong Li. Semi-supervised semantic segmentation via entropy minimization. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.

[45] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.

[46] Zehao Xiao and Cees GM Snoek. Beyond model adaptation at test time: A survey. *arXiv preprint arXiv:2411.03687*, 2024.

[47] Zhengqing Gao, Xu-Yao Zhang, and Cheng-Lin Liu. Unified entropy optimization for open-set test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23975–23984, 2024.

[48] Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022.

[49] Mingkui Tan, Guohao Chen, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Peilin Zhao, and Shuaicheng Niu. Uncertainty-calibrated test-time model adaptation without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[50] Jonghyun Lee, Dahuin Jung, Saehyung Lee, Junsung Park, Juhyeon Shin, Uiwon Hwang, and Sungroh Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. *arXiv preprint arXiv:2403.07366*, 2024.

[51] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.

[52] Samuel Budd, Emma C Robinson, and Bernhard Kainz. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical image analysis*, 71:102062, 2021.

[53] Martin Mundt, Yongwon Hong, Iuliia Pliushch, and Visvanathan Ramesh. A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning. *Neural Networks*, 160:306–336, 2023.

[54] Ali Ayub and Carter Fendley. Few-shot continual active learning by a robot. *Advances in Neural Information Processing Systems*, 35:30612–30624, 2022.

[55] Jaehyun Park, Dongmin Park, and Jae-Gil Lee. Active learning for continual learning: Keeping the past alive in the present. *arXiv preprint arXiv:2501.14278*, 2025.

[56] Bo Fu, Zhangjie Cao, Jianmin Wang, and Mingsheng Long. Transferable query selection for active domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7272–7281, 2021.

[57] Shurui Gui, Xiner Li, and Shuiwang Ji. Active test-time adaptation: Theoretical analyses and an algorithm. *arXiv preprint arXiv:2404.05094*, 2024.

[58] Guowei Wang and Changxing Ding. Effortless active labeling for long-term test-time adaptation. *arXiv preprint arXiv:2503.14564*, 2025.

[59] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Multimodal map pre-training for language-guided navigation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[60] Liuyi Wang, Zongtao He, Ronghao Dang, Mengjiao Shen, Chengju Liu, and Qijun Chen. Vision-and-language navigation via causal learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13139–13150, 2024.

 https://doi.org/10.52202/085713-1490

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims and contributions of our work are clearly presented in both the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have included discussions of the limitations in the conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Detailed descriptions of our experimental settings, including hardware specifications and hyperparameters, are provided in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

https://doi.org/10.52202/085713-1490

Answer: [No]

Justification: The source code will be publicly released upon acceptance of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Detailed descriptions of the experimental conditions can be found in both the main manuscript and the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our paper does not report error bars or statistical significance information.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Detailed descriptions of the computer resources are provided in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: No potential ethical harms are expected in our research.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have provided in the appendix a discussion of both potential positive and negative societal impacts of our research.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

https://doi.org/10.52202/085713-1490

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No such risks are expected in our paper.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have provided the licenses and credits for the benchmark dataset and the baseline models.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not introduce any new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our research does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our research does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/085713-1490

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Our work does not involve LLMs as important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.