STEP-Nav: Spatial-Temporal Efficient Visual Token Pruning for Vision-and-Language Navigation with Large Language Models

Yantao Lu[1], Shiqi Sun[1*], Ning Liu[2†], Bo Jiang[3], Ying Zhang[1], Jingchao Chen[1], Chenglie Du[1]

[1]Northwestern Polytechnical University
[2]Beijing Innovation Center of Humanoid Robotics Co., Ltd.
[3]Didi Chuxing
yantaolu@nwpu.edu.cn, shiqisun@nwpu.edu.cn, ningliu1220@gmail.com, scottjiangbo@didiglobal.com,
ying_zhang@nwpu.edu.cn, cjc@nwpu.edu.cn, ducl@nwpu.edu.cn

Abstract

Vision-and-Language Navigation (VLN) plays a critical role in tasks of embodied AI, particularly in unseen environments following natural language instructions. Recent advancements leverage large language models (LLMs) to improve the accuracy and generalizability of VLN systems by encoding image sequences as dense token representations. However, this tokenization approach incurs substantial computational overhead due to two key inefficiencies: 1) egocentric camera views often include navigation-irrelevant regions (e.g., sky or distant backgrounds), and 2) high-frame-rate image sequences introduce temporal redundancy. To address these challenges, we propose Spatial-Temporal Efficient Visual Token Pruning (STEP-Nav), a unified framework that simultaneously prunes redundant visual tokens and fine-tunes VLN models to preserve navigation performance. In particular, STEP-Nav incorporates a distance- and content-aware token evaluation mechanism to remove irrelevant tokens at the spatial level, along with temporal level similarity-based filtering to reduce redundancy across sequential frames. To ensure pruning does not harm task performance, we introduce a distortion-aware fine-tuning strategy that aligns pruned-token representations with their full-token counterparts while maintaining navigation accuracy. Experiments on the R2R and RxR benchmarks using Navid-CE and NavGPT-2 as base models demonstrate that STEP-Nav preserves over 95% of the performance while reducing 66.7% of tokens, outperforming existing token pruning baselines.

Introduction

Vision-and-Language Navigation (VLN) (Gu et al. 2022) has emerged as a pivotal capability in embodied AI, enabling agents to navigate complex and unseen environments by following natural language instructions. The dual requirement of multimodal understanding of both human language and visual observations introduces significant challenges for this field. A substantial body of research (Anderson et al. 2018b; Qi et al. 2020; Huang et al. 2022; Ku et al. 2020; Krantz et al. 2021) has been devoted to this problem, which can be broadly classified into two main paradigms: navigation in discrete action spaces and navigation in continuous action spaces. Navigation in discrete environments addresses this problem under simplified conditions. For example, the R2R (Krantz et al. 2020) task in the Matterport3D (MP3D) (Chang et al. 2017) simulator requires agents to make decisions within a discrete action space. In this setting, the environment is modeled as a connected graph, where

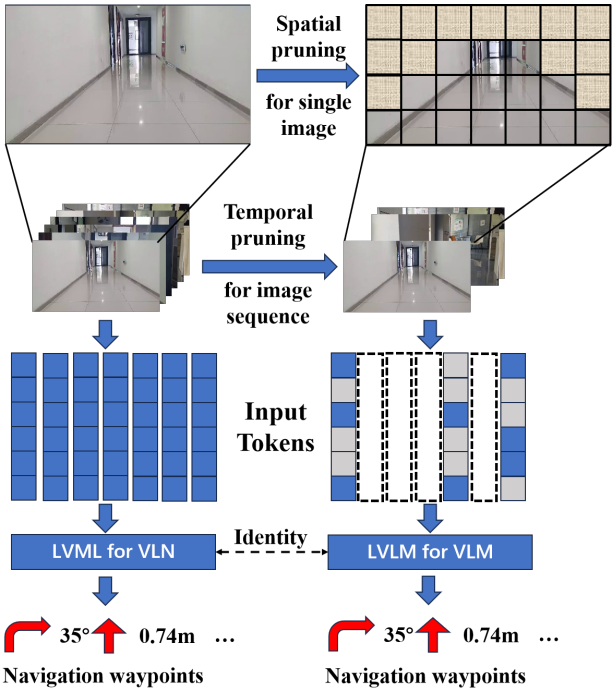

Figure 1: Motivation for STEP-Nav: Ego-centric visual inputs in VLN often contain spatially irrelevant regions (e.g., sky, distant scenery) and temporally redundant frames, resulting in excessive visual tokens that increase computation without improving navigation accuracy. Existing token pruning methods focus on texture-based features and are suboptimal for VLN. STEP-Nav tackles these issues by jointly pruning redundant tokens at both spatial and temporal levels while preserving task-critical information.

*Corresponding Author.
†Corresponding Author.

agents move by selecting from a set of predefined waypoints. Although these methods have evolved rapidly and achieved strong performance, they often fail to capture the continuous and dynamic nature of real-world navigation. To overcome this limitation, navigation in continuous environments (Krantz et al. 2020; Savva et al. 2019; Raychaudhuri et al. 2021; Georgakis et al. 2022) has gained attention, as it enables more realistic modeling to reduce the Sim2Real gap. Examples include R2R-CE and RxR-CE tasks within the Habitat (Savva et al. 2019) simulator. While VLN in continuous environments (VLN-CE) represents a promising step toward bridging the Sim2Real gap, challenges related to generalization in previously unseen environments remain. Recent advancements in VLN leveraging large language models (LLMs) (Zhang et al. 2024b; Zhou, Hong, and Wu 2024; Zhou et al. 2024; Zhang et al. 2024a) have shown considerable potential in addressing these challenges. By integrating extensive pretraining knowledge and sophisticated linguistic understanding, LLM-based approaches exhibit strong generalization capabilities across both unseen environments and Sim2Real scenarios.

