Open Peer Review on Qeios

RESEARCH ARTICLE

Fine-Grained Alignment in Vision-and-Language Navigation through Bayesian Optimization

Yuhang Song[1,2], Mario Gianni[1], Chenguang Yang[1], Anh Nguyen[1], Chun-Yi Lee[3], Te-Chuan Chiu[2], Kunyang Lin[4]

1 Department of Computer Science, University of Liverpool, United Kingdom
2 Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
3 Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, Province of China
4 School of Software Engineering, South China University of Technology, China

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.

Abstract

This paper addresses the challenge of fine-grained alignment in Vision-and-Language Navigation (VLN) tasks, where robots navigate realistic 3D environments based on natural language instructions. Current approaches use contrastive learning to align language with visual trajectory sequences. Nevertheless, they encounter difficulties with fine-grained vision negatives. To enhance cross-modal embeddings, we introduce a novel Bayesian Optimization-based adversarial optimization framework for creating fine-grained contrastive vision samples. To validate the proposed methodology, we conduct a series of experiments to assess the effectiveness of the enriched embeddings on fine-grained vision negatives. We conduct experiments on two common VLN benchmarks R2R and REVERIE, experiments on the them demonstrate that these embeddings benefit navigation, and can lead to a promising performance enhancement. Our source code and trained models are available at: https://anonymous.4open.science/r/FGVLN.

Corresponding authors: Yuhang Song, sgysong10@liverpool.ac.uk

1. Introduction

In recent years, Transformer[1] based architectures have revolutionized the processing and comprehension of instruction and path in Vision-and-Language Navigation (VLN) task[2][3][4][5]. For example, VLNBERT[6], aligning the instruction and path by bringing the embeddings of positive Path-Instruction (PI) pairs closer while pushing those of negative pairs apart. Prior studies conducted by[6][7][8] highlight the importance of better encoding in VLN and suggest that better-aligned embeddings generally result in improved representations of both the navigation instructions and the corresponding path sequences, which can, in turn, enhance overall VLN task performance. The majority of these methods improve the learned embeddings by pre-training on external augmented data, while limited attention has been given to enhancing learned embeddings by improving the quality of contrastive samples. Nonetheless, research in the domain of contrastive learning indicates that sampling negative examples can significantly impact the learned embeddings. More specifically,

sampling hard negative examples can potentially enhance the quality of these embeddings[9][10][11], which suggests room for further enhancement in VLN tasks.

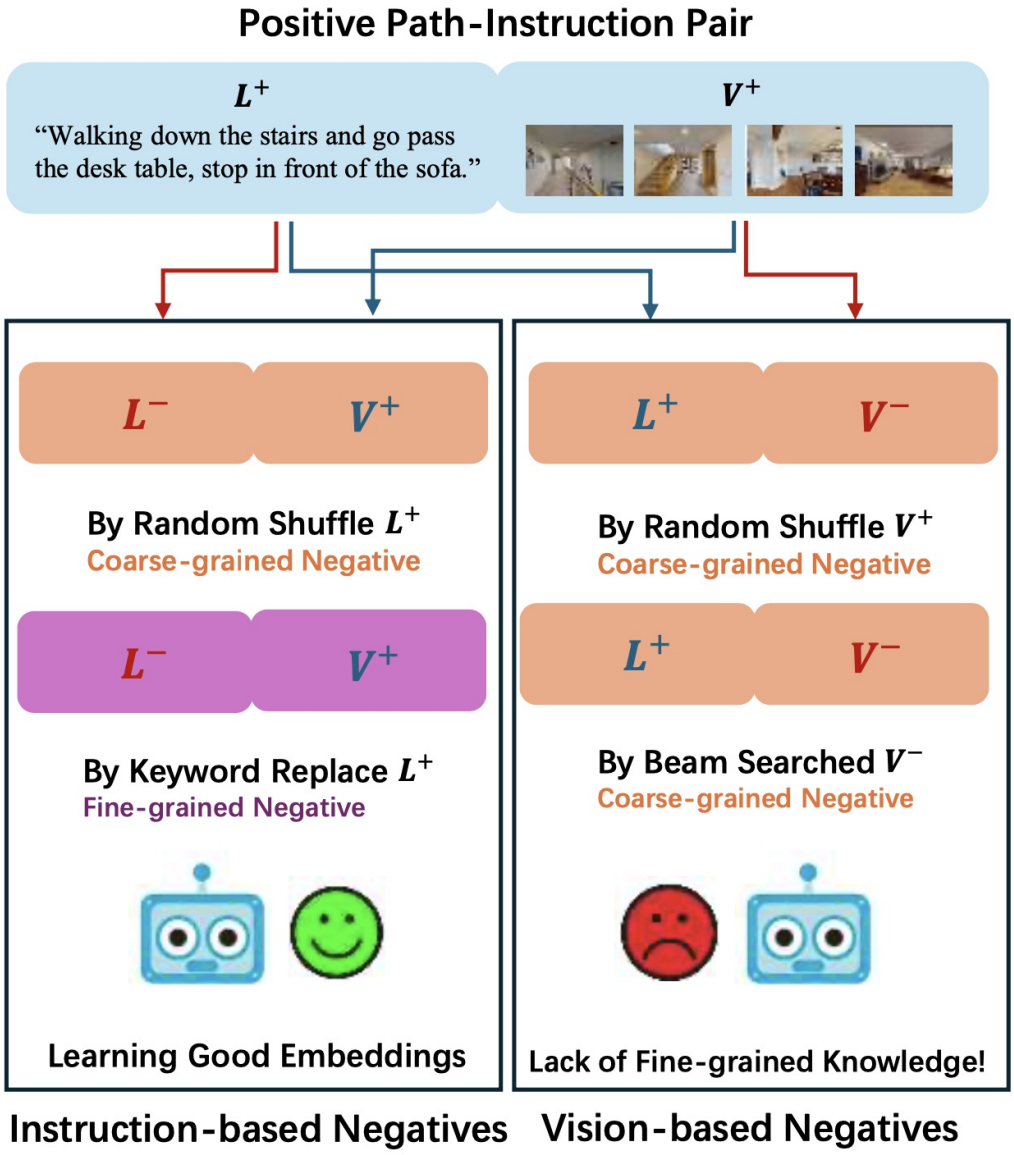

Figure 1. An illustration of existing strategies for generating instruction-based and vision-based path-instruction (PI) pairs, where only coarse-grained negative examples are generated and utilized for vision-based PI samples. *L* and *V* denote the instruction and path, + represents the positive samples, while − denotes the negative samples.

