Modality-invariant Visual Odometry for Embodied Vision Supplementary Material

Marius Memmel[1]* Roman Bachmann[2] Amir Zamir[2]
[1]University of Washington [2]Swiss Federal Institute of Technology (EPFL)

https://vo-transformer.github.io

A. Dataset

Dataset	`fwd`	`left`	`right`	Collisions	Rollouts	Total
training (VOT	143697	53140	53163	28255	4891	250000
training [12]	150000	49870	50130	27908	9782	250000
validation (VOT)	13990	5542	5468	2809	599	25000
validation [12]	15000	4999	5001	2770	1184	25000

Table A.1. Sample statistics of the training and validation sets for our experiments and ConvNet-baselines (separate [12]).

We collect separate datasets for training joint models (VOT, unified -based *cf*. Table 2 2) and separate models (separate -based *cf*. Table 2 1) [12]. We keep collision data as we find the model to struggle with estimating those. A future direction could be to increase the amount of collision data to deal with this issue.

B. Training Loss

To train VOT and the unified ConvNet-approach (*cf*. Table 2 2), we use the $L2$-norm between the ground truth VO parameters $\boldsymbol{\xi}, \beta$, and their estimated counterparts $\widehat{\boldsymbol{\xi}}, \widehat{\beta}$:

$$\mathcal{L}_{norm} = \|\boldsymbol{\xi} - \widehat{\boldsymbol{\xi}}\|_2^2 + \|\beta - \widehat{\beta}\|_2^2 \tag{1}$$

We further add the geometric (rotation rot and translation $trans$) invariance losses proposed by Zhao *et al.* [12]

$$\begin{aligned} \mathcal{L}_{inv}^{rot} &= \|\widehat{\beta}_{C_t \to C_{t+1}} + \widehat{\beta}_{C_{t+1} \to C_t}\|_2^2 \\ \mathcal{L}_{inv}^{trans} &= \|\widehat{\boldsymbol{\xi}}_{C_t \to C_{t+1}} + \widehat{R} * \widehat{\boldsymbol{\xi}}_{C_{t+1} \to C_t}\|_2^2 \end{aligned} \tag{2}$$

with subscript $C_t \to C_{t+1}$ denoting the estimated parameters from transforming the agent's coordinate system from C_t to C_{t+1} and $C_{t+1} \to C_t$ vice versa.

The final loss can then be written as

$$\mathcal{L} = \mathcal{L}_{norm} + \lambda_1 \mathcal{L}_{inv}^{trans} + \lambda_2 \mathcal{L}_{inv}^{rot} \tag{3}$$

with hyperparameters λ_1, λ_2 to balance the different components. We found $\lambda_1 = 1., \lambda_2 = 1.$ to work well.

*Work done on exchange at EPFL

C. Baselines

We use [12] as our baseline and adapt the publicly available implementation [11]. To match model capacity as suggested by [6], we replace the ResNet-18 backbones with ResNet-50. Note that we explicitly decide against comparing to [6] as their work aims to scale up the dataset size significantly which is the exact opposite of our goal.

We train separate (one model for all actions) and unified (one model for each action) models. However, we find that using the proposed geometric invariance losses for training the unified model yields improved results. Training the separate models closely follows [12] in terms of loss selection and training procedure, *i.e.*, we train the `fwd` model with a batch size of 256 for 150 epochs using $150\,\mathrm{k}$ samples, and the `left` and `right` models with a batch size of 256 for 75 epochs separately until jointly fine-tuning them for 75 additional epochs with a batch size of 224 using $100\,\mathrm{k}$ samples. We use a dropout of 0.2, warm-up training with 10 steps included in the epochs, and do not fine-tune the navigation policy. Table A.1 describes the dataset statistics.

Because the authors published their code and pre-trained weights [11] and we closely follow their setup, we can compare how their approach compares to ours. We, therefore, also use the same navigation policy for all evaluations.

We, further experimented with using ResNet-50 weights pre-trained on ImageNet [4] classification as initialization for the baseline. Pre-training shows to be beneficial for initializing separate models as it allows them to learn faster from a smaller amount of data. However, initializing a unified model with those actually hurts performance.

