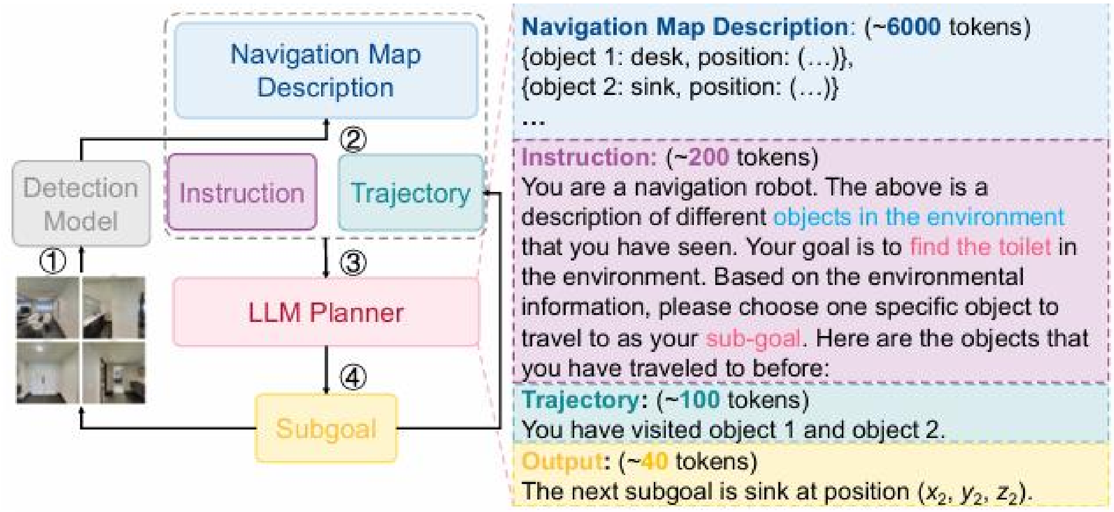
EfficientNav: Towards On-Device Object-Goal Navigation with Navigation Map Caching and Retrieval

Zebin Yang, Sunjian Zheng, Tong Xie, Tianshi Xu, Bo Yu*,
Fan Wang, Jie Tang, Shaoshan Liu, Meng Li*

* Corresponding author. Emails: boyu@cuhk.edu.cn, meng.li@pku.edu.cn

Background

- Object-goal navigation tasks an agent with navigating to locations of specific objects in an unseen environment
- Large language models with memory have been introduced for **long-term planning** in a **zero-shot manner**
- ObjNav works in a step-by-step manner; in each step, the planner chooses a sub-goal for further exploration
- The information of explored areas and visited objects (navigation map), the instruction to find the final goal, and the history trajectory information will be given to the LLM planner



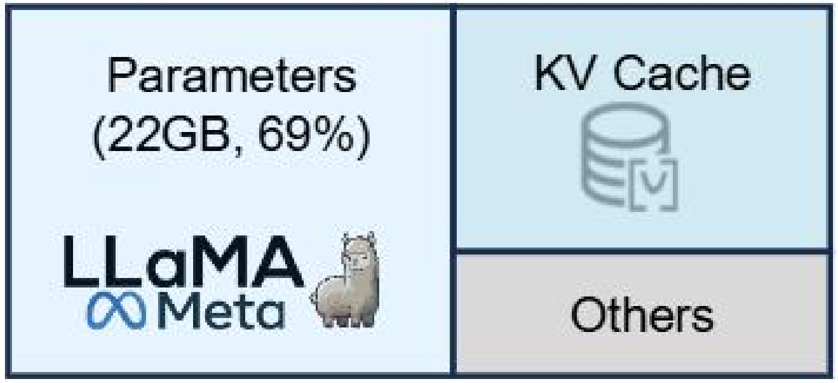
北京大學 PEKING UNIVERSITY

Background

- For better accuracy, existing works use **giant LLMs** (GPT-4, GPT-4V), which must be deployed on online servers
- These cloud offloading methods suffer from **high latency**, **privacy concerns**, and **a heavy reliance on WiFi**
- To overcome this, we optimize the planning process and deploy the whole system on local devices
- However, deploying the ObjNav system on local devices faces challenges because of **tight memory constraints**

Table 1: Comparison with prior methods.