Current VLN approaches[6][7][8] generate negative PI pairs from positive PI pairs by either: (1) altering the positive instruction to generate **instruction negative** PI pairs or (2) altering the positive path to generate**vision negative** PI pairs. A common method for these alterations involves randomly shuffling the instruction or path sequences. To further diversify the styles of negative samples and enhance the learned embeddings, previous studies have explored alternative methods for sampling additional negative pairs. AirBert[7] attempted to create additional instruction negative samples using a keyword replacement method proposed by[12]. These pairs are fine-grained language-based negatives that differ

from the positive PI pair in instructions with only minor lexical variations, which has been demonstrated to significantly benefit the model in training. This finding emphasizes the importance of fine-grained samples. On the other hand, for the vision negative PI pairs, the authors in[6] employed beam search to collect additional candidate paths for each instruction with a greedy instruction follower model[13]. Paths that fail to accomplish the instruction are also considered vision negative PI pairs. Unfortunately, unlike the fine-grained instruction negatives, paths in both random shuffled and beam-searched vision PI negative pairs significantly differ from the positive path. These vision negative PI pairs can be considered coarse-grained negatives. Fig. 1 illustrates the sampling methods of negative PI pairs in contemporary approaches, where only coarse-grained vision negatives are involved.

Generating effective fine-grained vision PI negative pairs can be challenging, particularly when determining the appropriate key elements to replace in the vision sequence. Considering the aforementioned challenges and the need to address the difficulty in identifying the most impactful fine-grained negatives for vision sequences, we propose to utilize Bayesian Optimization (BO). BO-based methods are well-regarded for their efficiency in exploring search spaces, which is critical in our context for pinpointing vision negatives. Our proposal draws inspiration from[14], which employs adversarial examples to identify the weaknesses of a model. Building on this concept, our framework is designed to generate vision fine-grained negative pairs that refine the model's vision-language alignment capabilities. Our BO-based framework iteratively locates the frames that would most significantly impact the model's predictions. Replacing these frames to form fine-grained vision negatives in training facilitates VLN tasks and results in a tailored training set that includes a balanced mix of coarse negatives, and fine-grained negatives. To sum up, we propose a Fine-grained VLN (FGVLN) framework that involves a strategic Bayesian-based optimization via adversarial training to integrate BO into our training process. To validate our framework we evaluate the resulting learned vision embeddings. Our findings reveal that the encoder trained with our framework captures more fine-grained visual information. We further perform experiments on the common VLN discriminative benchmark Room-to-Room (R2R)[2], and adapt our trained backbone into two benchmarks R2R and REVERIE[15] in generative setting. The results validate the effectiveness of the fine-grained embeddings learned with our method in enhancing performance in both settings. We further provide an ablation study to validate the BO design choice. Our contributions are summarized as follows:

- We highlight the importance of fine-grained samples for VLN and emphasize that coarse-grained cross-modal features learned by the encoders result in less accurate PI alignments.
- We find that our method results in encoders with uniform attentions across sequences, capturing better fine-grained details, which allows the model to form complex decision boundaries.
- We incorporate the encoders with enhanced embeddings obtained from our method to the VLN tasks and improve the performance in both discriminative and generative settings.

II. Related Work

VLN[2] has garnered attention, with a range of follow-up studies in recent years[16][17][18][19][20][15][21][22][23][24]. VLN tasks include discriminative and generative settings, described as follows.

Discriminative Vision-and-Language Navigation.

Discriminative navigation considers the navigation problem as a path selection task. In this setup, the agent is tasked with choosing the most appropriate path from a set of candidates based on a given instruction[6][7][8][25][26][27][28][29][30][31][32]. The study in[6] first pre-trained the agent on web image-caption datasets. Nevertheless, alignment issues persisted due to the out-of-domain nature of the web image-caption datasets, which are not consistent with downstream tasks. This challenge was tackled by Airbert[7], which used in-domain Airbnb image-caption pairs for more realistic PI sample generation, supplemented by tasks such as masked language modeling[33]. Further advancements were made by Lily[8], a technique that incorporated indoor YouTube video data to enhance the alignment more closely with actual navigation tasks. Although these methods were effective, existing approaches primarily focused on improving the learned embeddings by data augmentation. In contrast, our work diverges from these traditional methods by investigating the impact of fine-grained vision negatives on the embeddings, and proposes a BO-based method to produce fine-grained vision negatives, which enables the encoding of more fine-grained path information.

Generative Vision-and-Language Navigation.

In this setting, the agent's goal is to predict the action distribution given navigation instructions and observations. Some prior methods predicted actions using sequential models[2][13][26]. To capture cross-modal relationships, methods based on the Transformer architecture[1] have been proposed and adapted for agent training[34], with some of them also leveraging Vision-Language pre-training[28][30][35][36][37][38][39]. Inspired by BERT[40], several works proposed to use different variants of BERT[40] for large-scale visio-linguistic pretraining[33][6][7][8][41]. Among them, ViLBERT[33] has been widely adopted and proven effective. Our work thus uses ViLBERT as the backbone. We adapt our trained encoders into[41] to show that fine-grained vision negatives can improve performance in the generative setup.

III. Preliminaries

Following[6], to train ViLBERT[33] encoders, we formulate the VLN task as a path selection problem, where the navigation task involves identifying the path that best aligns with the given instructions. Given a set of candidate paths V and an instruction L, the problem of VLN is defined as finding a trajectory v^* such that:

$$v^* = \underset{v_i \in V}{\operatorname{argmax}} F_c(v_i, L), \quad F_c(v_i, L) = f_\theta(h_{v_i} \odot h_L) = s_i,$$

where F_c is a compatibility function that assesses whether a given trajectory follows the instruction and stops near the intended goal, which produces a compatibility score s_i. h_{v_i} is the embedded representation of the trajectory, and h_L is the embedded instruction, both encoded by encoders parameterized by ϕ. $f_\theta(\cdot)$ denotes learned transformations parameterized by θ, which maps the embedding into a s_i of a given trajectory v_i with respect to L. $\odot$ denotes a dot product operation.