Despite these advances, the practical deployment of such models remains constrained by computational inefficiencies, especially in real-time embodied AI agent systems. A key bottleneck arises from the large volume of image tokens generated by high-resolution, high-frequency ego-centric cameras. These images often contain semantically irrelevant regions—such as distant backgrounds or sky—which provide minor utility for navigation but consume significant processing capacity. Moreover, the temporal redundancy across consecutive frames exacerbates the computational overhead, as many visual tokens encode near-identical spatial features.

To mitigate these inefficiencies, token pruning has gained attention as a promising technique for reducing the input token space while preserving task-critical information (Rao et al. 2021; Bolya et al. 2022; Kong et al. 2022). However, existing token pruning methods are devised for generic vision-language models (VLMs) and fail to account for the structure of navigation-related visual data. In particular, two factors hinder the direct application of conventional pruning techniques to VLN: (1) the inability to effectively discard visually complex but navigation-irrelevant content, and (2) the neglect of temporal and spatial redundancies inherent in sequential image data from mobile platforms.

In this work, we introduce **S**patial-**T**emporal **E**fficient Visual Token **P**runing (STEP-Nav), a novel and unified framework for efficient VLN that jointly prunes redundant visual tokens and fine-tunes VLN models to retain navigation performance. Our method integrates a distance- and content-aware token evaluation mechanism to remove uninformative tokens at the single-frame spatial level and applies temporal similarity filtering to reduce redundancy across the temporal level. To further safeguard navigation accuracy under aggressive token pruning, we develop a distortion-aware fine-tuning framework that explicitly adapts the VLN model to sparsified visual inputs. Our approach enforces consistency between the intermediate representations derived from pruned and full-token inputs. By minimizing a feature-level alignment loss across multiple layers, the model learns to preserve critical semantic cues even when large portions of visual tokens are removed.

We validate our approach on the standard VLN benchmarks R2R (Anderson et al. 2018b) and RxR (Ku et al. 2020), using Navid-CE (Zhang et al. 2024b) and NavGPT-2 (Zhou et al. 2024) as base models. Experimental results show that STEP-Nav consistently reduces tokens and computational latency while preserving or improving navigation success rates, outperforming prior token pruning baselines.

The remainder of this paper is organized as follows. The Related Work section reviews related work in VLN, VLM, and token pruning. The Methodology section introduces the proposed STEP-Nav framework, including our Spatial-Temporal token evaluation methods and finetuning optimization strategy. The Experiments section presents experimental results and ablation studies on multiple benchmarks. Finally, the Conclusion section concludes with a discussion of our findings and future research directions.

Our contributions are as follows:

- We identify and address two critical inefficiencies in current VLN pipelines: spatially irrelevant visual content and temporally redundant frame sequences. Then, we propose a spatial-level distance- and content-aware token evaluation mechanism and temporal-level similarity filtering to alleviate the aforementioned issues.
- We design a distortion-aware fine-tuning framework that jointly optimizes token pruning and navigation accuracy by aligning pruned-token representations with full-token representations.
- Extensive experiments on R2R and RxR validate the effectiveness of our method in enhancing inference efficiency with minimal accuracy degradation.

Related Work

Vision-and-Language Navigation

VLN (Anderson et al. 2018b) has become a fundamental benchmark for embodied AI, where an agent must follow natural language instructions to reach a goal location using visual observations. Early works (Fried et al. 2018; Wang et al. 2019) integrated RNN-based encoders and attention mechanisms to process instructions and egocentric images, achieving steady performance gains on benchmarks like R2R and RxR (Ku et al. 2020). More recent advances have incorporated cross-modal transformers (Majumdar, Baral, and Yu 2020; Hong et al. 2021) to enable richer reasoning between vision and language. However, most of these methods process all image tokens from every frame, resulting in substantial computational overhead. Few existing works consider the spatial or temporal redundancy in image sequences, which leads to inefficiencies that hinder deployment in real-time or resource-limited settings.

Large Vision-Language Models for Embodied AI

The success of LLMs such as GPT and LLaMA has inspired research on integrating them with vision encoders for embodied reasoning (Tsai et al. 2023; Driess et al. 2023). These Large VLMs (LVLMs) extract dense visual tokens using Vision Transformers (ViTs) (Dosovitskiy et al. 2021) or

