Function-centric Bayesian Network for Zero-Shot Object Goal Navigation

Supplementary Material

Table 1. The selected categories of scenes, objects, and function groups in our FBN.

	Category
Scenes	Living room, Kitchen, Bedroom, Bathroom, Dining room, Corridor
Objects	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, Microwave, Oven, Toaster, Refrigerator
Function groups	Resting, Storage, Working, Cooking, Cleaning, Entertaining, Sleeping, Dining, Toileting, Grooming, Exercising, Bathing, Playing, Reading, Dressing, Decorating

Table 2. Depiction sentences to each scene category.

Scenes	Description
Living room	A living room designed for family gatherings and leisure activities, includes sofa, armchair, television, coffee table, decorative artwork, and potted plant.
Kitchen	A kitchen designed for cooking, includes refrigerator, stove, sink, countertop, cabinet, and microwave.
Bedroom	A bedroom designed for rest and sleeping, includes bed, nightstand and lamp.
Bathroom	A bathroom designed for hygiene, includes shower, bathtub, sink, toilet, and towel.
Dining room	A dining room designed for meals and gatherings, includes dining table, chair, and sideboard.
Corridor	A corridor designed for connecting spaces, includes passage, walls, lighting and door.

1. Category Definition

In our work, the probabilistic semantic map is constructed by using an open-vocabulary detection model to support open-vocabulary object targets. As shown in Tab. 1, we consider common indoor scene and object categories as text prompts for the open-vocabulary detection model (i.e., Grounding DINO [1]). Note that these categories can be freely defined.

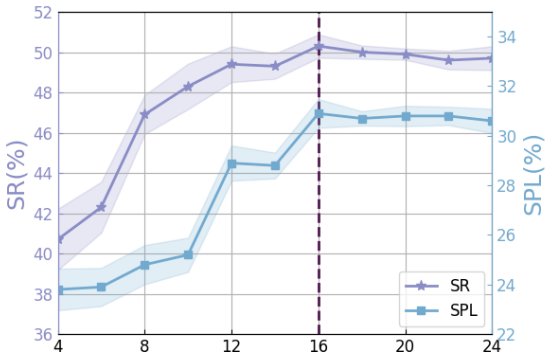

Figure 1. Effect of the number of functional groups on navigation performance in HM3D. Experiments are based on the FBN-Llama model, with light-colored shading indicating variance.

Moreover, since scene categories are inherently abstract concepts, we refine their representation by expanding them into descriptive sentences to provide a more comprehensive depiction, as shown in Tab. 2. These enriched descriptions, which replace singular scene categories, are subsequently employed as text prompts for Grounding DINO, leading to improved detection performance.

Additionally, our function-centric bayesian network (FBN) is constructed by considering not only the aforementioned scene and object categories but also a series of functional groups, as detailed in Tab. 1.

2. Ablation on Functional Group

We conduct ablations to evaluate the impact of the number of functional group categories on navigation performance, as shown in Fig. 2. The results indicate that as the number of function group categories increases, navigation performance improves progressively. However, beyond a certain threshold (e.g., 16 categories), further increases yield negligible performance gains.

On the other hand, an increase in the number of function group categories also elevates the complexity of SCG. Additionally, the number of prompts required for LLMs to construct SCG increases, leading to greater overall model complexity.

Table 3. **Real environment experiments.** We compare the navigation performance on SR(%) of our FBN and VLFM [2] in real-world.

Environment	Method	Target					Avg.
		plant	couch	toilet	tv	towel	
Env(a)	VLFM	40.00	60.00	30.00	50.00	20.00	40.00
	FBN	60.00	80.00	40.00	50.00	50.00	56.00
Env(b)	VLFM	30.00	50.00	50.00	30.00	30.00	38.00
	FBN	60.00	70.00	50.00	40.00	50.00	54.00
Env(c)	VLFM	50.00	50.00	40.00	40.00	10.00	38.00
	FBN	70.00	50.00	60.00	60.00	40.00	56.00
Avg.	VLFM	40.00	53.33	40.00	40.00	20.00	38.67
	FBN	63.33	66.67	50.00	50.00	46.67	55.33

Considering these findings, we ultimately select 16 functional group categories, as detailed in Tab. 1, to balance performance and model complexity.

3. Real Robot Evaluation

To evaluate the generalization ability of our FBN in real-world, we conduct additional experiments in a realistic environment, as shown in Tab. 3. To construct navigation environment in real world, we utilize a 100-square-meter open space, dividing it into several rooms using movable walls. The rooms are then decorated with common indoor objects. Since both the walls and objects can be freely rearranged, we create three distinct environments to test our method. Five objects are selected as navigation targets. We use the Locobot-wx250s as the agent. Notably, both VLFM and our FBN are zero-shot methods and do not require training. For each target object, the agent is initialized at 10 random starting positions. We evaluate navigation performance using the success rate (SR).