According to the formulation in[33], VLN tasks can separately encode visual navigation trajectory patches and language

sequence tokens using two distinct Transformers. Assume a visual navigation trajectory $v = (\tau_1, \tau_2, \ldots, \tau_K) \in \mathrm{R}^{K \times W \times H \times C}$, where K denotes the trajectory length (i.e., number of frames τ), and W, H, C represent the frame dimensions. To align with ViLBERT, the visual trajectory is reshaped such that each frame comprises P visual region patch nodes x_p^k, with $k \in K$ and $p \in P$. The trajectory input is thus represented as $X_v = [[\mathrm{IMG}], x_1^1, \ldots, x_2^1, \ldots, [\mathrm{IMG}], x_1^K, \ldots, x_P^K]$. Similarly, given a language instruction sequence $L = (l_1, l_2, \ldots, l_T) \in \mathrm{R}^{T \times D}$, where T is the number of tokens and D is the token dimension, the tokenized text input to the model can be represented as: $X_L = [[\mathrm{CLS}], x_1, \ldots, x_T, \ldots, [\mathrm{SEP}]]$, where [IMG], [[CLS], [SEP]] are special tokens. Based on the above formulation, an aligned positive Trajectory-Instruction pair can be expressed as $X^+ = (X_v^+, X_L^+)$, and the generated negative pair as $X^- = (X_v^-, X_L^-)$. The output embedding at the location of the first [IMG] and the [CLS] is taken as the output of the model for trajectory and instructions, respectively, which can then be utilized for the two embeddings h_{v_i} and h_L in Eq. (1).

To concentrate on the contrastive learning aspect, in this work, the pre-training stage of Lily[8] is kept unchanged, and the VLN model is fine-tuned in the downstream path ranking (PR) task using a Bayesian-based optimization framework. PR aims to minimize a contrastive loss given a positive pair and several negative pairs $\mathrm{L}_{PR}(X^+, \{X^-\}^N)$, where N generated negative pairs have either a different trajectory or a different instruction. The negative pairs can be expressed as $X^- = \{(X_v^-, X_L^+)\}$ or $X^- = \{(X_v^+, X_L^-)\}$. The PR loss L_{PR} can then be formulated as follows:

$$\min_{\theta, \phi} \mathrm{L}_{PR}(X^+, \{X^-\}^N) = -\log \frac{\exp(f_\theta(X^+))}{\exp(f_\theta(X^+)) + \sum_N \exp(f_\theta(X^-))},$$

where $f_\theta(\cdot)$ denotes the learned transformations on the outputs of the backbone encoders as in Eq. (1). The objective is to minimize L_{PR} with respect to model parameters.

IV. Methodology

In this section, we present in Section IV-A of the proposed FGVLN framwork. Section IV-B elaborates on an encoder synchronization and optimization strategy.

A. *Bayesian-based Optimization by Adversarial Training*

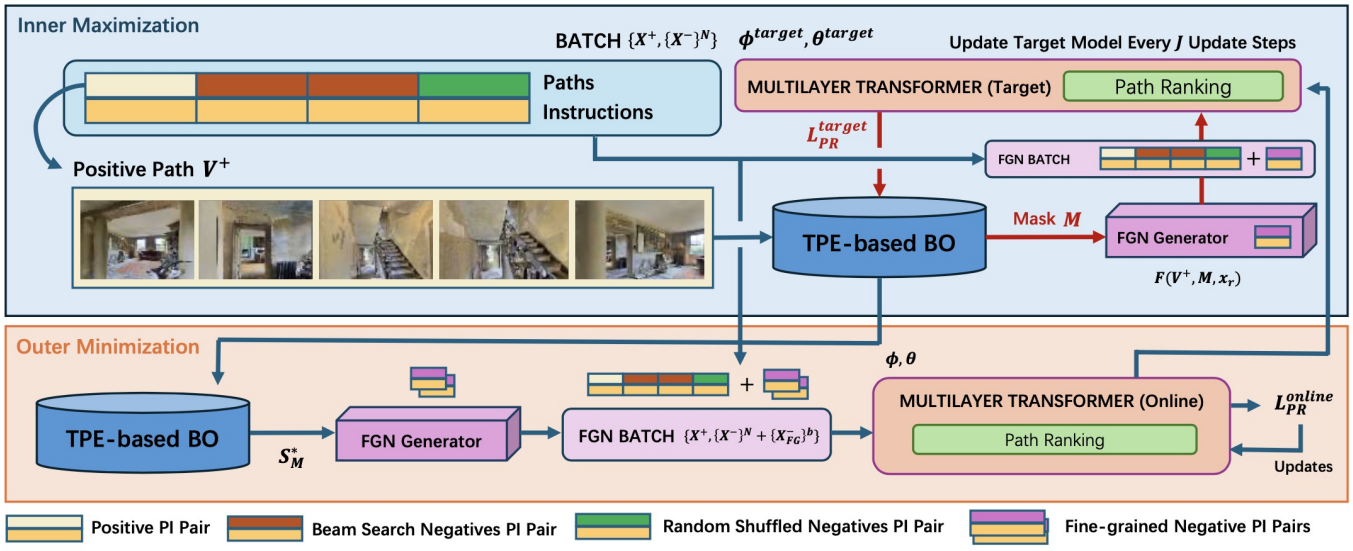

Figure 2. Overview of the proposed Fine-grained VLN (FGVLN). In the *Inner Maximization*, the Bayesian optimizer evaluates different masks M based on L_{PR}^{target}, this process is repeated several iterations (as denoted by lines in red), and resulting a set of best masks S_M^*. In outer minimization procedure, the online model is updated given the FGN batch generated based on S_M^*.