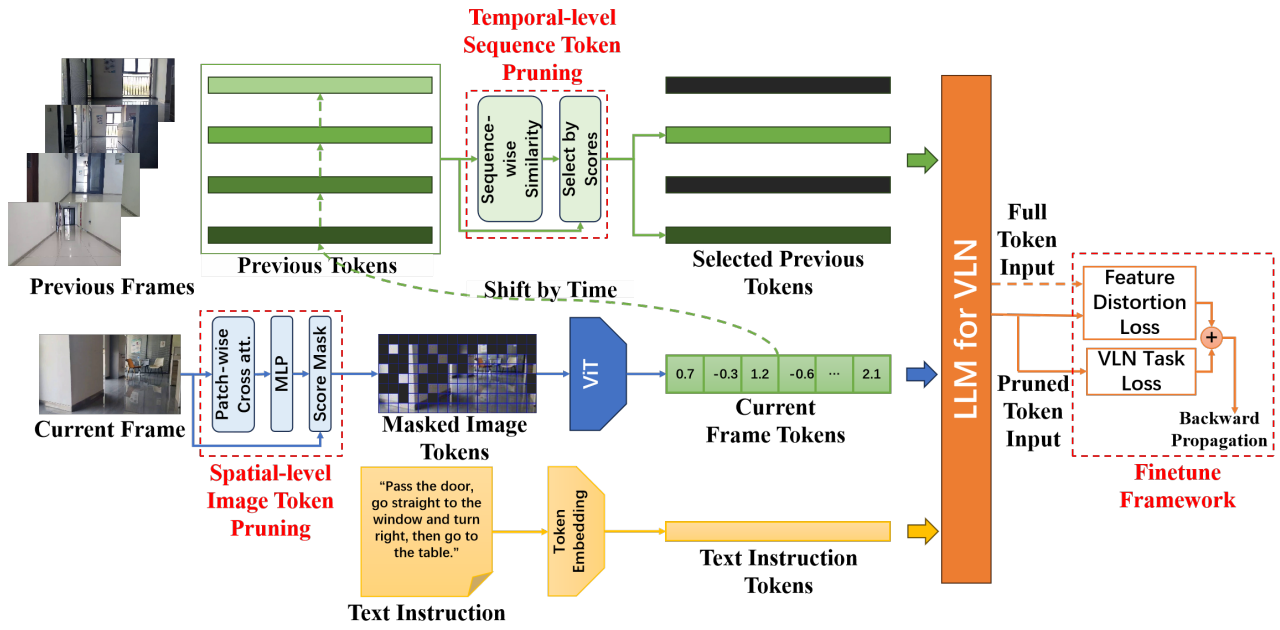

Figure 2: Overview of the proposed STEP-Nav framework. The method combines image-level token pruning, which filters uninformative visual patches using attention and depth cues, with sequence-level pruning that removes temporally redundant frames via cosine similarity. The pruned tokens are fed into the VLN model, and a distortion-aware fine-tuning strategy aligns pruned representations with full-token features to maintain navigation accuracy.

Swin Transformers (Liu et al. 2021) and use frozen or lightly tuned LLMs for trajectory planning, instruction understanding, and interactive feedback. While effective, such models suffer from high computational costs, particularly during the prefill stage of token processing (Fu et al. 2024). In VLN, where the input includes high-frame-rate egocentric sequences, the volume of image tokens rapidly grows and becomes a major bottleneck. To reduce this overhead, token compression or efficient prompting methods have been explored, but few target the VLN setting with spatial and temporal priors in mind.

Token Pruning and Efficiency Optimization

Token pruning is an effective approach to reduce the computational burden of transformer-based models. In computer vision, DynamicViT (Rao et al. 2021) and EViT (Liang, Sun, and Yan 2022) dynamically drop low-importance tokens during inference to speed up ViTs. Other works like ToMe (Bolya et al. 2022) merge similar tokens based on feature proximity, while SPViT (Kong et al. 2022) introduces latency-aware pruning for deployment on real-time systems. In the language domain, LazyLLM (Fu et al. 2024) and IVTP (Huang et al. 2025) prune context tokens in LLMs using attention scores or instruction guidance.

Despite these advances, few methods explicitly consider the unique characteristics of embodied environments. In particular, traditional token pruning approaches often fail to account for: (1) irrelevant but visually salient regions such as sky or distant backgrounds, and (2) the high frame-to-frame similarity in egocentric image streams. Our work builds upon these insights and introduces a VLN-specific token pruning framework that integrates spatial awareness and temporal filtering, enabling efficient inference without sacrificing navigation performance.

Methodology

To address the computational overhead of existing LLM-based VLN models, we propose STEP-Nav, a unified spatial-temporal visual token pruning framework designed for efficiency. Specifically, the tokenization of sequential image observations often introduces substantial computational cost during subsequent LLM inference, necessitating the reduction of redundant tokens. STEP-Nav tackles this challenge through two complementary components: 1) spatial-level token evaluation, which prunes redundant tokens within individual image observations, and 2) temporal-level token evaluation, which further eliminates redundancy across sequences of image tokens. Both modules are subsequently optimized via fine-tuning under a navigation accuracy constraint and are jointly trained using a distortion-aware objective. An overview of the complete architecture is presented in Fig. 2.

Preliminaries

In this work, we formulate VLN tasks as follows: At time step t, the agent receives a natural language instruction $\mathcal{C}$ consisting of l tokens and a sequence of observation $\mathcal{O}_t = \{I_0, \ldots, I_t\}$, representing the sequence of frames observed up to the current step. The goal of the agent is to select a low-level action $\mathbf{a}_{t+1} \in \mathcal{A}$ that will transition it to the next state, yielding a new observation I_{t+1}. The navigation

process can be described as a Partially Observable Markov Decision Process (POMDP), represented by the trajectory $\{I_0, \mathbf{a}_1, I_1, \mathbf{a}_2, \ldots, I_t\}$. In this formulation, the observation space $\mathcal{O}$ consists solely of visual data captured using a monocular RGB camera, without incorporating any additional sensor inputs. The action space $\mathcal{A}$ comprises both categorical action types and their associated continuous parameters (low-level actions, following (Krantz et al. 2020)). This design facilitates a realistic navigation paradigm in which observations rely entirely on visual input and actions are directly executable, closely resembling human navigational behavior.

Spatial-level Image Token Pruning

In VLN, ego-centric image observations often include visually salient yet semantically irrelevant regions, such as distant scenery or the sky. To address this spatial redundancy, we propose a Spatial-level token pruning module based on a distance- and content-aware token evaluation mechanism. Given i-th image frame I_i from the whole T frames sequence, we firstly divide it into a grid of patches and embedding them to image tokens $T_i = \{p_i^{(1)}, p_i^{(2)}, ..., p_i^{(N)}\}$ for vision embedding (e.g., ViT or Swin Transformer), where p_i denotes image token in i-th image frame and N denotes number of image tokens of each frame. For each token $p_i^{(j)}|_{j=1}^{N} \in T_i|_{i=1}^{t}$, we estimate its importance using two cues: (1) its average attention relevance across transformer heads (content-aware), and (2) its estimated spatial proximity to the agent (distance-aware), obtained through geometric projections.

