Distilling LLM Prior to Flow Model for Generalizable Agent's Imagination in Object Goal Navigation

Badi Li[1,2], **Ren-Jie Lu**[1] , **Yu Zhou**[1], **Jingke Meng**[1,*], **Wei-Shi Zheng**[1,3]
[1]Sun Yat-sen University [2] The University of Hong Kong
[3]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education
badi.li.cs@connect.hku.hk, mengjke@gmail.com, wszheng@ieee.org

Abstract

The Object Goal Navigation (ObjectNav) task challenges agents to locate a specified object in an unseen environment by imagining unobserved regions of the scene. Prior approaches rely on deterministic and discriminative models to complete semantic maps, overlooking the inherent uncertainty in indoor layouts and limiting their ability to generalize to unseen environments. In this work, we propose GOAL, a generative flow-based framework that models the semantic distribution of indoor environments by bridging observed regions with LLM-enriched full-scene semantic maps. During training, spatial priors inferred from large language models (LLMs) are encoded as two-dimensional Gaussian fields and injected into target maps, distilling rich contextual knowledge into the flow model and enabling more generalizable completions. Extensive experiments demonstrate that GOAL achieves state-of-the-art performance on MP3D and Gibson, and shows strong generalization in transfer settings to HM3D. Codes and pretrained models are available at https://github.com/Badi-Li/GOAL.

1 Introduction

Embodied navigation [2, 22, 30, 43, 56, 70], which enables agents to move purposefully through complex, realistic environments, is a fundamental challenge in embodied intelligence. Within this domain, Object Goal Navigation (ObjectNav) tasks an agent with locating an instance of a user-specified object category (e.g., "find a chair") in an unseen environment, relying solely on visual observations.

To succeed at ObjectNav, the agent must not only recognize the goal object when it becomes visible but also infer its likely location before it is seen. This imagination step is particularly challenging, as it requires reasoning about contextual and co-occurrence relationships between objects (e.g., chairs often appear near tables). Recent approaches [17, 25, 75] address this by incrementally constructing top-down semantic maps and predicting full-scene semantic maps through discriminative and deterministic models. However, their deterministic nature, which directly maps inputs to fixed outputs with a strict one-to-one mapping, inherently limits generalization to unseen data.

In contrast, we argue that semantic map completion is inherently uncertain: Multiple plausible full scenes can correspond to the same partial map, and multimodal outputs could benefit generalization capabilities [10]. We therefore formulate this task as a probabilistic generation problem, leveraging recent advances in flow-based generative modeling [21, 26, 28, 53, 54] to learn the semantic distribution of indoor scenes (illustrated in Fig. 1). However, while generative models offer better generalizability, we find that applying generative models to ObjectNav poses three core challenges, which we address in this work.

* Corresponding author.

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

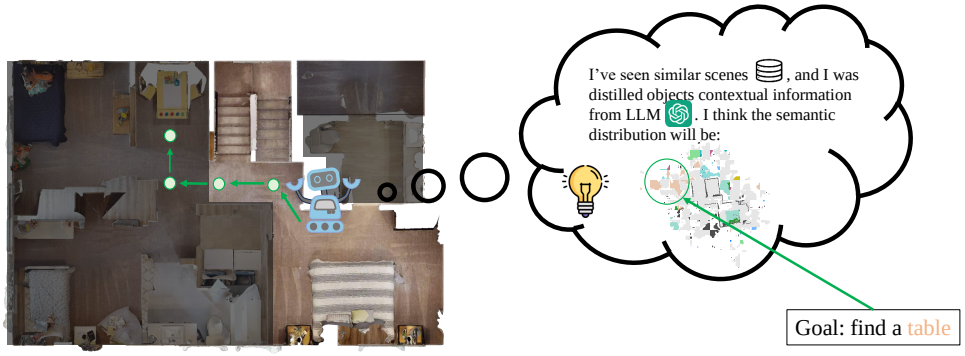


Figure 1: We incorporate a flow model to generate the semantic distribution of unobserved regions (in dark), based on dataset-internal patterns and external knowledge from language models.

First, generative models typically require large and diverse datasets to effectively learn the latent distribution. However, existing indoor scene datasets remain limited in both scale and variety. To overcome this, we incorporate external knowledge from large language models (LLMs) by *firstly modeling it as a natural Gaussian distribution in the latent space of the flow model, which enriches contextual signals during training*. In particular, we prompt LLMs to infer spatial contextual priors, including common distances between object pairs and associated confidence scores. These distance-confidence pairs are transformed into two-dimensional Gaussian priors and injected into semantic maps during training. This process distills the rich contextual information from LLM to our generative flow model, enhancing its understanding of object co-occurrence and spatial context. Crucially, this external supervision is applied only during training, avoiding inference-time costs such as API latency and memory overhead, and enabling the flow model to operate as a plug-and-play semantic reasoner.

Second, we identify the problem of inefficient conditioning. Traditional diffusion models assume the source distribution to be standard Gaussian and develop their theory based on this assumption. As a result, conditional generation in the diffusion literature typically relies on additional mechanisms, such as cross-attention, concatenation, or FiLM [38], to incorporate conditioning information. While these mechanisms are acceptable for image generation and restoration, they introduce additional computation, which may be prohibitive in visual navigation tasks that require multiple inferences per episode and real-time interaction. In contrast, the Flow Matching algorithm does not strictly assume a Gaussian source distribution, allowing us to design a more efficient conditioning mechanism: *directly modeling the dependent couplings between noise injected partial semantci maps and full LLM-enriched semantic maps*.

Third, semantic maps constructed during navigation are prone to accumulating errors from upstream segmentation models, which can degrade the performance of generative models. To mitigate this, *we aggregate past RGB-D observations into unified point clouds representations and perform joint segmentation using 3D perception models*, inspired by how humans implicitly integrate multi-frame observations. This method captures both spatial geometry and temporal consistency more effectively than traditional 2D semantic segmentation methods used in ObjectNav [20, 23]. As a result, we achieve more accurate and consistent scene understanding.

In conclusion, we propose GOAL (**G**uiding Agent's imaginati**O**n with gener**A**ive f**L**ow), a generative framework that incorporates external knowledge as training supervision, models direct couplings to better leverage semantic map priors, and integrates multi-view observations for enhanced scene-level understanding. These components together lead to strong and generalizable performance on the ObjectNav task.

Evaluations on large-scale datasets Gibson[63] and MP3D [8] show that our approach significantly outperforms baselines. Additionally, transfer experiments, training on MP3D and testing on HM3D [41], demonstrate the strong generalization capabilities of our approach to unseen environments.

2 Related Work

ObjectGoal navigation. ObjectNav approaches fall into two main categories: end-to-end and modular. End-to-end methods map visual inputs to actions using reinforcement or imitation learning,

focusing on improving visual representations [7, 64, 65, 66] or tackling policy learning challenges like sparse rewards and overfitting [42, 50, 59, 67, 68]. Modular methods decompose the task into components such as mapping, planning, and policy learning. Given a semantic map, these methods explore waypoint or frontier selection [9, 32, 40], distance estimation [79], target probability prediction [73], and semantic map completion [17, 25, 75]. T-Diff [72] firstly introduced generative modeling to ObjectNav via a DDPM conditioned on semantic maps for trajectory generation. Alternatively , we propose a generative flow-based model that imagines full-scene semantics, using LLM-derived priors to improve generalization to unseen environments.

Diffusion and flow-based generative models. Generative models such as diffusion [21], and flow-based methods [26, 28] generate data via iterative denoising or learned velocity fields. They have shown strong performance in image generation and restoration tasks [31, 45, 47, 48, 78]. We draw an analogy between agent's imagination via completing a partial semantic map and image restoration, where the goal is to reconstruct missing content from degraded input. Most approaches begin from Gaussian noise and condition on the input via concatenation, FiLM [38], or cross-attention. Recent alternatives have explored directly bridging degraded images and targets via Schrödinger Bridges [27] and stochastic interpolants [1] , but often trade off quality for interpretability. In contrast, we show that for sparse semantic maps, direct coupling via flow matching can be more efficient without sacrificing the performance. We adopt flow matching framework for its faster sampling and mathematically simple yet expressive formulation.

3 Flow Matching Preliminaries

The generative task is typically defined as a mapping $\psi_t,\ t \in [0, 1]$, which transports samples X_0 from a source data distribution p to samples X_1 in a target distribution q. In general, the source and target samples may come from a joint distribution $(X_0, X_1) \sim \pi_{0,1}(X_0, X_1)$.