Fig. 2 illustrates the proposed adversarial training framework, named Fine-Grained VLN (FGVLN), which utilizes BO to generate fine-grained vision negative samples. The framework comprises two optimization processes: *inner maximization* and *outer minimization*. The *inner maximization* process aims to discover the most effective fine-grained vision negatives that maximize L_{PR}, while the *outer minimization* procedure employs these negatives to train our model to minimize L_{PR}. Specifically, during each round of *outer minimization*, an *inner maximization* process trains a BO-based sampler to identify the most impactful frames in the positive trajectory for replacement. The *outer minimization* then utilizes the trained BO model to sample fine-grained negative PI pairs and optimize the model's learning based on these negatives. Since both processes need to assess L_{PR}, the framework maintains two multilayer Transformer-based ViLBERT[33] models for each process: an *online model* for the outer process, parameterized by ϕ, θ, and a *target model* for the inner process, parameterized by ϕ^{target}, θ^{target}. The *target model* is a copy of the *online model* and is periodically updated by it. The loss from the *online model* L_{PR}^{online} is used to update the *online model* itself, while the loss from the *target model* L_{PR}^{target} is for evaluating the discovered fine-grained negatives.

In the *inner maximization* process, a Tree-structured Parzen Estimator (TPE) based BO model[42] is first initialized. Given a positive trajectory, the BO model samples several frames from the positive trajectory. These sampled frames are then transformed into replacement frames by a fine-grained negative (FGN) generator, which results in a fine-grained vision negative PI pair that consists of a fine-grained negative path and a positive instruction. The generated fine-grained negative PI pairs are concatenated with the PI pairs in the original batch to form a new batch referred to as the FGN batch. This batch is then passed to the *target model* to determine their difficulties, quantified through L_{PR}^{target}. This procedure is repeated for several iterations to optimize the BO sampler, and the result is an optimized BO model employed by the outer process. During the *outer minimization*, based on the sampling results from the BO model, the generated fine-grained negative PI pairs are concatenated with other PI pairs in the batch to form a final batch, which is

employed to train the *online model*.

1. *Inner-Maximization*

Defining a fine-grained negative PI pair as $X_{FG}^{-} = (X_{FG}^{-}, X_{L}^{+})$, the framework aims to select b best fine-grained negative pairs $\{X_{FG}^{-}\}^{b}$ in conjunction with other negative pairs $\{X^{-}\}^{N}$ to maximize L_{PR}^{target}. A TPE-based Bayesian optimizer is employed to select the frames for modification. Given an unprocessed positive path v^{+} with K frames, the optimizer samples a mask indicator $M = (m_1, m_2, \ldots, m_K) \in R^{K}$. This binary mask M indicates the frames to be replaced, and $m_k = 1, k \in K$ signifies that frame k is to be replaced. The objective function for this can be written as follows:

$$\max L_{PR}^{target}(X^{+}, \{X^{-}\}^{N} + \{X_{FG}^{-}\}^{b}).$$

This process is iterated R times to maximize L_{PR}^{target}, after which the optimal M is selected. To produce the fine-grained negatives, a generation function $F(v^{+}, M, x_r)$ replaces the frames indicated by M in the positive trajectory v^{+} with a replacement frame x_r to produce X_{FG}^{-}. The generation flow for the replacement frame is discussed in Section V-D. The generation function F is defined as:

$$X_{FG}^{-} \triangleq F(v^{+}, M, x_r) = v^{+} \cdot \hat{M} + x_r \cdot M,$$

where $\hat{M}$ represents the complement of M. By selecting b optimal masks to obtain a set of masks $S_M = \{M\}^{b} \in R^{b \times K}$, the objective can be formulated as maximizing L_{PR}^{target} with respect to S_M:

$$\max_{S_M} L_{PR}^{target}(X^{+}, \{X^{-}\}^{N} + \{X_{FG}^{-}\}^{b}).$$

After iterations, the inner-maximization process eventually results in a set of b optimal masks S_M^{*}.

2. *Outer-Minimization*

The outer-minimization process receives the result from the inner-maximization process, and utilizes the generation function in Eq. (4) to produce b fine-grained negatives $\{X_{FG}^{-}\}^{b}$. These fine-grained negative PI pairs are concatenated with other negative PI pairs $\{X^{-}\}^{N}$ to produce $\{X^{-}\}_{cat}^{N} = \{X^{-}\}^{N} + \{X_{FG}^{-}\}^{b}$. The objective of this process is to minimize L_{PR}^{online} given S_M^{*}, formulated as:

$$\min_{\theta, \phi} L_{PR}^{online}(X^{+}, \{X^{-}\}_{cat}^{N}) \quad \text{subject to} \quad S_M^{*}.$$

B. *Delayed Updates*

Given that the inner optimization process optimizes based on the output from the learned encoders, which are subsequently updated by the outer optimization stream, employing rapid updates across both processes could potentially lead to the selection of a suboptimal mask set S_M as validated later in Section. V-D. This issue is particularly pronounced during the initial stages of training, where the outputs of the encoders in both processes may not accurately reflect the

desired embeddings. This discrepancy could affect the direction of gradient descent in the outer optimization stream, and potentially lead to a feedback loop that detracts from model performance. To mitigate this issue, we propose maintaining a separate copy of the model parameters within the inner optimization process, i.e., θ_t and ϕ_t. These parameters are updated after a fixed period of time to align with the model in the outer optimization process every J update steps. This strategy enables the inner optimization process to perform more stable and reliable frame selections, which reduces the likelihood of misleading gradients that can adversely impact the outer optimization process.

V. Experiments

In this section, we present the experiments for addressing three key aspects: (1) evaluating the effectiveness of the embeddings for fine-grained vision negatives after applying the proposed method in comparison to the previous approach, (2) determining the extent to which these improved embeddings enhance the current model's performance in both discriminative and generative settings, and (3) exploring the design space of the BO-based sampler by an ablation study.

A. *Experimental Setup*

Baselines.

To evaluate the navigation performance of the proposed framework, we compare the navigation results of our framework to the existing works in the discriminative setting that improve learned embeddings through various types of data augmentations. In the generative setting, we adapt our encoders into[41] and compare the performance of our framework to the existing end-to-end generative navigation methods that enrich the embeddings solely through data augmentation. The baselines for these settings are presented in Tables II, III and IV, respectively.

Benchmark and Metrics.