The content relevance score for token $p_i^{(j)}$ is computed as:

$$\gamma^{\text{content}}(p_i^{(j)}) = \frac{1}{N} \sum_{k=1}^{N} A(p_i^{(j)}, p_i^{(k)}) \tag{1}$$

where $A(\cdot, \cdot)$ denotes the attention weights from the self-attention map. In this work, $A(\cdot, \cdot)$ is extracted from the multi-head self-attention (MHSA) layers of standard ViT model (Dosovitskiy et al. 2021).

The distance score is estimated by projecting auxiliary spatial cues onto the image plane. For each token, we compute its average depth and assign a higher priority to closer tokens:

$$\gamma^{\text{distance}}(p_i^{(j)}) = \frac{1}{\bar{d}(p_i^{(j)}) + \epsilon} \tag{2}$$

where $\bar{d}(p_i^{(j)})$ is the average distance for the patch region and ϵ is a small constant for numerical stability. In this work, the distance is directly obtained by the input RGB-D image.

The final image-level importance score for $p_i^{(j)}$ is:

$$\gamma_i^{(j)} = \alpha \cdot \gamma^{\text{content}}(p_i^{(j)}) + (1 - \alpha) \cdot \gamma^{\text{distance}}(p_i^{(j)}) \tag{3}$$

where $\alpha \in [0, 1]$ is a balancing hyperparameter, which is set to 0.5 in this work

Tokens with the lowest scores are pruned to reduce the input length to the VLN model. To preserve structural integrity, we apply diversity-aware filtering that ensures spatial coverage and avoids pruning entire regions.

Temporal-level Sequence Token Pruning

In high-frequency camera streams, consecutive frames often exhibit temporal redundancy due to similar agent motion and scene overlap. Processing all such frames leads to redundant computation and diminished marginal utility. Therefore, we introduce a sequence-level pruning module that filters out repetitive frames before they are tokenized.

For the sequence of observations $\{I_1, I_2, ..., I_t\}$, we firstly extract global descriptors from each image by average pooled patch embeddings. We compute cosine similarity between adjacent frame descriptors and skip frames whose similarity exceeds a threshold τ:

$$\text{Sim}(I_t, I_{t-1}) = \frac{\langle f_t, f_{t-1} \rangle}{\|f_t\| \|f_{t-1}\|} > \tau \Rightarrow \text{drop } I_t \tag{4}$$

To avoid missing critical transitions (e.g., turns or obstacles), we retain keyframes with high predicted motion difference or those aligned with instruction shifts. Optionally, we incorporate odometry-based priors to adjust τ dynamically during fast movements.

This temporal filtering produces a reduced but informative frame set, significantly lowering the total number of visual tokens processed by the VLN model.

Finetune Framework

To ensure that the proposed token pruning modules do not degrade navigation performance, we employ a fine-tuning strategy that aligns the pruned-token representations with those derived from the full token set. Specifically, we fine-tune the base VLN model on the pruned visual inputs while optimizing a composite objective that preserves navigation accuracy.

Let $\mathcal{L}_{\text{nav}}^{\text{full}}$ and $\mathcal{L}_{\text{nav}}^{\text{pruned}}$ denote the standard navigation loss computed using the full token set and the pruned token set, respectively. During fine-tuning, the pruned-token model parameters θ_p are optimized to minimize:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{nav}}^{\text{pruned}} + \lambda \cdot \mathcal{L}_{\text{dist}}, \tag{5}$$

where λ is a weighting coefficient, and $\mathcal{L}_{\text{dist}}$ is a distortion-aware loss that encourages the pruned representation to approximate the full-token representation.

The distortion-aware loss is defined as:

$$\mathcal{L}_{\text{dist}} = \frac{1}{t} \sum_{i=1}^{M} \|g_{\theta_f}(I_i) - g_{\theta_p}(I_i)\|_2^2, \tag{6}$$

where $g_{\theta_f}(\cdot)$ and $g_{\theta_p}(\cdot)$ are the feature embeddings of the VLN encoder with full tokens and pruned tokens, respectively. M is the number of targeted intermediate feature layers. This alignment ensures that pruning does not discard semantically critical information for decision making. To further stabilize training, we initialize θ_p from a pre-trained full-token VLN model and jointly fine-tune the pruning modules and the VLN policy head. The pruning thresholds (for both spatial- and temporal-level pruning) are treated as hyperparameters and updated periodically to balance token reduction and task performance. The detailed procedures of the proposed method are presented in Algorithm 1.

Algorithm 1: Algorithm for the proposed STEP-Nav

Require: Instruction $C = \{c_1, \ldots, c_l\}$; Observations $\mathcal{O}_t = \{I_1, I_2, \ldots, I_t\}$
Ensure: Next action a_{t+1}

Initialize empty sequence of pruned frame embeddings: $\mathcal{F}_{\text{pruned}} \leftarrow \emptyset$
Tokenize I_t into patches $T_t = \{p_1, \ldots, p_N\}$
for all token $p_j \in T_t$ **do**
    Compute attention score $\gamma_{\text{content}}(p_j)$
    Compute depth score $\gamma_{\text{distance}}(p_j)$
    $\gamma_j = \alpha \cdot \gamma_{\text{content}}(p_j) + (1 - \alpha) \cdot \gamma_{\text{distance}}(p_j)$
end for
Select top-K tokens $T_t^{\text{pruned}} \subseteq T_t$
Compute embedding $f_t \leftarrow g_\theta(T_t^{\text{pruned}})$
Compute similarity $\text{Sim}(f_t, f_{t-1})$
if $\text{Sim}(f_t, f_{t-1}) < \tau$ **then**
    $\mathcal{F}_{\text{pruned}} \leftarrow \mathcal{F}_{\text{pruned}} \cup \{f_t\}$
end if
Compute pruned feature representation $f_{\text{pruned}} \leftarrow \text{Concat}(\mathcal{F}_{\text{filtered}})$
Compute full-token feature $f_{\text{full}} \leftarrow g_{\theta_f}(\mathcal{O}_t)$
Compute loss: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{nav}}(f_{\text{pruned}}) + \lambda \cdot \|f_{\text{pruned}} - f_{\text{full}}\|_2^2$
Backpropagate and update θ_p
Predict next action a_{t+1}
return a_{t+1}