To address this transformation from X_0 to X_1, Flow Matching [26] interpolates a probability path p_t between the source and target samples:

$$X_t = \alpha_t X_0 + \beta_t X_1 \sim p_t, \tag{1}$$

with boundary conditions $\alpha_0 = \beta_1 = 1$ and $\alpha_1 = \beta_0 = 0$. To learn this path, we solve the following Ordinary Differential Equation (ODE):

$$\frac{d}{dt} X_t = u_t(X_t), \tag{2}$$

where u_t is a time-dependent velocity field, also referred to as the *drift*, typically parameterized by a neural network u_t^θ which trained by minimizing:

$$\mathcal{L}_{FM}(\theta) = \mathbb{E}_{t, X_t \sim p_t} \left[D\left(\dot{X}_t, u_t^\theta(X_t) \right) \right], \tag{3}$$

where $\dot{X}_t = \frac{d}{dt} X_t$ denotes time-derivative of X_t and D denotes general Bregman Divergences measuring the dissimilarity. Once trained, this velocity field can be used to generate samples by integrating the ODE (Eq. 2) numerically. In our work, we solve the ODE by simplest method Euler integration:

$$X_{t+h} = X_t + h u_t^\theta(X_t), \tag{4}$$

where $h = \frac{1}{n}$ and n is a hyperparameter representing the number of forward steps.

4 Method

4.1 ObjectNav Definition

We consider the ObjectGoal Navigation (ObjectNav) task, where an embodied agent is initialized at a random location in an unseen indoor environment and is instructed to navigate to an instance of a user-specified object category (e.g., a *chair*). At each timestep t, the agent receives egocentric observations I_t (i.e., RGB-D images) and its pose ω_t, which includes its location and orientation. Based on this sensory and positional input, the agent selects an action $a_t \in \mathcal{A}$, where $\mathcal{A}$ includes `move_forward`, `turn_left`, `turn_right`, and `stop`. The navigation episode terminates either when the agent issues the `stop` action or after a maximum of T steps. An episode is deemed successful if the agent issues `stop` with the target object visible and within a threshold distance.

4.2 Navigation Overview

In line with previous works [9, 40, 72, 73, 74, 75], we incrementally build a local semantic map, but via scene segmentation instead of single-frame understanding. Our trained generative flow model then completes the partial map by generating the full semantic distribution, especially for unobserved areas. This distribution guides the agent to likely goal object locations, followed by deterministic local navigation. An overview of the navigation pipeline is illustrated in Fig. 2 (a).

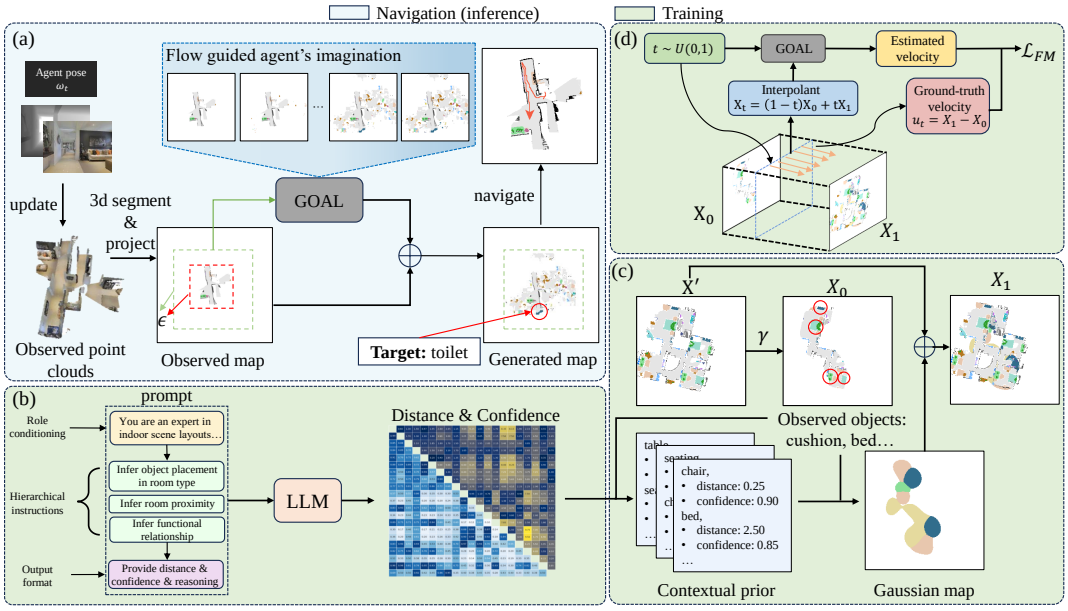


Figure 2: Overview of GOAL framework, with navigation (inference) in blue and training in green. (a) shows the navigation pipeline where the agent imagines future maps using a flow-guided model. (b) illustrates how we prompt a LLM with hierarchical instructions to generate contextual priors (for full prompt and response see Appendix E). (c) visualizes how we use LLM priors to construct data-dependent couplings. (d) demonstrates how the flow model is trained using these couplings through interpolated velocity supervision.

4.3 Generative Flow Models For Agent's Imagination

For training GOAL, we first prompt an LLM to obtain contextual priors between objects (Fig.2(b)), use these priors to construct dependent couplings as training samples (Fig.2(c)), and then train the model using standard interpolation scheme (Fig. 2(d)). We detail each step below.

Prompting LLM for contextual prior. Due to the scarcity of large-scale, densely annotated indoor scene datasets, models often struggle to generalize beyond seen environments. This is largely attributable to the limited diversity of contextual object arrangements in existing data. Here, we describe how we solve the problem by enriching training signals with commonsense knowledge about object co-occurrence, extracted from LLMs.

Specifically, we prompt LLM using various modern prompting techniques such as Chain-of-Thought (CoT) [58], Role Conditioning [44], Few-shot prompting [5], to generate the common distances between different objects $\mathcal{D} = \{d_{ij}\}_{i,j=1}^{N_c}$ and its confidences on each response $\mathcal{C} = \{c_{ij}\}_{i,j=1}^{N_c}$ (Fig. 2(b)). Then, given the partially observed semantic maps, we cluster the connected grids with the same semantic labels into objects $\mathcal{O} = \{o_i\}_{i=1}^{N}$, where N is the number of observed objects. For each observed object o_i, we refer to the LLM responses for its likely co-occurring objects using preset distance threshold τ_d and confidence threshold τ_c. Other objects o_j that either has a distance to the observed object d_{ij} within τ_d or a confidence score c_{ij} larger than τ_c will be considered as co-occurring candidates. Formally, co-occurring candidates set for object o_i is defined as:

$$\mathcal{N}(o_i) = \{o_j \mid (d_{ij} \leq \tau_d) \vee (c_{ij} \geq \tau_c)\}. \tag{5}$$

During training, every clustered observed object will randomly choose co-occurring object among their candidates set $\mathcal{N}(o_i)$ to enrich semantic context and avoid model collapse (In practice, rather

than strictly random selection, we actually prioritize the objects in the intersection of candidates set of multiple observed objects, to take account for a functional cluster with more than 2 objects.). Intuitively, if an object is expected to co-occur near an observed object, but has not yet been observed, it is likely located in the unobserved region of the scene. Thus we connect the clustered objects' centers with their nearest frontiers (the spatial boundaries between known free space and unexplored regions, by definition), and the centroid of the co-occurring objects will be on this connected line, offset by a distance d_{ij}. Specifically:

$$\boldsymbol{\mu_j} = \boldsymbol{\mu_i} + d_{ij}\boldsymbol{v}, \tag{6}$$

where $\boldsymbol{\mu_i}$ and $\boldsymbol{\mu_j}$ are the centroid of observed object o_i and predicted centroid of co-occurring object o_j, and $\boldsymbol{v}$ is a unit vector pointing from o_i to its nearest frontier grid. Then the confidence score c_{ij} is converted to a standard deviation via linear transformation:

$$\sigma_{ij} = \sigma_{\min} c_{ij} + \sigma_{\max}(1 - c_{ij}), \tag{7}$$

where $\sigma_{\min}$ and $\sigma_{\max}$ are hyperparameters representing the lower and upper bounds of standard deviation, respectively. With the computed deviation and centroid, we model the co-occurring objects as two-dimensional Gaussian distributions, which are added across all observed objects:

$$p_{\text{LLM}}^{(j)} = \sum_{o_i \in \mathcal{O}} \frac{1}{2\pi\sigma_{ij}^{(1)}\sigma_{ij}^{(2)}} \exp\left\{ -\frac{1}{2}\left[\left(\frac{x - \boldsymbol{\mu_j^{(1)}}}{\sigma_{ij}^{(1)}} \right)^2 + \left(\frac{y - \boldsymbol{\mu_j^{(2)}}}{\sigma_{ij}^{(2)}} \right)^2 \right] \right\}, \; o_j \sim U(\mathcal{N}(o_i)), \tag{8}$$

where we take $\sigma_{ij}^{(1)} = \sigma_{ij}^{(2)} = \sigma_{ij}$, resulting in an isotropic Gaussian over the semantic map. Notation $U(X)$ refers to uniform distribution, which is used for random sampling. $p_{\text{LLM}}^{(j)}$ is then the LLM prior distribution reflecting the commonsense spatial expectations of object o_j derived from LLMs. The final LLM prior is stacked across all channels:

$$p_{\text{LLM}} = \left[p_{\text{LLM}}^{(1)}, p_{\text{LLM}}^{(2)}, \ldots, p_{\text{LLM}}^{(N_c)} \right], \tag{9}$$

which has the same shape as the input semantic maps and is used to guide the flow model toward plausible object arrangements in unobserved regions of the scene during training.