We first evaluate our proposed method on the common VLN benchmark R2R[2] in discriminative setting, which contains detailed paired instructions and photo-realistic observations. R2R is based on the Matterport 3D[43] dataset, containing a total of 21,567 path-instruction pairs from 90 scenes. Following the standard setting presented in[2], we adopt several representative metrics for evaluating R2R: success rate (SR), success rate weighted by the ratio between the shortest path length and the predicted path length (SPL), trajectory length (TL), as well as navigation error (NE). We also adapt our trained backbone into the generative setting on two benchmarks, R2R and REVERIE. For REVERIE, we use four metrics to evaluate navigation performance: SR, OSR, SPL, and TL as in[7]. Additionally, we assess object grounding performance using two metrics: remote grounding success (RGS) and RGS weighted by path length (RGSPL). Following standard settings[8]. The REVERIE dataset uses the same data splits as the R2R dataset, but it additionally requires the agent to select the bounding box of the target object.

Implementation Details.

The framework was implemented using the PyTorch framework and followed a two-stage training process:*pre-training* and *fine-tuning*. For pre-training, we utilized the pre-trained model described in Lily[8]. During the fine-tuning phase, we adhered to the settings outlined in[8] to ensure a fair comparison. This process involved initially training the model with Masked Language Modeling (MLM)[7] and Masked Vision Modeling (MVM)[7] losses. The training was conducted with a batch size of 12 across four NVIDIA Tesla V100 GPUs, and a learning rate of4×10^{-5}. Subsequently, the model was further trained using our framework on L_{PR}, distributed across eight NVIDIA Tesla V100 GPUs, with a learning rate of 1×10^{-5} and a batch size of 16 for 30 epochs until convergence. The models included in the ablation studies were trained on subsets using the default settings provided in[8], with a batch size of eight. For adaptation to the generative setting, we followed the methodology outlined in[7] to adapt recurrent VLN[41]. Our trained FGVLN model served as the backbone network for the recurrent VLN and was trained using imitation learning and A2C[44] for 300,000 iterations. This training was conducted on a single NVIDIA GeForce RTX 4080 GPU, with a batch size of eight and a learning rate of 1×10^{-5}.

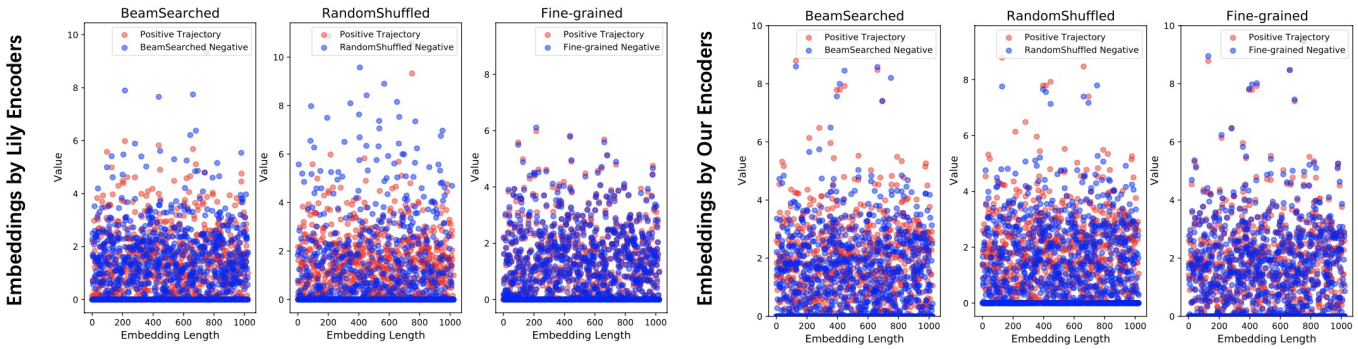

Figure 3. A comparison of the embeddings from the vision encoder trained by different methods.

B. *Examination on the Learned Embeddings*

We examine the embeddings h_{v_i} from Eq. (1). These embeddings are derived by the vision encoder trained by different methods. To demonstrate the impact of our method across different negative PI pairs, we utilize PI pairs sampled from the R2R validation dataset and plot the embeddings from the positive trajectories and the altered negatives. Fig. 3 presents a comparison of the embeddings generated by Lily[8] and our FGVLN, in which the red dots represent the embedding entry from the positive trajectory, while the blue dots denote the embedding of negative samples generated from different approaches, including random shuffling, beam search, and fine-grained replacement. It can be observed that the negative embeddings generated by both encoders through random shuffling and beam search display diverse and distinguishable distributions compared to the embeddings of the original positive trajectories. However, when encoding fine-grained negative vision-based PI pairs, Lily encodes these pairs in a manner highly similar to the positive path, which results in a significant overlap of the dots. In contrast, our method captures subtle differences in information from fine-grained negative paths and can produce embeddings with better distinguishability.

Table 1. Statistical resuts of *L2* distance of embeddings.

Encoder	Negative Path Generated by Various Method					
	Beamsearch		RandomShuffle		Fine-grained	
	μ	σ	μ	σ	μ	σ
Lily[8]	13.47	81.44	74.80	22.12	4.72	95.79
Ours	13.32	131.19	43.25	200.18	7.64	47.35

Table 1 further presents a statistical analysis based on 1,000 sampled PI pairs of the*L2* distance between the embeddings of the trajectories encoded by different encoders. The results reveal that negative embeddings generated by random shuffling diverge the most from the embeddings of the positive trajectories. Negative embeddings generated through beam search exhibit the second-highest divergence, while fine-grained negative trajectories show the least divergence. The encoder trained by our approach captures more subtle differences even after fine-grained alteration.

C. *Navigation Performance on the R2R Benchmark*

Table II. Comparison on R2R under the discriminative setting.

Methods	Val Seen				Val Unseen			
	TL	NE (↓)	SR (↑)	SPL (↑)	TL	NE (↓)	SR (↑)	SPL (↑)
VLN-BERT[6]	10.28	3.73	70.20	0.66	9.60	4.10	59.26	0.55
Airbert[7]	10.21	3.14	74.12	0.70	9.63	3.95	62.84	0.58
Lily[8]	9.99	3.12	77.45	0.74	9.64	3.37	66.70	0.62
FGVLN (Ours)	10.05	**3.08**	**78.59**	**0.74**	9.79	3.40	**67.69**	**0.64**

Discriminative VLN.