Meanwhile, for the training configuration, including data sampling and optimization, we follow the strategy introduced in Navid-CE. To address the limited diversity and realism of available simulation data, Navid-CE adopts a training scheme that augments learning with non-oracle trajectories collected through a DAgger-style rollout (Ross, Gordon, and Bagnell 2011). The agent is first trained on oracle navigation trajectories collected from 61 MP3D scenes, after which additional non-oracle trajectories are generated by deploying the trained agent in VLN-CE environments. The oracle and non-oracle datasets are then combined for final training.

Experimental Results

Experiment Setup

To evaluate the effectiveness of the proposed pruning method, STEP-Nav, we conduct experiments in both simulated environments and real-world onboard settings using data collected in-house. To comprehensively assess VLN performance, we examine both graph-structured environments in the MP3D simulator (Chang et al. 2017) and continuous environments (Krantz et al. 2020) in the Habitat simulator (Savva et al. 2019). For these settings, we adopt various baseline models, including Navid-CE (Zhang et al. 2024b) for continuous environments and NavGPT-2 (Zhou et al. 2024) for graph-structured environments. To further validate the generalizability of STEP-Nav, we evaluate its performance across multiple datasets, including R2R (Krantz et al. 2020) and RxR (Ku et al. 2020). The full details of the experimental setup are provided in the following sections.

Simulated environments. Our experiments are conducted on the VLN benchmarks, which offer photorealistic indoor environments for fine-grained navigation tasks. It is important to note that the MP3D simulator and the Habitat simulator differ in their action spaces: MP3D provides graph-structured, high-level actions in a discrete space, whereas Habitat supports low-level control actions in a continuous space. We evaluate our pruning method under both settings to ensure a comprehensive assessment. Within this framework, we focus on two widely used datasets: R2R and RxR, both implemented in the simulators. To ensure a fair evaluation, all models are trained on the R2R training set, which consists of 10,819 trajectories, and evaluated on the R2R val-unseen split, containing 1,839 trajectories, and the RxR val-unseen split, containing 1,517 trajectories.

Real-world environments. To evaluate the real-world performance of our STEP-Nav, we follow the experimental protocol described in and design a comprehensive evaluation encompassing diverse indoor environments and varying levels of instruction complexity. Experiments are conducted across diverse indoor environments, including a bedroom, meeting room, and office corridor, using the Scout-Mini[1] platform equipped with an Azure Kinect DK camera for capturing synchronized RGB and depth data. For baseline methods that require odometry, the Scout-Mini's integrated Ouster OS0 LiDAR module is employed. Additional implementation details can be found in the supplementary material.

Metrics. We employ commonly adopted evaluation metrics for simulated and real-world settings, following general practices in navigation research (Zhang et al. 2024b; Zhou, Hong, and Wu 2024; Zhou et al. 2024; Krantz et al. 2020; Anderson et al. 2018a). For simulated environments, metrics include the success rate (SR), oracle success rate (OS), success rate weighted by path efficiency (SPL), total trajectory length (TL), and the navigation error to the goal (NE). Note that the sparsity reported in the table indicates the percentage of tokens pruned relative to the original number of tokens. For real-world experiments, we assess performance using only SR and NE, in alignment with the evaluation setup described in Navid-CE.

Baselines Models To assess the effectiveness and generalizability of STEP-Nav, we integrate it into two representative LLM-based VLN models: Navid-CE and NavGPT-2. Navid-CE is a video-based LVLM that leverages a pretrained vision encoder for visual perception and a LLM for reasoning over navigation actions. In contrast, NavGPT-2 is a LLM-based model, that aims to bridge the gap between traditional VLN-specific architectures and LLM-based navigation frameworks. It retains the interpretability of LLMs while aligning visual inputs within a frozen language model, enabling effective integration of visual understanding and policy learning for action prediction and navigational reasoning. Specifically, since NavGPT-2 supports multiple LLM backbones to explore their capabilities, we

[1]Scout-Mini overview: https://global.agilex.ai/products/scout-mini.

