GridMM: Grid Memory Map for Vision-and-Language Navigation
— Supplementary Material —

Zihan Wang[1,2], Xiangyang Li[1,2], Jiahao Yang[1,2], Yeqi Liu[1,2], Shuqiang Jiang[1,2]
[1]Key Lab of Intelligent Information Processing Laboratory of the Chinese Academy of Sciences (CAS),
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
zihan.wang@vipl.ict.ac.cn, lixiangyang@ict.ac.cn,
{jiahao.yang, yeqi.liu}@vipl.ict.ac.cn, sqjiang@ict.ac.cn
https://github.com/MrZihan/GridMM

A. Datasets

We evaluate our approach in discrete environments (*e.g.*, R2R [2], REVERIE [11], and SOON [15]), and further analyze many characteristics of our approach in continuous environments (*e.g.*, R2R-CE [9] and RxR-CE [10]).

All the benchmarks in discrete environments build upon the Matterport3D environment [3] and contain 90 photo-realistic houses. Each house contains a set of navigable locations, and each location is represented by the corresponding panorama image and GPS coordinates. We adopt the standard split of houses into training, val seen, val unseen, and test splits. Houses in the val seen split are the same as in training, while houses in val unseen and test splits are different from training. All splits in discrete environments are consistent with Chen *et al.* [4].

R2R-CE [9] transfers the discrete paths in R2R dataset to continuous trajectories on the Habitat simulator [13]. RxR-CE [10] transfers the discrete paths in RxR dataset to continuous trajectories on the Habitat simulator [13].

B. Performance in RxR-CE

Method	TL	NE↓	SR↑	SPL↑	nDTW↑	SDTW↑
VLN-CE [9]	7.33	12.1	13.93	11.96	30.86	11.01
CMA [8]	20.04	10.4	24.08	19.07	37.39	18.65
VLNBERT [8]	20.09	10.4	24.85	19.61	37.30	19.05
DUET [4](Ours)	21.48	9.78	29.93	23.12	42.46	25.39
GridMM (Ours)	21.13	**8.42**	**36.26**	**30.14**	**48.17**	**33.65**

Table 1. Evaluation on the test unseen split of RxR-CE dataset.

As shown in Table 1, our GridMM achieves competitive results on longer trajectory navigation such as RxR-CE.

C. Experimental Details

C.1. Training Details

For the REVERIE dataset, we combine the original dataset with augmented data synthesized by DUET [4] to pre-train our model with a batch size of 32 and a learning rate of 5e-5 for 100k iterations, using 3 NVIDIA RTX3090 GPUs. Then we fine-tune it with the batch size of 4 and a learning rate of 1e-5 for 50k iterations on 3 GPUs.

For the SOON dataset, we only use the original data with automatically cleaned object bounding boxes, sharing the same settings with DUET [4]. We pre-train the model with a batch size of 16 and a learning rate of 5e-5 for 40k iterations using 3 GPUs, and then fine-tune it with a batch size of 2 and a learning rate of 5e-5 for 20k iterations on 3 GPUs.

For the R2R dataset, additional augmented data in [7] is used for pre-training following DUET [4]. Using 3 GPUs, we pre-train our model with a batch size of 32 and a learning rate of 5e-5 for 100k iterations. Then we fine-tune it with the batch size of 4 and a learning rate of 1e-5 for 50k iterations on 3 GPUs.

For the R2R-CE dataset, we transfer the model pre-trained on the R2R dataset to continuous environments, and fine-tune it with a batch size of 8 and a learning rate of 1e-5 for 30 epochs using 3 RTX3090 GPUs.

For all the datasets, the best model is selected by SPL on the val unseen split.

C.2. Ablation Details

Top-down semantic map. We follow CM^2 [6] to obtain a 448×448 top-down semantic map. Specifically, we use a pre-trained UNet [12] from CM^2 [6] to produce semantic segmentation of observation images, and then project pixels into a unified top-down semantic map. After dividing the top-down semantic map into multiple patches with a scale of 32×32, a convolution layer is used to encode these patches into embeddings with a hidden size of 768. We take these semantic embeddings as the map features.

Map with object features. A pre-trained detection model VinVL [14] is utilized to detect multiple objects in each view image, and then we take 10 object features with the highest confidence score as substitutes for grid features. For the coordinate of each object, it is obtained via the center point of the bounding box.