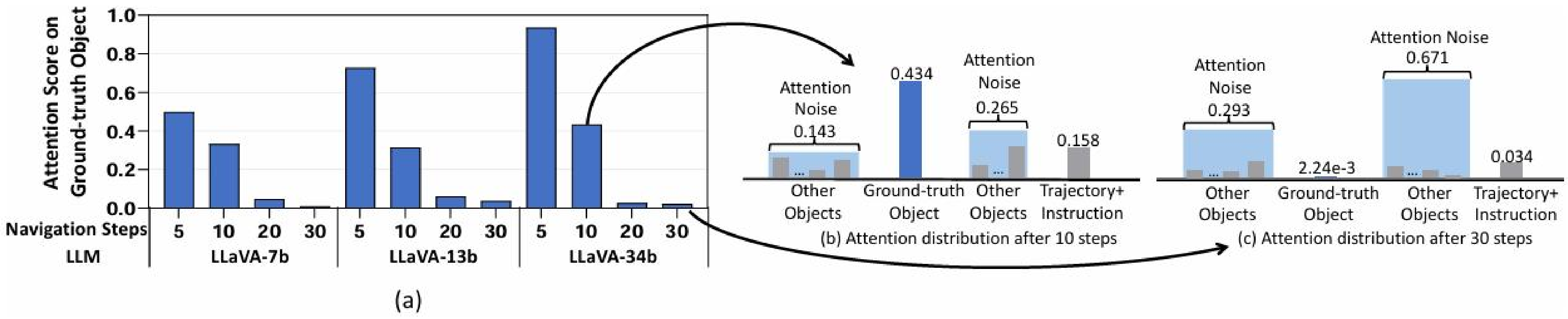
Method	Zero-shot	LLM	On-device Inference	Memory Augmented
ViKiNG [51]	✗	-	✓	✓
NaVid [69]	✗	Vicuna	✓	✗
Skip-SCAR [36]	✗	-	✓	✓
Pixel Navigation [7]	✓	GPT-4	✗	✗
InstructNav [37]	✓	GPT-4V	✗	✓
MapGPT [10]	✓	GPT-4	✗	✓
LFG [50]	✓	GPT-4	✗	✓
EfficientNav (**Ours**)	✓	LLaMA	✓	✓



NVIDIA Jetson Orin: 32GB

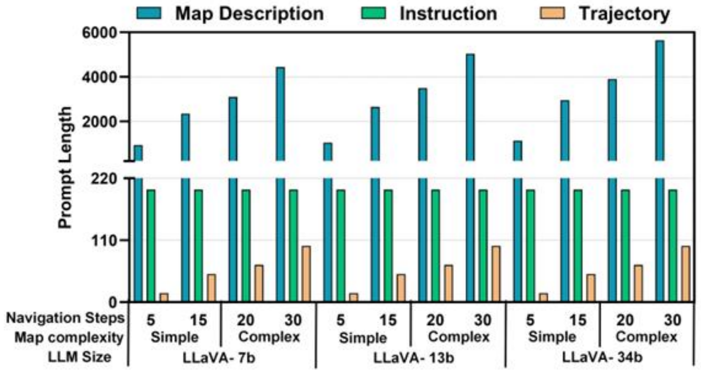
Challenge 1: Model Capacity

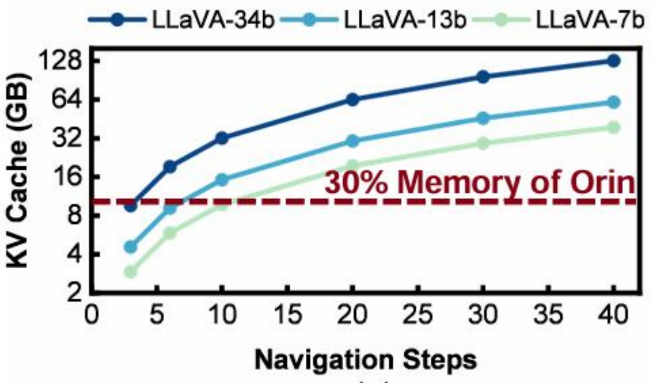
- The tight memory constraint forces to use **smaller LLMs** (e.g., LLaMA-11b), which have poorer model capacity
- In each step, newly detected objects will be added to the navigation map, and environmental information will increase with the exploration process, among which includes **redundant information**
- For smaller LLMs, the redundant information in the map will negatively impact the planning performance