Navid-CE (Zhang et al. 2024b)															
Sparsity	66.7%					77.8%					88.9%				
Metric	TL	NE↓	OS↑	SR↑	SPL↑	TL	NE↓	OS↑	SR↑	SPL↑	TL	NE↓	OS↑	SR↑	SPL↑
100% Tokens	7.63	5.47	49.1	37.4	35.9	-	-	-	-	-	-	-	-	-	-
Random	9.71	10.81	25.3	13.9	10.1	11.27	9.21	17.8	9.3	7.1	14.53	10.21	9.2	0.0	0.0
ToMe	8.37	6.98	44.7	34.0	32.9	8.74	7.59	40.8	31.1	27.2	9.34	7.89	36.8	28.3	25.9
VisionZip	8.12	6.37	47.1	35.6	34.6	8.81	6.82	46.7	35.7	33.9	9.53	8.13	45.2	32.6	32.9
STEP-Nav (Ours)	7.68	5.67	47.9	36.4	35.3	7.81	5.82	47.4	36.3	34.7	7.92	6.72	46.2	34.9	33.2
NavGPT-$2_{\text{FlanT5-XL}}$ (1.5B) (Zhou et al. 2024)															
Sparsity	66.7%					77.8%					88.9%				
Metric	TL	NE↓	OS↑	SR↑	SPL↑	TL	NE↓	OS↑	SR↑	SPL↑	TL	NE↓	OS↑	SR↑	SPL↑
100% Tokens	12.81	3.33	78.5	69.9	58.9	-	-	-	-	-	-	-	-	-	-
Random	14.59	4.36	59.2	52.6	44.2	14.34	5.44	56.2	39.4	30.5	14.77	6.05	37.7	25.5	17.5
ToMe	14.52	4.07	71.8	63.6	53.4	14.74	4.52	66.1	58.0	45.3	14.91	5.56	59.2	52.6	44.9
VisionZip	13.32	3.77	75.4	67.1	56.5	13.57	3.92	74.6	66.5	55.7	13.91	4.36	72.1	64.2	54.9
STEP-Nav (Ours)	12.91	3.42	76.7	68.3	57.7	13.14	3.57	76.1	67.6	56.9	13.71	3.92	73.9	65.6	55.5

Table 1: Comparing performance on VLN-CE R2R Val-Unseen for Navid-CE and VLN R2R Val-Unseen for NavGPT-$2_{\text{FlanT5-XL}}$ (1.5B). We list the models pruned by different approaches within 66.7%, 77.8% and 88.9% token pruning ratios.

evaluate our method using both NavGPT-$2_{\text{FlanT5-XL}}$ (1.5B) and NavGPT-$2_{\text{FlanT5-XXL}}$ (5B). This setup allows us to further demonstrate the generalizability of our approach across various LLMs scales. Detailed results are presented in the following section.

Baselines Pruning Methods The "random" baseline follows the approach described in (Yao et al. 2022), where prompt tokens are randomly pruned prior to being processed by the LLM. To provide a more comprehensive comparison, we also evaluate additional pruning strategies that perform token selection and compression at the vision encoder stage. Token Merging (ToMe) (Bolya et al. 2022) is a lightweight method designed to improve the throughput of ViT models without additional training. It iteratively merges visually similar tokens within the transformer using an efficient matching algorithm, offering pruning-level speed while generally achieving higher accuracy. VisionZip (Yang et al. 2025) adopts a two-step procedure: it first identifies tokens with strong attention responses and then merges them according to their similarity, preserving both salient and contextually informative tokens. Both approaches effectively reduce the number of tokens passed to the LLM, significantly lowering computational cost while maintaining competitive performance.

Comparison on Simulated Environment

VLN R2R Results. To evaluate the overall performance of our proposed method, STEP-Nav, we first conduct experiments on the widely adopted VLN-CE R2R val-unseen benchmark dataset. Specifically, all models are trained on the R2R training split, following the settings established in Navid-CE. We implement Navid-CE as the baseline model to compare navigation performance before and after applying STEP-Nav. Compared to the original, unpruned Navid-CE model, our method retains over 95% of the performance while reducing 66.7% of the input tokens, as measured by SPL, NE, OS, and SR. This results in a substantial improvement in computational efficiency. For the token pruning baseline ToMe, which is evaluated under the same experimental settings, our method achieves substantial performance gains on the Navid-CE baseline, increasing the OS from 36.8% to 46.2% and the SPL from 25.9% to 33.2%, with a token pruning ratio of 88.9%. Furthermore, our approach outperforms the current state-of-the-art pruning method VisionZip by 0.3% in SPL and 2.5% in SR, also at a token pruning ratio of 88.9%. To further validate the efficiency of STEP-Nav across different model architectures, we also conduct R2R experiments using NavGPT-2 (Zhou et al. 2024). Consistent with the results observed in Navid-CE, STEP-Nav maintains high performance while significantly reducing the number of input tokens, as shown in Table 1. These results highlight the effectiveness of STEP-Nav, demonstrating its ability to maintain high navigation accuracy while aggressively reducing token input.

VLN RxR Results. To assess the cross-dataset generalization of the pruned models, we follow the training settings of Navid-CE, where models are trained on R2R trajectory-instruction pairs and evaluated in a zero-shot setting on the RxR val-unseen split. The RxR dataset features more detailed instructions and longer navigation trajectories, often referencing diverse and context-rich visual landmarks. This increased linguistic and visual complexity makes it well-suited for assessing the generalization capabilities of navigation models. As shown in Table 2, STEP-Nav not only maintains high computational efficiency but also achieves substantial performance gains over existing methods across multiple metrics, including NE, OS, SR, and SPL. These findings highlight the generalization strength of our pruning strategy, demonstrating its effectiveness in accelerating inference within LLM-based VLN frameworks through efficient token reduction.

Comparison with Different LLM Scales. To assess the efficiency of the proposed pruning method STEP-Nav, we evaluate its performance across language models of varying scales. Specifically, we build upon the NavGPT-2 framework, applying a fixed token pruning ratio of 85%. NavGPT-

Baseline Model	Navid-CE (Zhang et al. 2024b)														
Sparsity	66.7%					77.8%					88.9%				
Metric	TL	NE↓	OS↑	SR↑	SPL↑	TL	NE↓	OS↑	SR↑	SPL↑	TL	NE↓	OS↑	SR↑	SPL↑
100% Tokens	10.59	8.41	34.5	23.8	21.2	-	-	-	-	-	-	-	-	-	-
Random	12.51	9.89	25.2	15.3	11.7	12.59	9.73	21.1	13.2	7.2	12.89	10.03	15.4	6.2	4.3
ToMe	11.01	8.89	30.0	20.7	18.4	11.33	9.08	27.7	19.4	17.4	11.47	9.31	25.5	17.4	15.5
VisionZip	10.65	8.43	33.5	23.2	20.5	10.95	8.78	32.8	22.7	19.9	11.32	9.11	31.6	21.8	19.3
STEP-Nav (Ours)	10.67	8.45	33.8	23.3	20.8	10.81	8.63	33.2	22.8	20.3	11.01	8.77	32.7	22.4	20.0

Table 2: Comparing performance on VLN-CE RxR Val-Unseen. We list the models pruned by different approaches within 66.7%, 77.8% and 88.9% token pruning ratios. The baseline model is Navid-CE.