Building data-dependent couplings with LLM-derived supervision. Following standard practice in diffusion and score-based models [21, 53], the source distribution is typically set to a standard Gaussian $\mathcal{N}(0, I)$, resulting in an independent coupling $\pi_{0,1}(X_0, X_1) = p(X_0)q(X_1)$. However, we find this strategy will complicate the model architecture with additional conditioning mechanism. Instead, we directly couple the partial semantic map with the LLM-enhanced target, eliminating the need for conditioning mechanisms like cross-attention and yielding more consistent and effective generation for navigation. Below, we describe how this dependent coupling is constructed.

During training, we have access to the full-scene semantic map X'. Following [40], we employ the Fast Marching Method (FMM) [49] to simulate a realistic navigation trajectory by planning a path between two randomly sampled points on the map. The visible region along this path serves as a binary mask γ, representing the observed area. To incorporate stochasticity and avoid model collapse, we still inject Gaussian noise with relatively small deviation $\Delta\sigma$ into the unobserved regions of the scene. Specifically, the source semantic map is defined as:

$$X_0 = \gamma \odot X' + \overline{\gamma} \odot \mathcal{N}(0, \Delta\sigma^2), \tag{10}$$

where $\odot$ denotes the Hadamard product, and $\overline{\gamma} = 1 - \gamma$ denotes the complement of the visibility mask. The target semantic map X_1 is constructed by adding LLM-derived priors to the unobserved regions of the full ground-truth map:

$$X_1 = \lambda\, \overline{\gamma} \odot p_{\text{LLM}} + X', \tag{11}$$

where p_{LLM} is derived from Eq. 8 and Eq. 9, and λ is a hyperparameter to control the prior strength. Notably, this formulation results in a data-dependent coupling between X_0 and X_1, since both maps are conditioned on the same ground-truth map X' and share the same visibility mask γ. As a result, the joint distribution $\pi_{0,1}(X_0, X_1)$ described in Sec. 3 is no longer factorizable into independent marginals $p(X_0)q(X_1)$, but instead captures a strong, structured relationship between source and target data.

Training. Given the constructed data-dependent couplings, we adopt the Optimal Transport (OT) displacement interpolant as our interpolation scheme (see Eq. 1), defined as:

$$X_t = (1-t)X_0 + tX_1. \tag{12}$$

For the Bregman Divergence term in Eq. 3, we use the Euclidean distance, resulting in the training objective being a Mean Squared Error (MSE) loss:

$$\mathcal{L}_{FM} = \mathbb{E}_{t, X_t \sim p_t}\left[||\dot{X}_t - u_\theta(X_t, t)||_2^2\right], \tag{13}$$

where the ground-truth velocity field is given by $\dot{X}_t = X_1 - X_0$, derived from the linear interpolation in Eq. 12. For details of the training pipeline, refer to the Appendix B for pseudocode.

4.4 ObjectNav with GOAL

In this subsection, we detail the process of building and preprocessing the semantic map to serve as input to GOAL. followed by how the agent uses the output of it to guide exploration and take effective actions.

Semantic map construction via 3D scene understanding. We construct a semantic map by transforming RGB-D observations into 3D point clouds, segmenting them, and projecting the results to a top-down view, as detailed below. As described in Sec. 4.1, the agent receives RGB-D observations I_t and its pose ω_t at each timestep t, along with the camera intrinsic matrix P inherently available. The observations I_t are first back-projected to point clouds by aligning the RGB values and depth values of each pixel. Then egocentric-to-geocentric transformation is conducted using ω_t and P. The registered point clouds will be segmented at once to take account of historical context, geometric structure, and achieve scene-level perception. Though modern architectures such as the Point Transformer series [61, 62, 76] have demonstrated strong performance in 3D scene understanding, we adopt a Sparse Convolutional network [18], the 3D counterpart of standard 2D CNN-based segmentation models, to reduce potential confounding effects from using highly expressive architectures. After segmentation, points are then projected to bird-eye view to form a local map $M_t \in \mathbb{R}^{(N_c+2)\times h \times w}$, where N_c represents the number of semantic categories, and the additional 2 dimensions correspond to obstacles and free space, respectively (we denotes $M'_t \in \mathbb{R}^{N_c \times h \times w}$ to 'semantic map').

Agent's imagination With GOAL. To guide navigation toward the goal, the agent uses a generative model to imagine unobserved regions of the scene and select the grid with the highest probability to be the target as long-term way-point. Given the semantic map M'_t at time t, we first compute the minimum bounding box of the observed area and crop a surrounding region scaled by a factor ϵ. This factor controls the imagination's spatial extent: too small may limit foresight, while too large may reduce reliability. The cropped region is resized to a fixed shape $m'_t \in \mathbb{R}^{n \times L \times L}$ (with $L = 256$), where unobserved areas are injected with Gaussian noise similar to Eq.10. This map serves as the source sample X_0 in Eq.12 and is passed to the generative flow model, which applies iterative Euler steps (Eq. 4) to generate a complete semantic map. The output is then resized and merged back into the original map, resulting in the imagined full map $\hat{M}_t$. Finally, the grid cell with the highest value in the target object channel c_g is selected as the long-term waypoint g_t:

$$g_t = \arg\max_{(h,w)} \hat{M}_{t(h,w)}^{(c_g)}. \tag{14}$$

Navigation policy. After determining the waypoint $g_t = (x_t, y_t)$, we follow prior works [9, 40, 75] by implementing a local planner based on the Fast Marching Method (FMM) [49], which computes the shortest path from the agent's current position to the waypoint using the occupancy channels of the semantic map. Unlike previous approaches that select waypoints at fixed intervals, we adaptively sample a new waypoint only when the agent is either too close to or too far from the previous one. This strategy reduces computational overhead while maintaining effective path planning.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate our method, **GOAL**, on the validation sets of two standard ObjectNav benchmarks: Gibson [63], Matterport3D (MP3D) [8]. Additionally, we perform transfer experiments

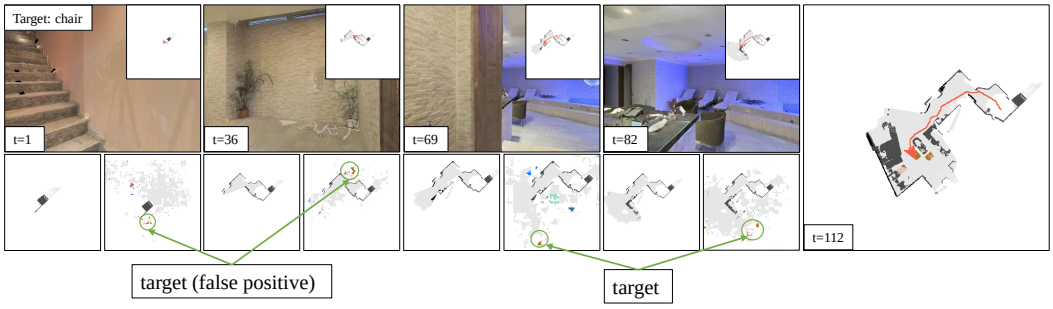


Figure 3: Visualization of navigation with GOAL on MP3D (val). The top row shows RGB observations and agent trajectories; the bottom row displays the observed semantic maps and generated full-scene maps.

Table 1: Transfer experiments results on HM3D. We compare the SR, SPL, and DTS of state-of-the-art methods in different settings and training set. * denotes our implementation using model weights and codes from official repositories, with improvement using scene segmentation.

Method	**Train set**	**LLM usage**		**HM3D**		
		Training	Inference	SR ↑	SPL ↑	DTS ↓
ZSON [34]	HM3D	✗	✗	25.5	12.6	–
PixNav [6]	HM3D	✗	✓	37.9	20.5	–
ESC [77]	–	–	✓	39.2	22.3	–
VoroNav [60]	–	–	✓	42.0	26.0	–
PONI* [40]	MP3D	✗	✗	41.8	20.1	4.63
GOAL w/o LLM	MP3D	✗	✗	47.6	22.5	4.14
GOAL	MP3D	✓	✗	**48.8**	23.1	**4.11**

by training on MP3D and evaluating on HM3D. For Gibson, we follow the tiny-split protocol from [9, 40], using 25 training and 5 validation scenes, with 1,000 validation episodes covering 6 target object categories. For MP3D, we use the standard Habitat simulator setting [35, 39, 55], which includes 56 training and 11 validation scenes, 2,195 validation episodes, and 21 target categories. For HM3D, we use only the 20 validation scenes, with 6 goal categories and 2,000 validation episodes for transfer evaluation. For details of target object categories, please refer to Appendix C.2.

Evaluation metrics. We adopt three standard metrics for evaluating the navigation performance. **SR** (Success Rate) indicates the proportion of success episodes. **SPL** (Success weighted by Path Length) represents the success rate of episodes weighted by path length, measuring the efficiency of navigation. **DTS** (Distance To Goal) is the distance to the goal at the end of the episode. For mathematical expression of these metrics, please refer to Appendix C.1.