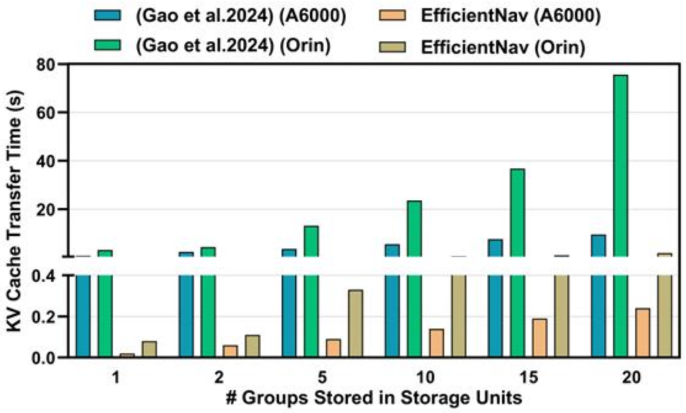


Challenge 2: Memory Capacity

- Environmental information increasing with the exploration process will introduce long prompt, which will introduce long real-time latency because of **high prefilling computation**
- Tight memory constraints of local devices limit the **saving of the KV cache** of the navigation map description
- Traditional methods offload KV cache to CPU, while this introduces large **memory communication** overhead

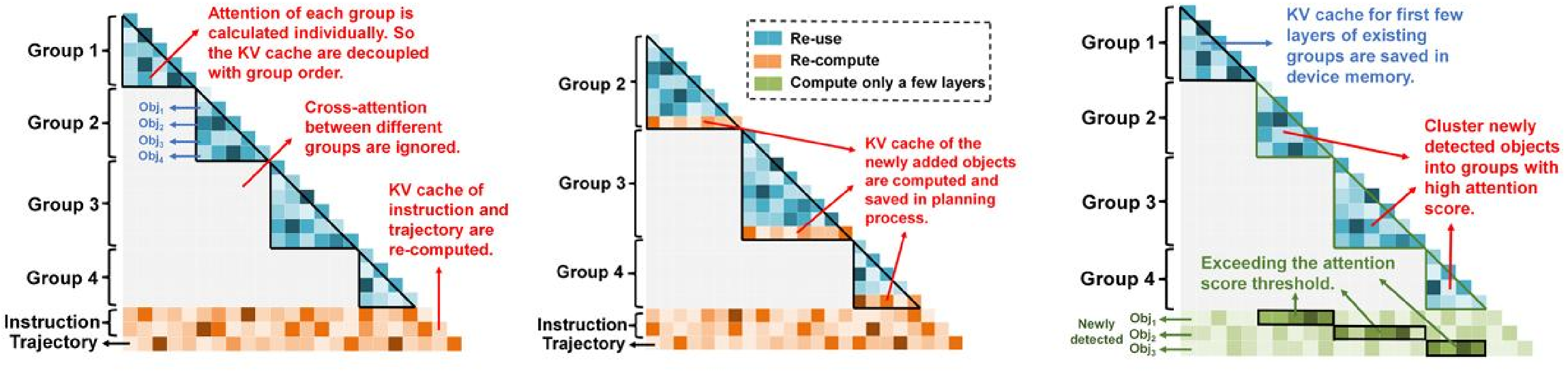






Method

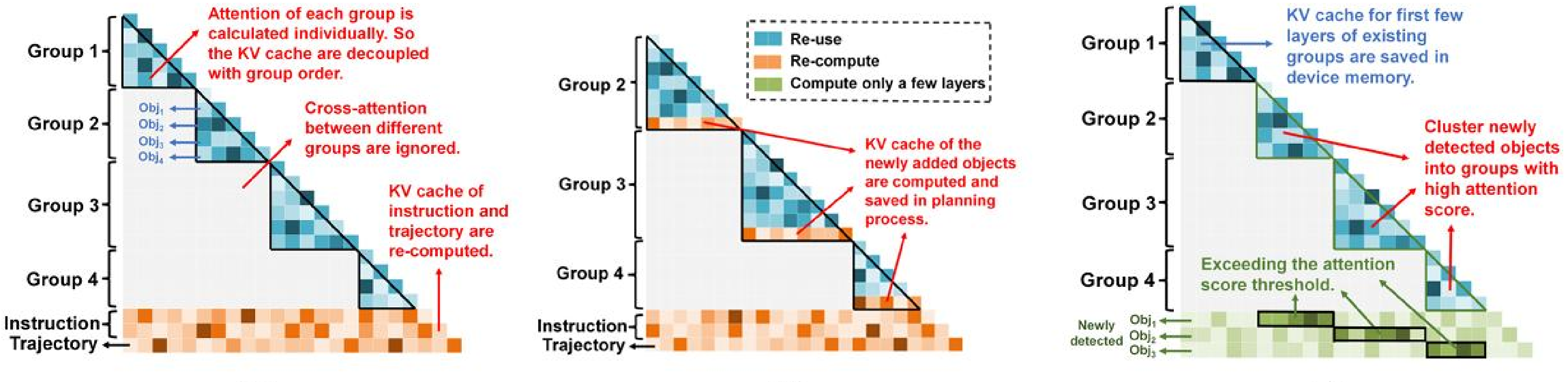
- To meet the memory constraints and improve model performance, we design a novel **navigation map caching and retrieval method**, which can remove redundant information and reduce latency
- However, with different information retrieved, the **prefix of prompt changes**, making the saved KV unusable