Metric	TL	**NE↓**	**OS↑**	**SR↑**	**SPL↑**
NavGPT-$2_{\text{FlanT5-XL}}$ (1.5B)	12.81	3.33	78.5	69.9	58.9
Random	14.68	5.80	44.2	32.7	23.4
ToMe	14.80	5.05	63.3	54.2	44.7
VisionZip	13.76	4.19	73.2	64.5	55.1
STEP-Nav (Ours)	13.52	3.71	74.9	66.3	56.0
NavGPT-$2_{\text{FlanT5-XXL}}$ (5B)	14.04	2.98	83.9	73.8	61.1
Random	15.53	5.37	48.2	36.5	26.4
ToMe	14.90	4.98	69.7	57.9	47.6
VisionZip	14.26	4.27	78.1	70.1	59.1
STEP-Nav (Ours)	14.09	3.44	79.7	69.6	58.4

Table 3: Comparison of model performance across different LLM scales, including NavGPT-$2_{\text{FlanT5-XL}}$ (1.5B) and NavGPT-$2_{\text{FlanT5-XXL}}$ (5B), on the R2R val-unseen split. Models pruned using different approaches with an 85% token pruning ratio are evaluated.

2, which is based on InstructBLIP (Dai et al. 2023), supports multiple variants of LLMs, making it suitable for evaluating our approach across different model scales. We select two representative configurations: NavGPT-$2_{\text{FlanT5-XL}}$ (1.5B) and NavGPT-$2_{\text{FlanT5-XXL}}$ (5B), using LLMs FlanT5-XL (3B) and FlanT5-XXL (11B), respectively. All models utilize the same vision encoder (ViT-g/14 (Fang et al. 2023)), and both the vision encoder and LLM components remain frozen throughout training. As shown in Table 3, STEP-Nav achieves SOTA performance, surpassing existing pruning methods and demonstrating consistent effectiveness across LLMs of varying scales. Specifically, our method achieves notable performance improvements, retaining a SR of 66.3% (original: 69.9%) and a SPL of 56.0% (original: 58.9%) for the 1.5B model. For the 5B model, it retains an SR of 69.6% (original: 73%) and an SPL of 58.4% (original: 61.1%), surpassing the performance of the current state-of-the-art method, VisionZip.

Comparison on Real-world Environment

To further evaluate the generalizability of the proposed method in more challenging scenarios, we conduct extensive experiments in real-world environments. In this setting, we use Navid-CE as the baseline with a token pruning ratio of 77.8% and evaluate performance across three challenging indoor environments: a meeting room, a bedroom, and an office corridor. As shown in Table 4, our method retains over 95% of the original performance while pruning 77.8% of the tokens, demonstrating that it effectively preserves the sim-to-real transfer capability of the models.

	Meeting Room		**Bedroom**		**Office Corridor**	
Method	SR↑	NE↓	SR↑	NE↓	SR↑	NE↓
Navid-CE	84%	1.43	73%	2.10	95%	0.89
Random	52%	3.25	38%	3.57	51%	2.04
ToMe	69%	2.83	60%	2.44	78%	1.97
VisionZip	78%	1.79	67%	2.35	84%	1.18
STEP-Nav (Ours)	80%	1.64	69%	2.07	91%	0.99

Table 4: Comparison of model performance across three diverse real-world environments (Meeting Room, Bedroom, and Office Corridor). Models pruned using different approaches at a 77.8% token pruning ratio are evaluated. The baseline model is Navid-CE, and the metrics reported are SR and NE, which are standard metrics for VLN.

Discussion and Conclusion

In this paper, we explore computational cost reduction in LLM-based VLN models, which, while offering strong generalization capabilities through LLMs, introduce substantial computational overhead. A spatial-temporal efficient visual token pruning framework STEP-Nav is introduced to address redundancy in these models. Specifically, STEP-Nav independently evaluates token redundancy at both the spatial and temporal levels, effectively reduces redundant tokens, and introduces a fine-tuning strategy to preserve overall model performance. Through extensive evaluations, STEP-Nav preserves over 95% of the original performance while reducing 66.7% of the input tokens. This substantial reduction significantly enhances computational efficiency and outperforms the established baselines on VLN-CE tasks.

In future work, we plan to extend STEP-Nav by addressing the textual branch to further improve its efficiency. For instance, consider the instruction: "Walk through the doorway towards the tub, turn left and walk towards the bar, right before the bar turn left, walk up the stairs and walk through the first door on the right, stop right before the table." Such instructions often contain substantial contextual information, including both relevant objects and positional descriptions as well as redundant action details. Additionally, once agents have passed the objects mentioned in the instruction, tokens referring to those objects could also be pruned, leading to further gains in both performance and efficiency. Therefore, we aim to explore the relationship between textual and visual inputs to design a more effective token selection strategy for VLN tasks in future work.

References

Anderson, P.; Chang, A.; Chaplot, D. S.; Dosovitskiy, A.; Gupta, S.; Koltun, V.; Kosecka, J.; Malik, J.; Mottaghi, R.; Savva, M.; et al. 2018a. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*.

Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Gould, S.; and van den Hengel, A. 2018b. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3674–3683.