We employ the pre-trained Lily[8] model and fine-tune it with our proposed FGVLN on the complete R2R benchmark under the discriminative setting. The performance of our model is compared with the previous baseline models. Table II presents the results of this comparison. Our FGVLN model outperforms all the previous models on the validation unseen datasets. In the validation unseen dataset, our model achieves a 1.48% improvement in terms of SR and a3.12% improvement in terms of SPL compared to the current state-of-the-art (SOTA) Lily model[8] that does not utilize BO for fine-grained negative sampling. These results confirm that incorporating challenging fine-grained vision negatives produced by BO into the training process enhances the performance of VLN models in the discriminative setting. Fig. 4 illustrates an example of the navigation trajectory determined by our framework compared to that determined by Lily[8]. It can be observed that with the enhanced embeddings, our framework is able to determine a trajectory with better alignment to the given instruction, which results in fine-grained inferencing.

Table III. Comparison on R2R under the generative setting.

Methods	Validation Seen				Validation Unseen			
	TL	NE (↓)	SR (↑)	SPL (↑)	TL	NE (↓)	SR (↑)	SPL (↑)
Seq2Seq-SF[2]	11.33	6.01	39	-	8.39	7.81	22	-
Speaker-Follower[26]	-	3.36	66	-	-	6.62	35	-
PRESS[45]	10.57	4.39	58	55	10.36	5.28	49	45
EnvDrop[13]	11.00	3.99	62	59	10.70	5.22	52	48
PREVALENT[46]	10.32	3.67	69	65	10.19	4.71	58	53
Rec (Airbert)[7]	10.31	**2.68**	**74**	66	12.12	**4.01**	59	54
Rec (FGVLN)	11.42	2.77	73	**68**	12.74	4.06	**61**	**55**

Table IV. Comparison with models with different backbones on REVERIE dataset under generative setting

Methods	Navigation				RGS	RGSPL
	SR	OSR	SPL	TL		
Random	1.7	11.93	1.01	10.76	0.96	0.56
Rec (OSCAR)[41]	25.53	27.66	21.06	14.35	14.20	12.00
Rec (ViLBert)[33]	24.57	29.91	19.81	17.83	15.14	12.15
Rec (VLN-Bert)[40]	25.53	29.42	20.51	16.94	16.42	13.29
Rec (AirBert)[7]	27.89	**34.51**	21.88	18.71	18.23	14.18
Rec (FGVLN)	**28.71**	30.14	**22.09**	19.10	**21.55**	**14.78**

Adaptation to Generative VLN.

Following the same adaptation scheme as[7], we further use our trained FGVLN as the backbone of the recurrent VLN[41] and adapt our model in the R2R and REVERIE under the generative setting. For R2R, we compare the performance of the models that were only fine-tuned on the original R2R dataset, without any augmented data from[13]. Table III presents the results of the navigation performance comparison of our method against the previous baseline approaches. It can be observed that FGVLN achieves the highest SPL in the validation seen split while maintaining comparable performance in terms of SR. In the validation unseen split, the proposed FGVLN outperforms all previous models, and achieves the best performance in both SR and SPL. The superior performance in the generative setting, especially in SPL, indicates that our encoders produce more aligned embeddings. This alignment assists the agent in closely following the designated instructions.

Table IV summarizes the navigation performance on the REVERIE dataset in previouse unseen environments under the generative setting. Our FGVLN approach demonstrates competitive results, particularly while generating to the unseen environments. Notably, in the validation unseen split, FGVLN achieves a Success Rate (SR) of 28.71%, and a higher SPL of 22.09%, indicating more efficient navigation and generalizing ability in unfamiliar environments. This suggests that our method allows the agent to follow instructions more closely and accurately, despite the complex and unseen scenarios

presented by the REVERIE dataset. These results validate the robustness of our Bayesian Optimization-based fine-grained negative sampling approach.

Figure 4. An illustration of an example trajectory determined by our framework for a given instruction compared to that determined by Lily. Each robot starts at position 0 (marked in blue). Our framework selects a path (marked in green) that stops at the top of the stairs, while the baseline selects a path (marked in yellow) that only ascends partway up the stairs before stopping in the middle.

Table V. Ablation Studies on Bayesian optimization-based sampler.

Index	Model Name	Bayesian Optimizer Configurations							Result (SR%)	
		3 Iters	5 Iters	Delayed	1FGN	2FGNs	In-domain	Out-domain	val_seen	val_unseen
1	Baseline Lily[8]	-	-	-	-	-	-	-	60.21	51.38
2	FGVLN-Rand	-	-	-	-	✓	-	✓	60.61	50.11
3	FGVLN-w/o-delayed	✓	-	-	✓	-	✓	-	60.18	49.52
4	FGVLN-w-delayed	✓	-	✓	✓	-	✓	-	57.66	51.02
5	FGVLN-outdomain	✓	-	✓	✓	-	-	✓	61.25	52.36
6	FGVLN-add-FGN	✓	-	✓	-	✓	-	✓	**63.48**	53.14
7	FGVLN-add-iter	-	✓	✓	-	✓	-	✓	61.98	**56.45**

**Models were tested under various configurations, including (1) -# Iters the different number of BO optimization iterations, (2) -Delayedthe use of delayed updates (3) -#FGNs the different number of the fine-grained negatives to sample for in each batch (4) -In-domain/Out-domain the selection of the replacement frame x_r, which could be either in-domain, aligning with the positive trajectory, or out-domain.*

D. *Ablation Study*

To determine the optimal configurations for FGVLN, we conducted a series of design space explorations. We utilized a subset of the original dataset for this exploration to efficiently explore the design space. Table V presents the comparison of FGVLN under different configurations, with explanations for each configuration included. This ablation study identifies *FGVLN-add-iter* as the best configuration, which outperforms all other settings in unseen environments. As a result, we

adopt this configuration for our FGVLN in all other experiments presented.

Effectiveness of Delayed Updates

To validate the effectiveness of the proposed delayed updates as described in Section 4.2, the comparisons in Table 4 of the main manuscript between the model with delayed updates (index 4) and another without updates (index 3) show that the model with delayed updates exhibited a 3% performance improvement on the unseen validation set. This finding supports our hypothesis regarding the benefits of delayed updates.