- We propose **discrete memory caching** to group memory and calculate the KV cache of each group individually
- This can **decouple the KV cache calculation and memory order,** thus enabling KV reuse



(a) (b) (c)

Method

- However, calculating KV of each group individually will cause **ignorance of cross-attention** between objects
- To avoid performance drop caused by this ignorance, we cluster object information by **object relevance**
- We propose **attention-based memory clustering**, using LLM attention to save related objects into same groups
- If the average attention between a newly detected object and an existing group exceeds a specific threshold, we cluster this object into the group

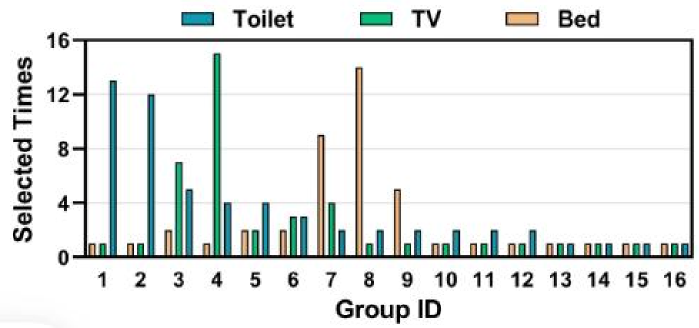


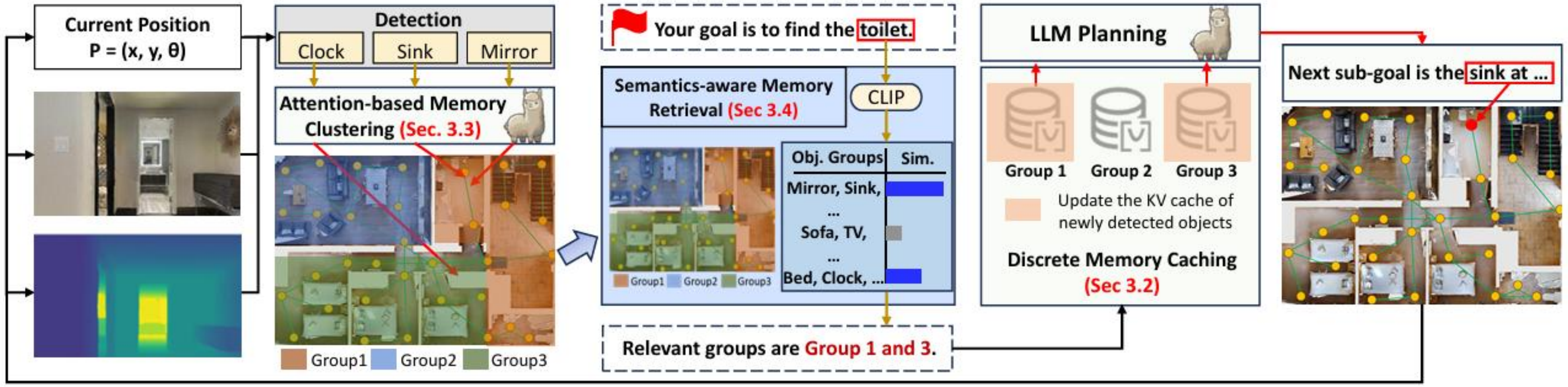
(a) (b) (c)