The experimental results demonstrate that our FBN outperforms VLFM in navigation tasks, especially for smaller objects such as towels. We hypothesize that smaller objects are more challenging for detection modules to identify directly, requiring prior knowledge to infer their potential locations. Our FBN constructs causal nexuses to explain and infer object locations, enabling the agent to achieve superior performance. The results from the real-world experiments support the strong generalization ability of our method in real-world scenarios. Additionally, we provide a video demo of real-world navigation in the supplementary material, available in the file `real_world.mp4`.

4. Video Demo

We provide video demonstrations of our method in both simulated and real-world environments, available in the supplementary material as `sim.mp4` and `real_world.mp4`, respectively.

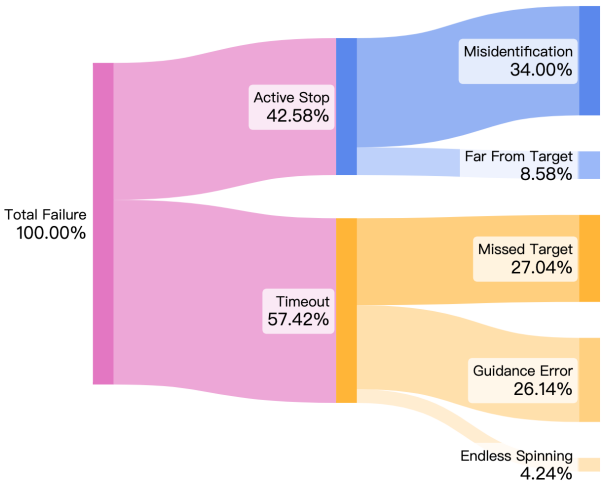

Figure 2. Failure cases of FBN on HM3D. 'Active Stop' refers to the agent stopping autonomously but failing to meet the success criteria, while 'Timeout' indicates the agent failing to find the target within the step limit (500 steps).

These videos showcase real-time RGB and depth views, along with the object and scene layers from the probabilistic semantic map. They also include the probability map of goal occurrences. In the probability map, regions with deeper red shades indicate higher probabilities of the target's presence.

5. Failure Case

We analyze the failure cases of our method and categorize them into two types. The first 'Active Stop' occurs when the agent autonomously stops, believing it has located the target. Of these, 34.00% are due to misidentification caused by detection module errors, while others result from stopping prematurely, failing to meet the target's distance threshold.

The second type 'Timeout' refers to the agent fails to find the target within the step limit. In 27.04% of these cases, the target appears in the agent's trajectory but is missed due to detection errors. Additionally, 4.24% are caused by endless spinning, where the agent gets stuck in corners due to environmental constraints. Furthermore, 26.14% of failures occur because the agent lacks proper guidance and fails to explore the target's region, primarily due to FBN errors. However, this accounts for a smaller portion of failures.

Overall, the majority of failures stem from inaccuracies in object detection, indicating the critical impact of detection module performance on navigation success.

6. Prompt

The complete prompt of CounterfactCoT with GPT-4o is presented below:

```
You are an expert in causal inference. I need to
    analyze the relations of object-function and
    scene-function  within an indoor environment.
     Given two nodes (X, Y) and a subgraph (their
     context), infer step by step, and answer
    whether a relationship exists (1 or 0) and
    its probability.

Guidance: Follow these steps:
Step 1: Identify the variables, like var1 -> var2
    , separated by commas.
Step 2: Determine the relation type. Choices
    include:
(1) object -> function
(2) scene -> function
Step 3: Assume the existence of the edge and
    formulate both the factual and counterfactual
     cases in symbolic form.
Step 4: Infer the probabilities of both the
    factual and counterfactual cases.
Step 5: Compare the probabilities of the factual
    and counterfactual cases and deduce the
    estimand
Step 6: Evaluate the existence (1 or 0) and
    probability.

Refer to the following examples:
Example 1:
Q: X:towel, Y:grooming, Subgraph: bathroom->
    grooming, sink->grooming
Step 1: towel -> grooming, bathroom->grooming,
    sink->grooming
Step 2: object -> function
Step 3: Variables: X: towel, Y: grooming, Z1:
    bathroom, Z2: sink, Fact: P(Y=1|X=1,Z1=1,Z2
    =1), Counterfact: P(Y=1|do(X=0),Z1=1,Z2=1),
    value 1: relation presence, value 0: remove
Step 4: P(Y=1|X=1,Z1=1,Z2=1)=0.90, P(Y=1|do(X=0),
    Z1=1,Z2=1)=0.20,
Step 5: P(Y=1|do(X=1),Z1=1,Z2=1)-P(Y=1|do(X=0),Z1
    =1,Z2=1)=0.70, 0.70>0.2 is high
Step 6: 1, 0.70
A: 1, 0.70

Now, determine the relationship between X and Y.
    Output format must be concise and strictly
    follow the example format:

Q: X: {}, Y: {}, Subgraph: {}
```

References

[1] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55, 2024. 1

[2] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 42–48. IEEE, 2024. 2