Bolya, D.; Fu, C.-Y.; Dai, X.; Zhang, P.; Feichtenhofer, C.; and Hoffman, J. 2022. Token merging: Your ViT but faster. *arXiv preprint arXiv:2210.09461*.

Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Niessner, M.; Savva, M.; Song, S.; Zeng, A.; and Zhang, Y. 2017. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*.

Dai, W.; Li, J.; Li, D.; Tiong, A.; Zhao, J.; Wang, W.; Li, B.; Fung, P. N.; and Hoi, S. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36: 49250–49267.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; and et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Driess, D.; Xia, F.; Clarke, C.; and et al. 2023. PaLM-E: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.

Fang, Y.; Wang, W.; Xie, B.; Sun, Q.; Wu, L.; Wang, X.; Huang, T.; Wang, X.; and Cao, Y. 2023. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19358–19369.

Fried, D.; Hu, R.; Cirik, V.; Rohrbach, A.; Andreas, J.; Morency, L.-P.; Berg-Kirkpatrick, T.; Klein, D.; Darrell, T.; and Saenko, K. 2018. Speaker-follower models for vision-and-language navigation. In *Advances in neural information processing systems*, 3314–3325.

Fu, Q.; Cho, M.; Merth, T.; Mehta, S.; Rastegari, M.; and Najibi, M. 2024. LazyLLM: Dynamic token pruning for efficient long context LLM inference. *arXiv preprint arXiv:2407.14057*.

Georgakis, G.; Schmeckpeper, K.; Wanchoo, K.; Dan, S.; Miltsakaki, E.; Roth, D.; and Daniilidis, K. 2022. Cross-modal map learning for vision and language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15460–15470.

Gu, J.; Stefani, E.; Wu, Q.; Thomason, J.; and Wang, X. E. 2022. Vision-and-language navigation: A survey of tasks, methods, and future directions. *arXiv preprint arXiv:2203.12667*.

Hong, Y.; Chen, Z.; Zhou, Y.; Chen, Q.; Chen, Y. C.; Wang, Z.; and Tao, D. 2021. VLN-BERT: A recurrent vision-and-language BERT for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1643–1653.

Huang, C.; Mees, O.; Zeng, A.; and Burgard, W. 2022. Visual language maps for robot navigation. *arXiv preprint arXiv:2210.05714*.

Huang, K.; Zou, H.; Xi, Y.; Wang, B.; Xie, Z.; and Yu, L. 2025. IVTP: Instruction-guided Visual Token Pruning for Large Vision-Language Models. In *European Conference on Computer Vision*.

Kong, Z.; Dong, P.; Ma, X.; Meng, X.; Niu, W.; Sun, M.; and Wang, H. 2022. SPViT: Enabling faster vision transformers via latency-aware soft token pruning. In *European Conference on Computer Vision*, 620–640.

Krantz, J.; Gokaslan, A.; Batra, D.; Lee, S.; and Maksymets, O. 2021. Waypoint models for instruction-guided navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15162–15171.

Krantz, J.; Wijmans, E.; Majumdar, A.; Batra, D.; and Lee, S. 2020. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, 104–120. Springer.

Ku, A.; Anderson, P.; Patel, R.; Ie, E.; and Baldridge, J. 2020. Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 4392–4412.

Liang, H.; Sun, J.; and Yan, L. 2022. Not all tokens are equal: Adaptive token sampling for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12345–12354.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Majumdar, A.; Baral, C.; and Yu, M. 2020. Improving vision-and-language navigation with image-text pairs from the web. In *European Conference on Computer Vision*, 259–276. Springer.

Qi, Y.; Pan, Z.; Zhang, S.; van den Hengel, A.; and Wu, Q. 2020. Object-and-action aware model for visual language navigation. In *European conference on computer vision*, 303–317. Springer.

Rao, Y.; Zhao, W.; Tang, Y.; Zhou, J.; and Lu, J. 2021. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, 13960–13971.

Raychaudhuri, S.; Wani, S.; Patel, S.; Jain, U.; and Chang, A. X. 2021. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. *arXiv preprint arXiv:2109.15207*.

Ross, S.; Gordon, G.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 627–635. JMLR Workshop and Conference Proceedings.

Savva, M.; Kadian, A.; Maksymets, O.; Zhao, Y.; Wijmans, E.; Jain, B.; Straub, J.; Liu, J.; Koltun, V.; Malik, J.; et al. 2019. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9339–9347.

Tsai, Y.-L.; et al. 2023. Navi: A benchmark for grounding navigation instructions in realistic environments. *arXiv preprint arXiv:2304.04756*.

Wang, X.; Wu, Q.; Chen, Q.; Wang, L. L.; and Gao, J. 2019. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6629–6638.

Yang, S.; Chen, Y.; Tian, Z.; Wang, C.; Li, J.; Yu, B.; and Jia, J. 2025. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 19792–19802.

Yao, Z.; Wu, X.; Li, C.; Holmes, C.; Zhang, M.; Li, C.; and He, Y. 2022. Random-ltd: Random and layerwise token dropping brings efficient training for large-scale transformers. *arXiv preprint arXiv:2211.11586*.

Zhang, J.; Wang, K.; Wang, S.; Li, M.; Liu, H.; Wei, S.; Wang, Z.; Zhang, Z.; and Wang, H. 2024a. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*.

Zhang, J.; Wang, K.; Xu, R.; Zhou, G.; Hong, Y.; Fang, X.; Wu, Q.; Zhang, Z.; and Wang, H. 2024b. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*.

Zhou, G.; Hong, Y.; Wang, Z.; Wang, X. E.; and Wu, Q. 2024. Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*, 260–278. Springer.

Zhou, G.; Hong, Y.; and Wu, Q. 2024. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 7641–7649.