Implementation details. For training of GOAL, we sample 400K sub-maps for training on each dataset. We implement GOAL based on DiT [37] models. We use AdamW [24] optimizer with a base learning rate of 1.5e-4, warmed up for 2 epochs, and applied cosine decay after that. The model is trained for 25 epochs, and exponential moving average (EMA) is used with a decay of 0.999 during training. We trained the GOAL model on 4 NVIDIA RTX 4090 GPUs with a batch size of 64 per GPU. For details of training scene segmentation module, please refer to Appendix.C.5.

5.2 Evaluation Results

Visualization of navigation with GOAL. Figure 3 illustrates an example episode where the agent is tasked with finding a chair. Initially, the agent is misled by a false positive prediction due to limited observations. As the agent explores and uncovers more of the environment, the model refines its prediction, effectively guiding exploration and ultimately leading agent to correctly identify and navigate to the target (rightmost column). Additional visualizations demonstrating generation quality and diversity are provided in Appendix F.

Table 2: Comparison between using dependent and independent couplings (Gaussian source). Navigation performance is reported in MP3D and Inference time is tested using an NVIDIA RTX 4090 GPU. The external knowledge is distilled from ChatGLM.

Base PDF	**Number of Parameters**	**Memory Usage**	**GFLOPs**	**Inference time**	**Navigation (MP3D)**		
					SR ↑	SPL ↑	DTS ↓
$x_0 \sim \mathcal{N}(0, I)$	149.92M	693.86 MB	29.34	17.13 ms	39.0	14.0	5.16
$x_0 \sim \rho(X', \gamma)$	**138.76M**	**562.57 MB**	**24.12**	**12.38 ms**	**41.5**	**15.5**	**4.85**

Table 3: Ablation study of individual components on MP3D. 'LP' refers to distillation of LLM priors; 'SS' indicates scene segmentation.

ID	**Modules**		**Navigation (MP3D)**		
	SS	LP	SR ↑	SPL ↑	DTS ↓
1			32.4	11.7	5.25
2	✓		38.8	14.6	5.08
3	✓	✓	**41.7**	**15.5**	**4.84**

Table 4: Effectiveness of model variants. Navigation performance is tested in MP3D.

Model Variants	**LLM**	**Navigation (MP3D)**		
		SR ↑	SPL ↑	DTS ↓
DiT-B	ChatGLM	41.5	15.5	4.85
DiT-L	ChatGLM	40.4	14.9	5.01
DiT-B	Deepseek	40.9	15.0	4.89
DiT-L	Deepseek	40.3	14.7	4.97
DiT-B	ChatGPT	**41.7**	**15.5**	**4.84**
DiT-L	ChatGPT	40.5	15.0	5.09

Generalizability of GOAL. We assess GOAL's generalization ability by training it on MP3D and directly evaluating on the HM3D dataset. We compare its performance against state-of-the-art methods across various training settings. As shown in Tab. 1, GOAL significantly outperforms prior methods, including those that heavily rely on LLMs or are trained directly on HM3D, demonstrating strong generalization capabilities. Notably, even the generative flow model without LLM supervision achieves competitive transfer performance, highlighting the benefits of generative modeling and diverse generation.

Effectiveness of data-dependent couplings. We compare bridging data-dependent couplings described in this paper and the traditional cross-attention method in DiT[37]. As shown in Tab. 2, building dependent couplings yields better navigation performance while simplifying model architecture and reducing inference time. Note that to condition the flow model on the partially observed semantic map, we introduce extra convolutional encoder to downsample the partial maps and then feed the patches to cross-attention layers of each DiT blocks, as an expressive mechanism to process and fuse the complex semantic map.

Ablation study on LLM prior and scene segmentation components. As shown in Tab. 3, scene segmentation alone significantly improves performance by enabling more consistent and complete scene understanding, already establishing a strong baseline (row 2). Notably, further integrating the LLM prior on top of this strong foundation yields a substantial additional gain (row 3), despite the typical difficulty of improving over high-performing baselines. This highlights the effectiveness of the LLM prior, and suggests that our bridging scheme between partial maps and full semantic distributions requires a reliable understanding of the observed scene.

Effectiveness of Flow Matching. Since GOAL shares the objective of inferring unobserved scene semantics with prior approaches [17, 25, 75], it is essential to compare the Flow Matching algorithm we adopt with these baseline prediction methods. As SGM [75] is the most recent approach in this line of work, we compare Flow Matching [26] with the masked autoencoder (MAE) [19] used in SGM, replacing our proposed scene segmentation module with the widely adopted RedNet [23] for a fair

Table 5: Comparison between Flow Matching (FM) and Masked Autoencoder (MAE) for semantics imagination. Models are trained on MP3D and evaluated on different datasets.

Alg.	**Eval. Dataset**	**SR ↑**	**SPL ↑**	**DTS ↓**
MAE	MP3D	31.9	**11.71**	5.31
FM	MP3D	**32.4**	11.67	**5.25**
MAE	HM3D	32.1	**14.7**	4.85
FM	HM3D	**35.9**	14.69	**4.65**

Table 6: Object-goal navigation results on Gibson and MP3D. We compare the SR, SPL and DTS of state-of-the-art methods in different settings. For SemExp [9], L2M [17] and Stubborn [32], we report results from [74]. For SSCNav [25], we report results from [75]. '-' under *Training* indicates zero-shot methods that do not require any training. '_' means the second best results.

Method	**Venues**	**LLM usage**		**Gibson**			**MP3D**		
		Training	Inference	SR ↑	SPL ↑	DTS ↓	SR ↑	SPL ↑	DTS ↓
Semexp [9]	NeurIPS 20	✗	✗	71.1	39.6	1.39	28.3	10.9	6.06
SSC-Nav [25]	ICRA 21	✗	✗	–	–	–	27.1	11.2	5.71
PONI [40]	CVPR 22	✗	✗	73.6	41.0	1.25	31.8	12.1	5.10
L2M [17]	ICLR 22	✗	✗	–	–	–	32.1	11.0	5.12
Stubborn [32]	IROS 22	✗	✗	–	–	–	31.2	13.5	5.01
CoW [16]	CVPR 23	–	✗	–	–	–	7.4	3.7	–
3DAware [74]	CVPR 23	✗	✗	74.5	42.1	1.16	34.0	14.6	**4.78**
T-Diff [72]	NeurIPS 24	✗	✗	79.6	**44.9**	1.00	39.6	15.2	5.16
L3MVN [71]	IROS 23	–	✓	76.1	37.7	1.10	34.9	14.5	–
SG-Nav [69]	NeurIPS 24	–	✓	–	–	–	40.2	<u>16.0</u>	–
UniGoal [70]	CVPR 25	–	✓	–	–	–	<u>41.0</u>	**16.4**	–
SGM [75]	CVPR 24	✓	✓	78.0	44.0	1.11	37.7	14.7	4.93
GOAL	Proposed	✓	✗	**83.5**	<u>44.2</u>	**0.83**	**41.7**	15.5	<u>4.84</u>

comparison. As shown in Table 5, while MAE and Flow Matching achieve comparable performance on MP3D, Flow Matching demonstrates stronger generalization in the transfer setting, where models are trained on MP3D and evaluated on HM3D.

Effectiveness of model variants and LLMs. As shown in Tab.4, increasing model complexity (e.g., using DiT-L) yields little to no performance gain, and may even degrade performance compared to the base model (DiT-B). We hypothesize this is due to overfitting, as complex models tend to overfit when trained on limited and less diverse data, even in a generative setting. To assess the impact of LLMs in GOAL, we compare three models: ChatGLM-4-plus [14], DeepSeek-R1 [12], and GPT-4 [36]. As shown in Tab. 4, the overall navigation performance remains similar across LLMs (though we observe significant per-scene variance). Since a number of prior works [69, 75, 77] also report minimal differences between LLM variants, we hypothesize that current evaluation datasets are too small and biased to reliably reflect the effects of LLM choice.

Comparison with related works. We evaluate the performance of our method, GOAL, on the ObjectNav task by comparing it with relevant baselines, categorized by their use of LLMs during training and inference. SemExp [9] first introduced semantic reasoning into object-goal navigation. PONI [40] improves it using supervised learning to predict two potential functions, avoiding the inefficiencies of reinforcement learning. L2M [17] and SSC-Nav [25] improve navigation by predicting the full top-down semantic map. SGM [75] further advances these approaches using the MAE algorithm [19] to train a ViT [13] model in a self-supervised manner for full-scene semantic imagination. To enhance generation capabilities, SGM also prompts an LLM for contextual object information, but requires it during both training and inference, introducing additional complexity via a cross-attention mechanism. In contrast, methods like L3MVN [71], SG-Nav [69], and UniGoal [70] rely solely on LLMs to achieve zero-shot performance. While these approaches reduce training requirements, they often suffer from drawbacks such as API latency and excessive memory consumption.

As shown in Tab. 6. GOAL consistently outperforms the current state-of-the-art approach [75], across all metrics and datasets. Although the improvement in SPL is relatively small, this can be attributed to the modified local policy we employ, described in Sec. 4.4. Specifically, the agent only updates its long-term goal when it becomes significantly closer to or farther from the previous one. As a result, the agent's ability to correct false positive predictions is limited, even when new observations provide better guidance, leading to relatively long path length. And we stress that this limitation can be solved by applying modern techniques [15, 29, 33, 51, 52] to reach faster inference for generative flow and adopt the general policy of changing long-term goal at fixed interval. We leave this for future work as these techniques will introduce complex formulation.