Effectiveness of Out-domain Replacement

To evaluate the impact of using different types of replacement frames x_r to generate fine-grained negatives, we assessed a strategy to generate the replacement frame by sampling a frame from an in-domain trajectory, specifically from the same room as the positive path, with results detailed in indices 3-4 in Table 4 of the main manuscript. In contrast, we also tested out-domain replacement frames, which were sampled from a different room (i.e., index 5). The results revealed that out-domain replacement frames are more effective. Under this setting, the model achieved a 2.6% improvement over the best in-domain x_r approach and a 1.9% improvement compared to the baseline model. We assume that this is due to potential overfitting caused by the in-domain replacement, which generates negative samples that are overly similar to the positive path and thus not sufficiently informative.

Effectiveness of Optimizer & Number of Additional Negatives

We assessed the impact of the number of iterations conducted by the Bayesian optimizer on mask selection. In particular, the configuration of the optimizer to produce two masks, as presented in index 6 of Table 4 in the main manuscript, resulted in two additional fine-grained negatives and enhanced performance on both the validation seen and unseen datasets compared to the previous models. This finding highlights the benefits of multiple fine-grained negatives. In addition, extending the optimizer's iterations (i.e., index 7) improved performance in the unseen dataset, which emphasizes the optimizer's effectiveness. However, for the seen dataset, the model with three iterations (row 5) performed better. This suggests that while additional iterations aid generalization in new environments, they may not yield the same benefits in familiar settings. This indicates a need for balanced optimization strategies tailored to various environmental complexities. As we focus more on the unseen rooms in VLN, we select the model setting with the best performance in the unseen dataset for all our experiments, which is referred to as FGVLN in the main manuscript.

Random Mask Selector

We also evaluated the model using a random mask selector under the optimal fine-grained negative setting (i.e., two additional negatives, using out-domain replacement frames) as presented in index 2 of Table 4. It can be observed that all models employing the selector based on the Bayesian optimization with identical fine-grained negative settings (index 6-7) demonstrated superior performance compared to the random mask selector. This finding confirms the effectiveness of the Bayesian optimization component.

VI. Conclusion

We propose a BO-based approach for generating fine-grained negatives was introduced by presenting the FGVLN framework. An analysis of the resulting embeddings of our encoders was provided. Experimental results demonstrated that the proposed framework is capable of capturing better fine-grained correspondence between paths and their corresponding instructions. This correspondence enables the model to make more informed decisions in VLN tasks. The performance of the encoders trained by our proposed framework was also assessed on the well-established VLN benchmark R2R, in both discriminative and generative settings, and a significant navigation performance enhancement was observed. Finally, an ablation study was provided to validate the design decisions.

References

1. [a, b] *Vaswani A, et al. (2017). "Attention is all you need."Advances in neural information processing systems (NIPS).*
2. [a, b, c, d, e, f, g] *Anderson P, et al. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proc. IEEE Conf. on Comp. Vision and Pattern Recog. (CVPR); 2018.*
3. [^] *Zachiotis GA, et al. "A Survey on the Application Trends of Home Service Robotics." In:Proc. IEEE Int. Conf. on Robotics and Biomimetics (ROBIO); 2018.*
4. [^] *Anderson P, et al. "Sim-to-real transfer for vision-and-language navigation." InProc. IEEE/CVF Int. Conf. on Comp. Vision (ICCV), 2021.*
5. [^] *Zhao C. Vision-and-Language Navigation in the Real-World [Ph.D. dissertation]. 2023.*
6. [a, b, c, d, e, f, g, h, i] *Majumdar A, Shrivastava A, Lee S, Anderson P, Parikh D, Batra D. "Improving vision-and-language navigation with image-text pairs from the web." In: In Proc. Eur. Conf. on Comp. Vision (ECCV). Springer; 2020.*
7. [a, b, c, d, e, f, g, h, i, j, k, l, m, n] *Guhur P-L, Tapaswi M, Chen S, Laptev I, Schmid C. "Airbert: In-domain pretraining for vision-and-language navigation." In: In Proc. IEEE/CVF Int. Conf. on Comp. Vision (ICCV); 2021.*
8. [a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q] *Lin K, Chen P, Huang D, Li TH, Tan M, Gan C (2023). "Learning vision-and-language navigation from youtube videos." In: Proc. IEEE/CVF Int. Conf. on Comp. Vision (ICCV).*
9. [^] *Robinson J, Chuang C-Y, Sra S, Jegelka S (2020). "Contrastive learning with hard negative samples." arXiv:2010.04592. 2020.*
10. [^] *Guo H, Shi L. "Ultimate Negative Sampling for Contrastive Learning." In:In Proc. ICASSP 2023-2023 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2023.*
11. [^] *Choi H, Beedu A, Essa I (2023). "Multimodal contrastive learning with hard negative sampling for human activity recognition." arXiv:2309.01262.*
12. [^] *Gupta T, et al. Contrastive learning for weakly supervised phrase grounding. In:In Proc. Eur. Conf. on Computer Vision (ECCV). Springer; 2020.*
13. [a, b, c, d] *Tan H, Yu L, Bansal M (2019). "Learning to navigate unseen environments: Back translation with environmental dropout." arXiv:1904.04195.*