D. Analysis of Computational Cost

Referring to [5], we describe how we calculate the number of Floating-point Operations (FLOPs) in VLN models as follows:

1) Matrix multiplication ($A_{m \times k} \times B_{k \times n}$):

$$2mkn \text{ FLOPs}$$

2) 2-layer MLP (sequence length s, increase the hidden size to $4h$ and then reduces it back to h):

$$16sh^2 \text{ FLOPs}$$

3) Self-attention block (sequence length s, hidden size h):

$$4s^2h + 8sh^2 \text{ FLOPs}$$

4) Cross-attention block (query sequence length s, key and value sequence length t, hidden size h):

$$4sh^2 + 4th^2 + 4sth \text{ FLOPs}$$

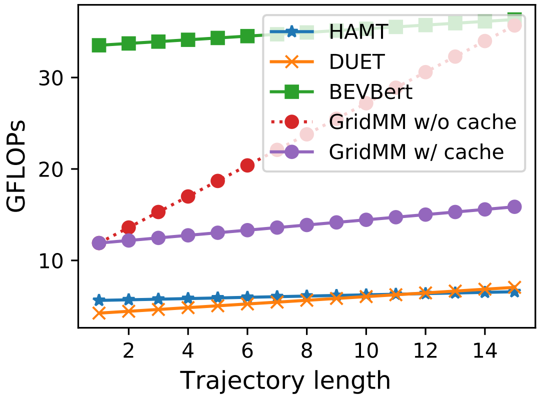

Figure 1. GFLOPs at different trajectory lengths keeping instruction length as 32. The computational cost of visual encoders and text encoders is omitted for a more intuitive comparison.

We calculate GFLOPs (Giga Floating-point Operations) on the R2R dataset, as illustrated in Fig. 1 and Fig. 2. "GridMM w/o cache" denotes that our GridMM updates each cell of the grid map in all navigation steps without any cache. By using the cache (which stores previous results for later use), the computational cost is significantly reduced. For the same grid features in all navigation steps, during updating the cells of the grid map, we only need to recompute the positions of grid features, without recomputing their relevance value in the relevance matrix with the instruction. The reason is that, for Equations (6) and (9), the outputs of $\hat{g}_{t,j} W_1^A$ (where $\hat{g}_{t,j}$ is a part of $\mathcal{M}^{rel}_{t,m,n}$), $\mathcal{W}' W_2^A$ and $W^E \hat{g}_{t,j}$ in all navigation steps is the same and can be cached for reuse. GFLOPs of "GridMM w/ cache" are significantly lower than that of BEVBert [1]. During attention computation, the number of metric map features in BEVBert exceeds 400, introducing a huge computational cost. However, the number of map features in GridMM is less than 200 and they are only used as key and value tokens in cross-attention computation, which greatly reduces the computational cost.

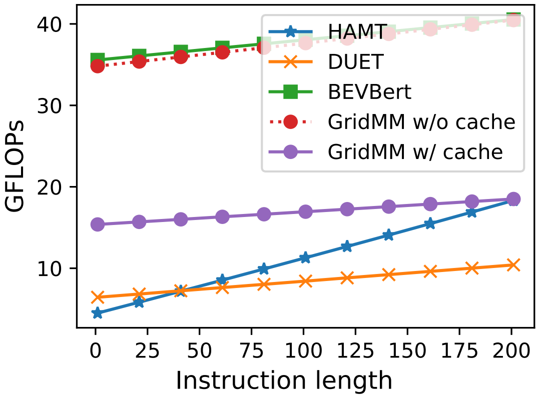

Figure 2. GFLOPs with different instruction lengths keeping trajectory length as 15. The computational cost of visual encoders and text encoders is omitted for a more intuitive comparison.

References

[1] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Topo-metric map pre-training for language-guided navigation. *arXiv preprint arXiv:2212.04385*, 2022.

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683, 2018.

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *3DV*, pages 667–676, 2017.

[4] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *CVPR*, pages 16537–16547, 2022.

[5] Narayanan Deepak, Shoeybi Mohammad, Casper Jared,

LeGresley Patrick, Patwary Mostofa, Korthikanti Vijay, Vainbrand Dmitri, Kashinkunti Prethvi, Bernauer Julie, Catanzaro Bryan, Phanishayee Amar, and Zaharia Matei. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021.

[6] Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *CVPR*, 2022.

[7] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *CVPR*, pages 13137–13146, 2020.

[8] Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *CVPR*, June 2022.

[9] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*, 2020.

[10] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *EMNLP*, pages 4392–4412, 2020.

[11] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *CVPR*, pages 9982–9991, 2020.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, page 234–241, 2015.

[13] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *ICCV*, pages 9339–9347, 2019.

[14] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *CVPR*, pages 5579–5588, 2021.

[15] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *CVPR*, pages 12689–12699, 2021.