6 Discussions and Limitations

There are several technical limitations and potential further improvements to consider.

First, we aim to introduce the generative flow matching algorithm into ObjectNav task, so our implementations adhers to the core theory in initial paper. However, recent advances in FM for natural image generation suggests some powerful training techniques to be used, such as time discretizations and time shift, as well as techniques that reduce the Number of Evaluations (NFEs) of flow matching[15, 29, 33, 51, 52]. Incorporating these techniques could enhance training stability and sampling efficiency, potentially yielding more robust navigation policies.

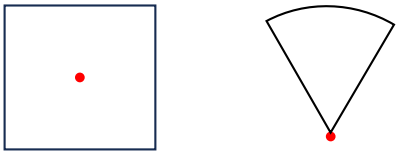


Figure 4: Comparison between the simulated visible area and actual visible area given the agent position (red dot). The left shows the simulated mask adopted by [40] and our work, while the right shows the actual mask, revealing a substantial gap.

Second, our method operates on a fixed-dimensional semantic map, with channels corresponding to a predefined set of object categories. This design inherently restricts generalization to open-vocabulary or zero-shot settings, where novel object classes may appear at test time.

Finally, we follow the setting of [40] by computing a path between two randomly selected points on the ground-truth semantic map, where a rectangular region centered at points along the path is treated as the visible area to generate the partially observed maps for training GOAL. However, we found a significant gap between these simulated partial observations and those encountered during real navigation. In practice, the agent's visible area is fan-shaped rather than rectangular (See Fig. 4). Designing training samples that better simulate actual agent observations could substantially improve model performance.

7 Conclusion

In this work, we propose GOAL, a generative flow model that distills rich contextual priors from LLM into its training supervision. We show that data-dependent couplings between partially observed maps and full semantic distributions significantly improve generation quality, outperforming traditional independent couplings commonly used in natural image domains. Additionally, we introduce a scene segmentation module to enhance holistic, geometry-aware, and temporally consistent scene understanding. Experimental results on large-scale datasets Gibson and MP3D validate the effectiveness of GOAL, while cross-dataset transfer experiments on HM3D further highlight its strong generalization capabilities.

8 Acknowledgments

This work was supported partially by NSFC(U21A20471,92470202, 62206315), by the National Key Research and Development Program of China (2023YFA1008503), Guangdong NSF Project (No. 2023B1515040025, No. 2024A1515010101), Guangzhou Basic and Applied Basic Research Scheme (No. 2024A04J4067).

References

[1] Michael S. Albergo, Mark Goldstein, Nicholas Matthew Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, 2024.

[2] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, 2025.

[3] Maxim Berman, Amal Rannen Triki, and Matthew B. Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018.

[4] Léon Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade - Second Edition*. 2012.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[6] Wenzhe Cai, Siyuan Huang, Guangran Cheng, Yuxing Long, Peng Gao, Changyin Sun, and Hao Dong. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, 2024.

[7] Tommaso Campari, Paolo Eccher, Luciano Serafini, and Lamberto Ballan. Exploiting scene-specific features for object goal navigation. In *Computer Vision – ECCV 2020 Workshops*, 2020.

[8] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from RGB-D data in indoor environments. In *2017 International Conference on 3D Vision, 3DV 2017, Qingdao, China, October 10-12, 2017*, 2017.

[9] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[10] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023.

[11] Pointcept Contributors. Pointcept: A codebase for point cloud perception research. `https://github.com/Pointcept/Pointcept`, 2023.

[12] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, 2025.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[14] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. GLM: general language model pretraining with autoregressive blank infilling. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 2022.

[15] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[16] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, 2023.

[17] Georgios Georgakis, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, and Kostas Daniilidis. Learning to map for active semantic goal navigation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[18] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018.

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 2022.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017.

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[22] Steeven Janny, Hervé Poirier, Leonid Antsfeld, Guillaume Bono, Gianluca Monaci, Boris Chidlovskii, Francesco Giuliari, Alessio Del Bue, and Christian Wolf. Reasoning in visual navigation of end-to-end trained agents: A dynamical systems approach. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, 2025.

[23] Jindong Jiang, Lunan Zheng, Fei Luo, and Zhijun Zhang. Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. *arXiv preprint arXiv:1806.01054*, 2018.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[25] Yiqing Liang, Boyuan Chen, and Shuran Song. Sscnav: Confidence-aware semantic scene completion for visual semantic navigation. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, 2021.

[26] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[27] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A. Theodorou, Weili Nie, and Anima Anandkumar. I^2sb: Image-to-image schrödinger bridge. In *International Conference on Machine Learning,ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, 2023.

[28] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[29] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[30] Renjie Lu, Jingke Meng, and Wei-Shi Zheng. PRET: planning with directed fidelity trajectory for vision and language navigation. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXVI*, 2024.

[31] Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 2022.

[32] Haokuan Luo, Albert Yue, Zhang-Wei Hong, and Pulkit Agrawal. Stubborn: A strong baseline for indoor object navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2022, Kyoto, Japan, October 23-27, 2022*, 2022.

[33] Weijian Luo. A comprehensive survey on knowledge distillation of diffusion models. *CoRR*, 2023.

[34] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. ZSON: zero-shot object-goal navigation using multimodal goal embeddings. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[35] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[36] OpenAI. GPT-4 technical report. *CoRR*, 2023.

[37] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, 2023.

[38] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018.

[39] Xavi Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Ruslan Partsey, Jimmy Yang, Ruta Desai, Alexander William Clegg, Michal Hlavac, Tiffany Min, Theo Gervet, Vladimír Vondruš, Vincent-Pierre Berges, John Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars and robots, 2023.

[40] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. PONI: potential functions for objectgoal navigation with interaction-free learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 2022.

[41] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M. Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3D): 1000 large-scale 3d environments for embodied AI. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021.

[42] Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 2022.

[43] Hao Ren, Yiming Zeng, Zetong Bi, Zhaoliang Wan, Junlong Huang, and Hui Cheng. Prior does matter: Visual navigation via denoising diffusion bridge models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, 2025.

[44] Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. In *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama Japan, May 8-13, 2021, Extended Abstracts*, 2021.

[45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 2022.

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, 2015.

[47] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *SIGGRAPH '22: Special Interest Group on Computer Graphics and Interactive Techniques Conference, Vancouver, BC, Canada, August 7 - 11, 2022*, 2022.

[48] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

[49] James A. Sethian. Fast marching methods. *SIAM Rev.*, 1999.

[50] Kunal Pratap Singh, Luca Weihs, Alvaro Herrasti, Jonghyun Choi, Aniruddha Kembhavi, and Roozbeh Mottaghi. Ask4help: Learning to leverage an expert for embodied tasks. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[51] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*, 2024.

[52] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, 2023.

[53] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

[54] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[55] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[56] Huilin Tian, Jingke Meng, Wei-Shi Zheng, Yuan-Ming Li, Junkai Yan, and Yunong Zhang. Loc4plan: Locating before planning for outdoor vision and language navigation. In *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, 2024.

[57] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Trans. Mach. Learn. Res.*, 2024.

[58] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[59] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019.

[60] Pengying Wu, Yao Mu, Bingxian Wu, Yi Hou, Ji Ma, Shanghang Zhang, and Chang Liu. Voronav: Voronoi-based zero-shot object navigation with large language model. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, 2024.

[61] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer V3: simpler, faster, stronger. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, 2024.

[62] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer V2: grouped vector attention and partition-based pooling. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[63] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018.

[64] Karmesh Yadav, Ram Ramrakhya, Arjun Majumdar, Vincent-Pierre Berges, Sachit Kuhar, Dhruv Batra, Alexei Baevski, and Oleksandr Maksymets. Offline visual representation learning for embodied navigation. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

[65] Jiaojie Yan, Qieshi Zhang, Jun Cheng, Ziliang Ren, Tian Li, and Zhuo Yang. Indoor target-driven visual navigation based on spatial semantic information. In *2022 IEEE International Conference on Image Processing, ICIP 2022, Bordeaux, France, 16-19 October 2022*, 2022.

[66] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[67] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary tasks and exploration enable objectgoal navigation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, 2021.

[68] Xin Ye and Yezhou Yang. Efficient robotic object search via HIEM: hierarchical policy learning with intrinsic-extrinsic modeling. *IEEE Robotics Autom. Lett.*, 2021.

[69] Hang Yin, Xiuwei Xu, Zhenyu Wu andJie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. In *Advances in Neural Information Processing Systems 38: Annual Conferenceon Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver,BC, Canada, December 10 - 15, 2024*, 2024.

[70] Hang Yin, Xiuwei Xu, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. Unigoal: Towards universal zero-shot goal-oriented navigation. *arXiv preprint arXiv:2503.10630*, 2025.

[71] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3MVN: leveraging large language models for visual target navigation. In *IROS*, 2023.