14. ^Mu R, Ruan W, Marcolino LS, Ni Q (2021). "Sparse adversarial video attacks with spatial transformations." arXiv:2111.05468. 2021.
15. a, bQi Y, et al. "Reverie: Remote embodied visual referring expression in real indoor environments." InProc. IEEE/CVF Conf. on Comp. Vision and Pattern Recog. (CVPR), 2020.
16. ^Anderson P, et al. (2018). "On evaluation of embodied navigation agents."arXiv:1807.06757.
17. ^Chen H, et al. "Touchdown: Natural language navigation and spatial reasoning in visual street environments." InProc. IEEE/CVF Conf. on Comp. Vision and Pattern Recog. (CVPR), 2019.
18. ^Krantz J, Wijmans E, Majumdar A, Batra D, Lee S. "Beyond the nav-graph: Vision-and-language navigation in continuous environments." In: In Proc. Eur. Conf. on Comp. Vision (ECCV). Springer; 2020.
19. ^Nguyen K, Daumé III H. "Help, Anna! Visual Navigation with Natural Multimodal Assistance via Retrospective Curiosity-Encouraging Imitation Learning." In: Inui K, Jiang J, Ng V, Wan X, editors. In Proc. Empirical Methods in Nat. Lang. Proc. and 9th Int. Joint Conf. on Nat. Lang. Proc. (EMNLP-IJCNLP); 2019.
20. ^Nguyen K, Dey D, Brockett C, Dolan B (2019). "Vision-based navigation with language-based assistance via imitation learning with indirect intervention." In: Proc. IEEE/CVF Conf. on Comp. Vision and Pattern Recog. (CVPR).
21. ^Shridhar M, et al. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In:Proc. IEEE/CVF Conf. on Comp. Vision and Pattern Recog. (CVPR); 2020.
22. ^Thomason J, Murray M, Cakmak M, Zettlemoyer L. "Vision-and-dialog navigation." In:In Proc. Conf. on Robot Learning (CoRL). PMLR; 2020.
23. ^Ding M, et al. "Embodied concept learner: Self-supervised learning of concepts and mapping through instruction following." In Proc. Conf. on Robot Learning (CoRL). PMLR; 2023.
24. ^Song CH, et al. "One step at a time: Long-horizon vision-and-language navigation with milestones." InProc. IEEE/CVF Conf. on Comp. Vision and Pattern Recog. (CVPR), 2022.
25. ^Anderson P, Shrivastava A, Parikh D, Batra D, Lee S (2019). "Chasing ghosts: Instruction following as bayesian state tracking." Advances in neural information processing systems (NIPS).
26. a, b, cFried D, et al. (2018). "Speaker-follower models for vision-and-language navigation."Advances in neural information processing systems (NIPS).
27. ^Ke L, et al. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. InProc. IEEE/CVF Conf. on Comp. Vision and Pattern Recog. (CVPR); 2019.
28. a, bMa C-Y, Lu J, Wu Z, AlRegib G, Kira Z, Socher R, Xiong C (2019). "Self-monitoring navigation agent via auxiliary progress estimation." arXiv:1901.03035.
29. ^Ma C-Y, Wu Z, AlRegib G, Xiong C, Kira Z (2019). "The regretful agent: Heuristic-aided navigation through progress estimation." In: Proc. IEEE/CVF Conf. on Comp. Vision and Pattern Recog. (CVPR).
30. a, bWang X, et al. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In: Proc. IEEE/CVF Conf. on Comp. Vision and Pattern Recog. (CVPR); 2019.
31. ^Wang X, Xiong W, Wang H, Wang WY (2018). "Look before you leap: Bridging model-free and model-based

reinforcement learning for planned-ahead vision-and-language navigation." In: Proc. Eur. Conf. on Comp. Vision (ECCV).

32. ^*Liang X, Zhu F, Li L, Xu H, Liang X (2022). "Visual-language navigation pretraining via prompt-based environmental self-exploration". arXiv:2203.04006.*
33. a, b, c, d, e, f, g*Lu J, Batra D, Parikh D, Lee S (2019). "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". Advances in neural information processing systems (NIPS)*
34. ^*Qi Y, Pan Z, Zhang S, van den Hengel A, Wu Q. "Object-and-action aware model for visual language navigation." In: In Proc. Eur. Conf. on Comp. Vision (ECCV) Springer; 2020.*
35. ^*Zhu F, Zhu Y, Chang X, Liang X (2020). "Vision-language navigation with self-supervised auxiliary reasoning tasks." In: Proc. IEEE/CVF Conf. on Comp. Vision and Pattern Recog. (CVPR).*
36. ^*Qiao Y, Qi Y, Hong Y, Yu Z, Wang P, Wu Q. "Hop: History-and-order aware pre-training for vision-and-language navigation." In: Proc. IEEE/CVF Conf. on Comp. Vision and Pattern Recog. (CVPR); 2022.*
37. ^*Chen S, Guhur PL, Schmid C, Laptev I (2021). "History aware multimodal transformer for vision-and-language navigation." Advances in neural information processing systems (NIPS) 2021.*
38. ^*Moudgil A, Majumdar A, Agrawal H, Lee S, Batra D (2021). "Soat: A scene-and object-aware transformer for vision-and-language navigation." Advances in Neural Information Processing Systems (NIPS).*
39. ^*Kuo C-W, Ma C-Y, Hoffman J, Kira Z (2023). "Structure-encoding auxiliary tasks for improved visual representation in vision-and-language navigation." In: Proc. IEEE/CVF Winter Conf. on Appl. of Comp. Vision (WACV)*
40. a, b, c*Devlin J, Chang M-W, Lee K, Toutanova K (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv:1810.04805.*
41. a, b, c, d, e, f*Hong Y, et al. VLN-BERT: A Recurrent Vision-and-Language BERT for Navigation. In:Proc. IEEE/CVF Conf. on Comp. Vision and Pattern Recog. (CVPR); 2021 Jun.*
42. ^*Watanabe S. "Tree-structured Parzen estimator: Understanding its algorithm components and their roles for better empirical performance." arXiv:2304.11127. 2023.*
43. ^*Chang A, Dai A, Funkhouser T, Halber M, Niessner M, Savva M, Song S, Zeng A, Zhang Y (2017). "Matterport3d: Learning from rgb-d data in indoor environments." arXiv:1709.06158.*
44. ^*Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K. "Asynchronous methods for deep reinforcement learning." In: In Proc. Int. Conf. on Machine Learning (ICML) PMLR; 2016. p. 1928-1937.*
45. ^*Li X, Li C, Xia Q, Bisk Y, Celikyilmaz A, Gao J, Smith N, Choi Y (2019). "Robust navigation with language pretraining and stochastic sampling". Proc. of the EMNLP-IJCNLP.*
46. ^*Hao W, Li C, Li X, Carin L, Gao J (2020). "Towards learning a generic agent for vision-and-language navigation via pre-training." In Proc. IEEE/CVF Conf. on Comp. Vision and Pattern Recog. (CVPR).*