北京大學 PEKING UNIVERSITY

Method

- To remove redundant information and improve performance, we propose **semantics-aware memory retrieval**
- We observe that with different final goals, the relevance of different groups varies a lot
- In memory retrieval, considering **retrieval efficiency and semantic matching**, we use a pre-trained semantic model CLIP to calculate the relevant probability between the final goal and groups
- To adapt to devices with different memory budgets, we formulate the group selection as a **knapsack problem**





Experiments

PEKING UNIVERSITY 北京大學

- EfficientNav achieves **11.1%** improvement in success rate on HM3D benchmark over GPT-4-based baselines
- EfficientNav demonstrates **6.7×** real-time latency and **4.7×** end-to-end latency reduction over GPT-4 planner

Table 2: SR and SPL comparison.

Method	Zero-shot	LLM	SR	SPL
DD-PPO [59]	✗	-	27.9	14.2
SemExp [9]	✗	-	37.9	18.8
Habitat-web [47]	✗	-	41.5	16.0
OVRL [63]	✗	-	62.0	26.8
ZSON [39]	✓	-	25.5	12.6
PixelNav [7]	✓	GPT-4	37.9	20.5
ESC [73]	✓	-	39.2	22.3
VoroNav [60]	✓	GPT-3.5	42.0	26.0
LLaVA Planner-34b [10]	✓	LLaVA-34b	42.7	21.0
L3MVN [67]	✓	RoBERTa-large	50.4	23.1
InstructNav [37]	✓	GPT-4V	58.0	20.9
LFG [50]	✓	GPT-4	68.9	36.0
EfficientNav-11b	✓	LLaMA3.2-11b	**74.2**	**39.5**
EfficientNav-34b	✓	LLaVA-34b	**80.0**	**41.5**

Table 3: Average latency comparison on A6000.

Method	LLM	RtL	E2EL
GPT-4 Planner [10]	GPT-4	5.80s	59.34s
LLaMA Planner-11b [10]	LLaMA3.2-11b	3.07s	46.40s
vllm [29]	LLaMA3.2-11b	2.27s	39.78s
EfficientNav-11b (**Ours**)	LLaMA3.2-11b	0.35s	12.70s
LLaVA Planner-34b [10]	LLaVA-34b	5.63s	55.32s
vllm [29]	LLaVA-34b	4.43s	47.95s
EfficientNav-34b (**Ours**)	LLaVA-34b	0.87s	12.51s

北京大學 PEKING UNIVERSITY

Conclusion

- To meet the memory constraints and improve model performance, we design a novel **navigation map caching and retrieval method**, which can remove redundant information and reduce real-time latency
- We propose **discrete memory caching** to decouple KV calculation and memory order, thus enabling KV reuse
- We propose **attention-based memory clustering** to recover accuracy drop caused by cross-attention ignorance
- We propose **semantics-aware memory retrieval** to remove redundant information and improve performance
- EfficientNav achieves **11.1%** improvement in success rate on HM3D benchmark over GPT-4-based baselines

Thanks for Listening

EfficientNav: Towards On-Device Object-Goal Navigation with Navigation Map Caching and Retrieval