[72] Xinyao Yu, Sixian Zhang, Xinhang Song, Xiaorong Qin, and Shuqiang Jiang. Trajectory diffusion for objectgoal navigation. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[73] Albert J. Zhai and Shenlong Wang. PEANUT: predicting and navigating to unseen targets. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, 2023.

[74] Jiazhao Zhang, Liu Dai, Fanpeng Meng, Qingnan Fan, Xuelin Chen, Kai Xu, and He Wang. 3d-aware object goal navigation via simultaneous exploration and identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, 2023.

[75] Sixian Zhang, Xinyao Yu, Xinhang Song, Xiaohan Wang, and Shuqiang Jiang. Imagine before go: Self-supervised generative map for object goal navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, 2024.

[76] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, 2021.

[77] Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. ESC: exploration with soft commonsense constraints for zero-shot object navigation. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, 2023.

[78] Yu Zhou, Dian Zheng, Qijie Mo, Renjie Lu, Kun-Yu Lin, and Wei-Shi Zheng. Decoupled distillation to erase: A general unlearning method for any class-centric tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, 2025.

[79] Minzhao Zhu, Binglei Zhao, and Tao Kong. Navigating to objects in unseen environments by distance prediction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2022, Kyoto, Japan, October 23-27, 2022*, 2022.

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: We clearly demonstrate our claims in both 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 discussed the limitations in Sec 6.

   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: This paper do 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: We illustrated all experimental setup and details for reproduction in Sec. 5.1 and Appendix C.

   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?

Answer: [Yes]

Justification: We will release the code upon the decision 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: We detailed all essential information for experimental setting in Sec. 5.1 and Appendix C.

   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: [Yes]

   Justification: We report results across five random seeds in Appendix D.2 to capture variability.

   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: We reported the compute resources in Sec. 5.1 and Appendix C.

   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: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   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: [NA]

    Justification: Our proposed method is only used for academic research currently.

    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.

- 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: The paper poses no such risks.

    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: All the datasets used in this paper are publicly available, and we cite them all.

    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: The paper does not release 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: The paper does not involve crowdsourcing nor 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: The paper does not involve crowdsourcing nor 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.

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: [Yes]

Justification: We incorporate LLMs into supervision of our model, which is one of the main claim of this paper. We describe the specific use in Sec. 4.3.

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.

Appendix

We provide additional information about our method and experiments in the appendix. Below is a summary of the sections:

- Appendix A presents the strict mathematical formulation of Conditional Flow Matching (CFM), with additional notation omitted in the main text for simplicity.
- Appendix B describes the training algorithm with pseudocodes.
- Appendix C outlines the experimental setup and further implementation details.
- Appendix D provides more experimental results and analysis.
- Appendix E includes the prompts used for querying the LLM and its responses.
- Appendix F provides additional visualizations of our results.

Appendix A Strict Formulation of CFM

The Flow Matching loss defined in Equation 3 is, in practice, not directly solvable (since the target velocity $\dot{X}_t$ is not tractable). Throughout the paper, we refer to the Conditional Flow Matching (CFM) algorithm without explicitly including the conditioning variables in the notation, for simplicity. In this section, we present the strict and complete formulation of the Conditional Flow Matching algorithm.

Conditioning design in CFM vary, examples include conditioning on source sample X_0, the target sample X_1, or joint coupling (X_0, X_1), and they are essentially equivalent. We exhibit the general formulation (conditioned on any random variable Z).

Following [57], suppose that *marginal probability path* $p_t(x)$ is a mixture of probability paths $p_t(x|z)$ that vary with some conditioning variable z:

$$p_t(x) = \int p_t(x \mid z) q(z) dz. \tag{15}$$

The *marginal velocity field*, which generates this marginal probability path, is given by averaging the conditional velocity field $u_t(x \mid z)$ across the condition z:

$$u_t(x) = \int u_t(x \mid z) p_{Z|t}(z \mid x) dz = \mathbb{E}\left[u_t(X_t|Z) \mid X_t = x\right] \tag{16}$$

Conditional Flow Matching Loss is then defined to be:

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{t, Z, X_t \sim p_{t|Z}(\cdot|Z)} \left[D\left(u_t(X_t \mid Z), u_t^\theta(X_t)\right)\right]. \tag{17}$$

In practice, we actually apply data-coupling conditioning (namely $Z = (X_0, X_1)$) as shown in Eq. 12, $X_t \sim p_t$ is given by a linear combination of X_0 and X_1, and velocity field is hence given by $X_1 - X_0$. Then the loss in Eq.13 can indeed solve the Flow matching problem introduced in Sec. 3.

Appendix B Algorithm

Algorithm 1 outlines the procedure for constructing data-dependent couplings and training the GOAL model. The corresponding sampling process is detailed in Algorithm 2. We stress that the sampling requires no additional pre-processing, making it a plug-and-play module.

Appendix C Experimental setup and implementation details

C.1 Metrics

As mentioned in the main text, we select **SR**, **SPL**, **DTS** as the evaluation metrics for ObjectNav performance. Following we give their mathematical formulation and explaination.

Algorithm 1 Training algorithm for **GOAL**

Input: dataset $\mathcal{X}$, initial model parameter θ, learning rate η, distance matrix $\mathcal{D}$ and confidence matrix $\mathcal{C}$ from LLM responses.
repeat
  sample $X' \sim \mathcal{X}$
  Randomly sample two points g_1, g_2 on the grid
  Compute visible mask γ by planning a path from g_1 to g_2
  $X_0 \leftarrow \gamma \odot X'$
  Cluster X_0 into observed objects $\{o_i\}_{i=1}^{N}$
  Initialize p_{LLM} as a zero vector with the same shape as X'
  for each o_i in $\{o_i\}_{i=1}^{N}$ **do**
    $p_{\text{LLM}}^{(j)} \leftarrow p_{\text{LLM}}^{(j)} + \texttt{ComputeLLMPrior}(o_i, \mathcal{D}, \mathcal{C})$ ▷ See Eq. 8
    $X_1^{(j)} \leftarrow X_1^{(j)} + \lambda\overline{\gamma} \odot p_{\text{LLM}}^{(j)}$
  end for
  $X_0 \leftarrow X_0 + \overline{\gamma} \odot \mathcal{N}(0, \Delta\sigma^2)$
  $t \sim \mathcal{U}[0, 1]$
  $X_t \leftarrow (1-t)X_0 + tX_1$
  $\hat{u}_t \leftarrow u_\theta(X_t, t)$
  $\mathcal{L} \leftarrow \text{MSE}(\hat{u}_t, X_1 - X_0)$
  $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}$
until convergence

Algorithm 2 Sampling algorithm for **GOAL**

Input: trained GOAL model u_θ, partially observed semantic map M, number of steps n, standard deviation $\Delta\sigma$.
$\gamma \leftarrow$ mask of empty area of M
$M \leftarrow M + \overline{\gamma} \odot \mathcal{N}(0, \Delta\sigma^2)$
for $k \leftarrow 1$ to n **do**
  $t_k \leftarrow \frac{k}{n}$
  $\Delta M \leftarrow u_\theta(M, t_k)$
  $M \leftarrow M + \frac{1}{n}\Delta M$
end for
return M

Success Rate (**SR**) represents the agent's accuracy in reaching the user-specified object goal, where higher values indicates better performance:

$$SR = \frac{1}{N}\sum_{i=1}^{N} S_i, \tag{18}$$

where N is the number of validation episodes and S_i indicates whether the i-th episode is successful.

Success weighted by Path Length (**SPL**) evaluates success relative to the shortest path, normalized by the actual path length agent takes:

$$SPL = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i^*}{\max(l_i, l_i^*)}, \tag{19}$$

where l_i^* denotes the shortest path length and l_i is the actual path length agent takes.

Distance To Goal (**DTS**) measure the distance of agent towards the target object when the episode ends:

$$DTS = \frac{1}{N}\sum_{i=1}^{N} \max(L_{i,g} - \xi, 0), \tag{20}$$

where $L_{i,g}$ is the distance between agent and goal, and ξ is the success threshold (0.1 meter in MP3D).

Table 7: Chosen object categories in Gibson [63], MP3D [8] and HM3D [41]

Dataset	Training	Evaluating
Gibson	*chair, couch, potted plant, bed, toilet, dining-table, tv, oven, sink, refrigerator, book, clock, vase, cup, bottle*	*chair, couch, tv, bed, toilet, potted plant*
MP3D	*chair, table, picture, cabinet, cushion, sofa, bed, chest of drawers, plant, sink, toilet, stool, towel, tv monitor, shower, bathtub, counter, fireplace, gym equipment, seating, clothes*	*chair, table, picture, cabinet, cushion, sofa, bed, chest of drawers, plant, sink, toilet, stool, towel, tv monitor, shower, bathtub, counter, fireplace, gym equipment, seating, clothes*
HM3D	–	*chair, couch, potted plant, bed, toilet, tv*

C.2 Object categories

Following the setup of previous works [40, 72, 75], we adopt 15 categories for training and 6 categories for validation in the Gibson dataset, and 21 categories for both training and validation in the MP3D dataset. Additionally, validation episodes in the HM3D dataset contain 6 categories. The adopted categories are detailed in Table 7.

C.3 Trivial hyper-parameters

There are a number of trivial hyper-parameters which are not tuned, we detail the choices we adopt intuitively in Tab. 8 for better reproduction.

Hyper-Parameters	Values
τ_d (Eq. 5)	2.5
τ_c (Eq. 5)	0.85
$\sigma_{\min}$ (Eq. 7)	20
$\sigma_{\max}$ (Eq. 7)	50
λ (Eq. 8)	1500
$\Delta\sigma$ (Eq. 8)	0.01

Table 8: Values for trivial hyper-parameters

C.4 Memory and Time Cost Analysis

Our method involves maintaining point cloud representations and performing scene segmentation, along with generative modeling for exploration. These components naturally raise concerns about memory and computation overhead.

Unlike semantic maps, whose memory usage is fixed by tensor grid dimensions, point cloud memory consumption varies significantly across scenes and episodes, depending on how much of the environment has been explored. As a result, a scene-independent comparison is difficult. In practice, we observed that running 6 parallel threads on a 24GB NVIDIA RTX 3090 consumes approximately 22GB of memory. For comparison, PONI [40] reports around 20GB under similar settings, indicating our approach adds roughly 350MB per thread.

Time cost also varies across scenes and depends on the agent's waypoint update frequency. Since GOAL is only invoked when the agent is either close to or far from the previous target, inference is sparse and input-dependent. Empirically, running 6 threads in parallel yields an average FPS between 1.2 and 1.8, while PONI ranges from 1.5 to 3.5. Despite this, we consider the trade-off worthwhile, as our method achieves over 30% improvement in success rate (SR), the primary metric of interest.

C.5 Training of scene segmentation model

We train a Sparse UNet [18, 46] with the assistance of Pointcept [11] codebase, using a sum of weighted Cross-Entropy loss $\mathcal{L}_{CE}$ and Lovász-Softmax loss [3] $\mathcal{L}_{LZ}$ as training objectives:

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{LZ}. \tag{21}$$

For optimizer, we simply select Stochastic gradient descent (SGD) [4] with a base learning rate of 0.05, momentum of 0.9 and a weight decay of 0.0001. Additionally, the first 5% of the training steps are used for warm-up, and the learning rate smoothly decays using cosine annealing over the remaining 95% of the training steps. We train the scene segmentation model on 2 NVIDIA RTX 4090 GPUs with a total batch size of 64.

Appendix D More experimental results

D.1 Hyper-parameters tuning

We tune two key hyperparameters: the expansion ratio of the observed map ϵ, and the number of Euler steps n used during generation. The effect of varying the number of Euler steps n is shown in Fig. 5. Navigation performance generally improves with more steps, saturating around $n = 96$. As discussed in Sec.5.2, while the overall performance across different LLMs is comparable, each exhibits a distinct preference for the expansion ratio ϵ. Therefore, we tune ϵ individually for each LLM, as shown in Fig.6. We observe that the flow model distilled with external knowledge from ChatGPT produces more reliable semantic distributions with a larger expansion ratio.

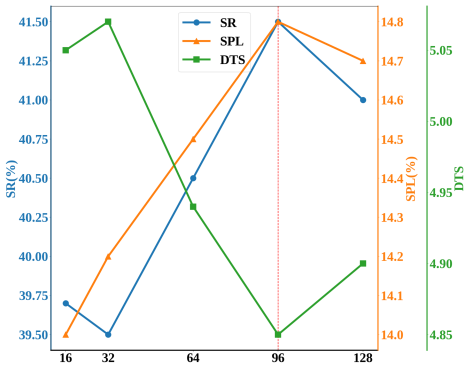
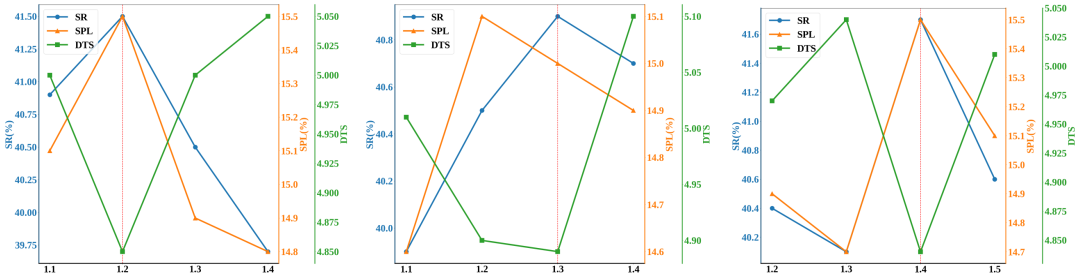


Figure 5: Effect of the number of Euler steps n on navigation performance.



(a) ChatGLM (b) Deepseek (c) ChatGPT

Figure 6: Tuning curve for hyper-parameter ϵ across different LLMs.

D.2 Evaluation Variability and Error Analysis

While probabilistic generative schemes offer improved generalization, they naturally introduce concerns about evaluation stability due to inherent stochasticity. To assess the robustness of our model, we conduct additional evaluations on the MP3D dataset using different random seeds (42, 75, 100, 123, 3407). The resulting success rates are 41.0%, 41.6%, and 41.7%, 41.6%, 42.2% respectively, yielding an average success rate of $41.6\% \pm 0.4\%$. These results indicate that our flow model can reliably capture the semantic distribution, despite the stochasticity introduced by the generative process. We report the result with seed 100 in the main text, as all experiments and hyperparameter tuning were conducted under this setting.

Appendix E Prompts and LLM responses

E.1 Prompts

We provide an example of prompts that query LLMs for objects contextual information. Specifically, we first condition the LLM with its role in the system prompt. Next, we provide contextual information, followed by a chain-of-thought [58] style of hierarchical prompting, which involves scenes, rooms, and objects. Moreover, for each step, we also provide some positive and negative examples to serve as few-shot learning samples. In addition to the distance-confidence pair, we further require the LLMs to output a brief reasoning for their responses. An example prompt is as follows:

Prompt

System Prompt: You are an expert in indoor scene layouts, with strong reasoning skills regarding object co-occurrence. Your task is to infer the typical distances between different object types in indoor environments, considering both object placement patterns and functional relationships. Output your answers in a clear, structured format with a confidence level that reflects the uncertainty of each estimate.

User Prompt: In indoor scenes, object layouts typically follow certain patterns; for example, chairs are usually placed around a table but are unlikely to be near a toilet. Suppose you are analyzing a large-scale indoor scene (e.g., a house with multiple rooms such as living rooms, bedrooms, bathrooms, etc.). Given the following list of objects: [chair, table, ...] and a specific *central object* in it, your tasks are as follows, step-by-step:

1. **Inferring Object Placement:** Determine where each object is typically placed in the scene based on common indoor layouts. For example:
   - Toilet → bathroom
2. **Room Proximity:** Identify the typical rooms surrounding the room where the *central object* is placed. For instance:
   - Chairs (living room) → nearby: kitchen, hallway
   - Bed (bedroom) → nearby: bathroom, closet
3. **Functional Relationships:** Consider the functional relationships between the *central object* and other objects:
   - Chair & table: strong (dining/study areas)
   - Chair & toilet: weak (different functional zones)
4. **Distance Inference:** Based on the inferred scene layout and functional relationships, estimate the typical distance between the central object and other objects. For example, if the *central object* is 'chair', then:
   - table to chair: close (functional pair)
   - Chair to closet: relatively farther away (possibly the same room but with weak functional relationship)
   - Chair to toilet: large distance (different functional zones)
5. **Provide Distance and Confidence:** Given *central object* = ..., for each object provide:
   - Distance: **continuous float value in meters**
   - Confidence: **score between 0 (uncertain) and 1 (certain)**
   - Reasoning: A brief explanation of how the distance and confidence were determined, considering the room layout and common object placements

E.2 Responses

Few examples for LLMs response are as follows:

Response
central object: table

- *quried object*: chair
  - distance: **0.5**
  - confidence: **0.95**
  - reasoning:
    Chairs are commonly placed directly around a table, especially in dining areas or meeting rooms, making the distance very short and the placement highly predictable.

- *quried object*: picuture
  - distance: **1.5**
  - confidence: **0.8**
  - reasoning:
    Pictures are often hung on walls near tables, especially in living or dining rooms, but not as close as chairs, leading to a moderate distance with good confidence.
- *quried object*: counter
  - distance: **4.5**
  - confidence: **0.65**
  - reasoning:
    Counters are often found in kitchens, which may be adjacent to dining areas but not directly next to tables, leading to a moderate distance with moderate confidence.
- *quried object*: bathtub
  - distance: **8.0**
  - confidence: **0.5**
  - reasoning:
    Bathtubs are also found in bathrooms, separate from areas where tables are commonly placed, resulting in a large distance with low confidence.

.....

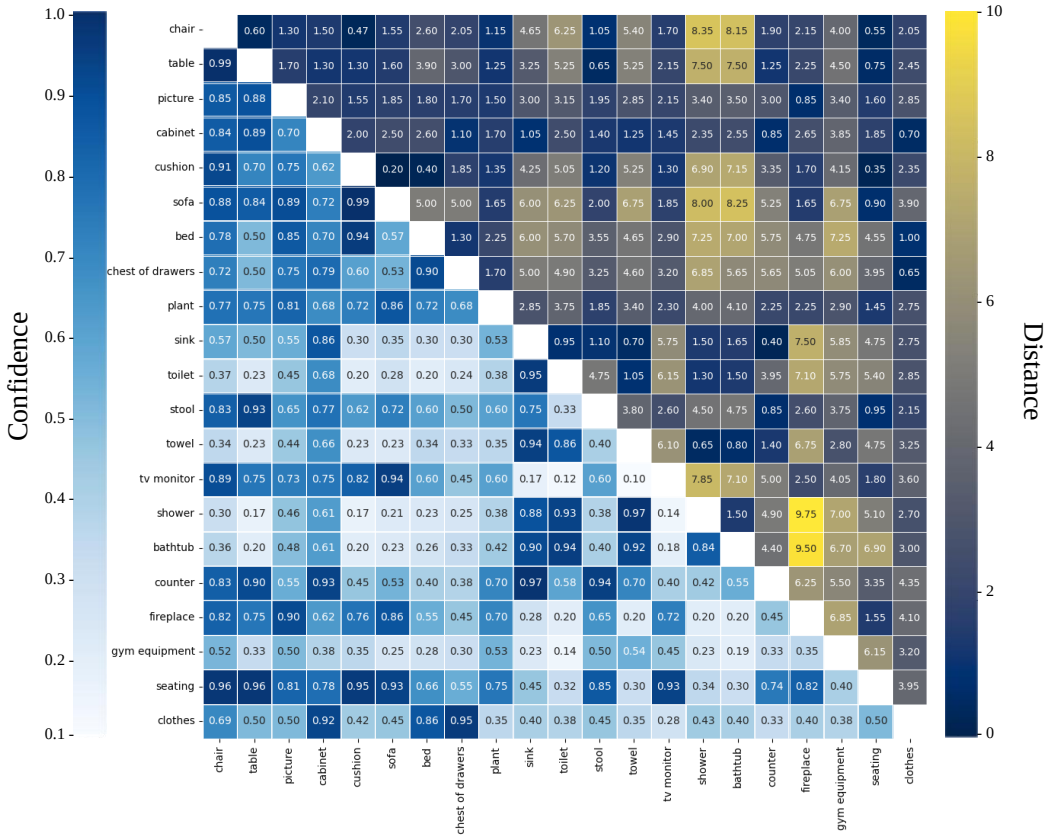


Figure 7: Structured distance matrix $\mathcal{D}$ and confidence matrix $\mathcal{C}$ obtained from GPT-4 for MP3D. We visualize the upper triangle of the distance matrix and the lower triangle of the confidence matrix within the same figure for compactness and clarity

Since the distance between objects should be bidirectional, we take the average of the symmetric distances and confidences, resulting in two symmetric matrices. Structured matrix representation of LLM response from GPT-4 is shown in Fig. 7.

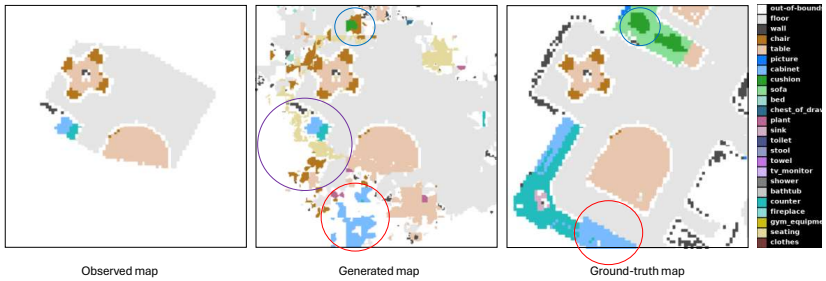


Figure 8: Example for semantic map generated by generative flow model. It successfully generates objects in GT semantic map like cabinet (highlighted in red circle) and cushion (highlighted in blue circle). Moreover, it also generate objects not in the ground-truth map but indeed reasonable, such as seating and chair (highlighted in purple circle) behind the table, which can improve the capability of generalization.

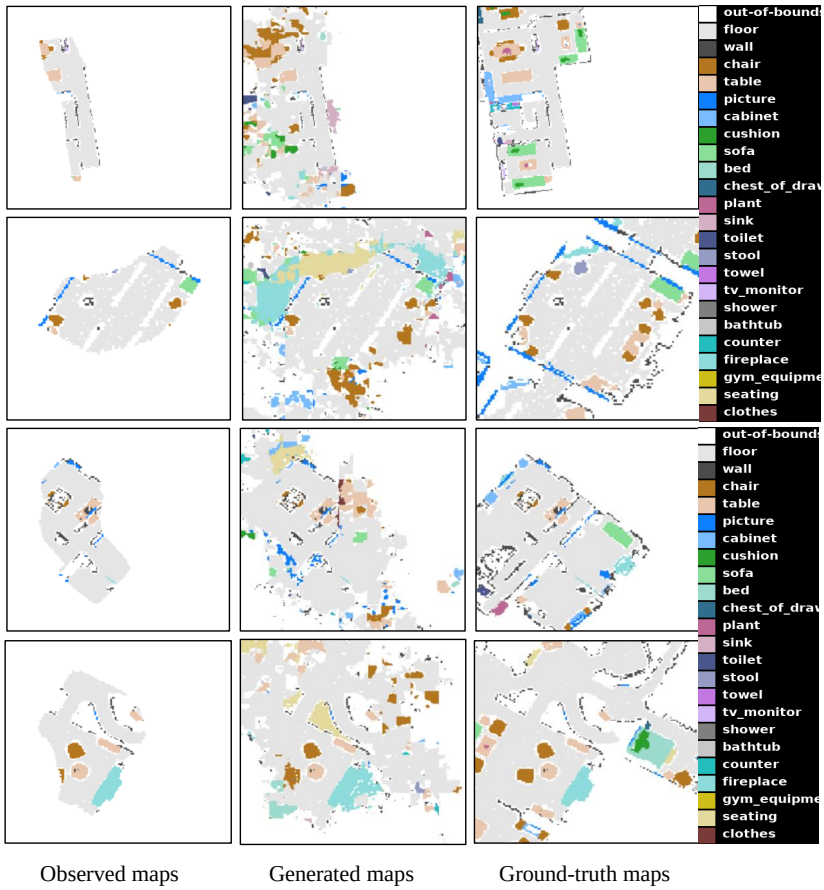


Figure 9: More visualizations for outputs of generative flow model.

Appendix F More visualizations and analysis

F.1 visualizations for generative flow

In this subsection, we present visualizations and analyses of the outputs produced by our generative flow model. Generating a complete semantic distribution of a scene is an inherently challenging task and unlikely to be perfectly accurate. While the visualizations may not appear flawless, they significantly contribute to navigation performance. We begin with an illustrative example accompanied by detailed analysis (see Fig.8), followed by additional qualitative results (Fig.9). In this paper, we stress that indoor scene semantics can vary greatly, and multiple plausible distributions may exist given the same partial semantic map. To capture this diversity, GOAL adopts a probabilistic generation scheme, which enhances the model's ability to generalize to unseen environments. We showcase a sample of this generative diversity in Fig. 10. While we highlight examples where objects are generated near observed ones to provide intuitive evaluation, the diversity applies to the full semantic distribution and is not limited to individual object placements.

F.2 visualizations for scene segmentation

In Fig. 11, we additionally present few visualizations for comparison between scene segmentation proposed in this paper and image segmentation traditionally adopted in ObjectNav.

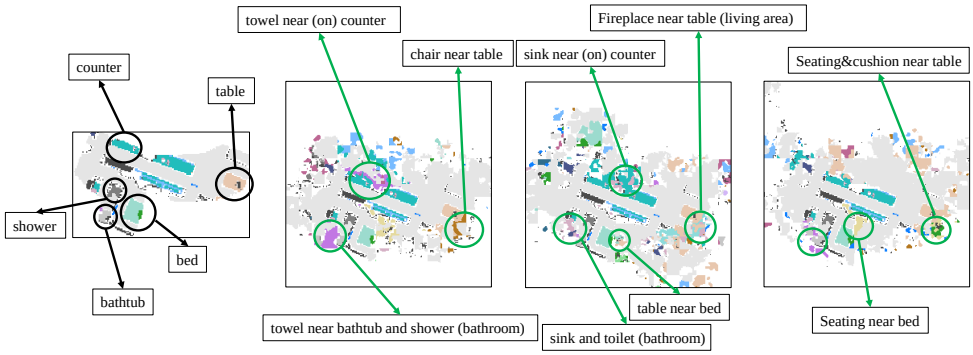


Figure 10: Visualization of generation diversity. Given a single partial map (left most), GOAL can generate multiple plausible full semantic distribution (right three).

Figure 11: Comparison between built semantic maps using image segmentation models and scene segmentation models.