Finally, we compare to Point-to-point Iterative Closest Point (ICP) [2] implemented in *Open3D* [13]. The method matches two point clouds at timestep t and $t+1$ and estimates the corresponding transformation $\widehat{H}$. We initialize $\widehat{H}$ with the mean transformation of each action and convert the `Depth` maps to point clouds. While the method is reliable for `fwd`, turning `left`,`right` ($\pm 30°$) causes too much deviation in consecutive point clouds, leading to drift and poor

Method	Observations	Samples	$S\uparrow$	SPL↑	SSPL↑	$d_g\downarrow$
oracle	`GPS+Compass`	250 k	97	74	73	29
VOT-B (MultiMAE)	`RGB`	250 k	59	45	66	66
VOT-B (MultiMAE)	`Depth`	250 k	93	71	72	38
VOT-B (MultiMAE)	`RGB-D`	250 k	88	67	71	42
ResNet-50, unified [12]	`RGB`	250 k	45	34	63	95
ResNet-50, unified [12]	`Depth`	250 k	59	45	65	81
ResNet-50, unified [12]	`RGB-D`	250 k	64	48	65	85
ResNet-50, separate [12]	`RGB-D`	250 k	22	13	31	305
ResNet-50, unified [12], pre-train	`RGB`	250 k	33	25	59	136
ResNet-50, unified [12], pre-train	`Depth`	250 k	54	41	64	93
ResNet-50, unified [12], pre-train	`RGB-D`	250 k	53	40	64	95
ResNet-50, separate [12], pre-train	`RGB-D`	250 k	47	36	62	103
DeepVO [9]†	`RGB-D`	1000 k	50	39	65	83
ResNet-18, separate [12]†	`RGB-D`	1000 k	81	62	70	51
ResNet-18, unified [12] †	`RGB-D`	1000 k	72	53	65	83
ICP	`Depth`	–	2.2	1.4	18.5	419.6

Table C.1. Comparison of baselines (w/ and w/o supervised *pre-train*ing on ImageNet [4])and VOT (*cf*. Table 2 13). Success S, SPL, SSPL, and d_g reported as e^{-2}.
† results from [12]

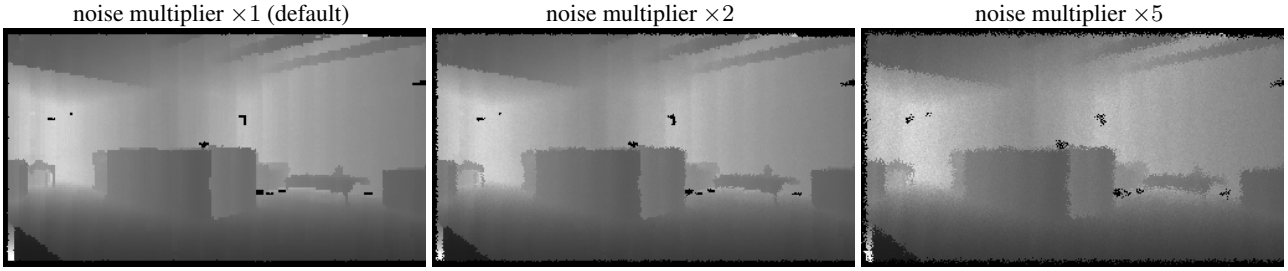


Figure C.1. Qualitative result for increasing the noise multiplier of the R*edwood Noise Model* [3, 7].

Method	Obs.	$S\uparrow$	SPL↑	SSPL↑	$d_g\downarrow$
VOT-B *noise* ×1	`Depth`	92.3	70.8	71.8	42.5
VOT-B *noise* ×2	`Depth`	92.4	70.5	71.5	37.4
VOT-B *noise* ×5	`Depth`	86.3	65.1	68.1	51.5

Table D.1. Sensitivity of pre-trained VOT-B with $[ACT]$ to `Depth` observations with *noise levels* ×1 *(default),* ×2, ×5. Metrics reported as e^{-2}.

performance.

D. Extended Ablation Study

D.1. Sensitivity To Noisy Depth

As shown in Tab. C.1, the Visual Odometry Transformer (VOT) relies on the more valuable information of `Depth` due to its more valuable information in contrast to `RGB` in the Visual Odometry (VO) task. While real-world sensors do not provide perfect `Depth` observations, we evaluate the VOT's sensitivity to different levels of noise in Tab. D.1. The PointNav(-v2) Habitat Challenge 2021 already uses the realistic R*edwood Noise Model* [3, 7]. However, we show VOT's robustness to noisy `Depth` even if the model's noise multiplier is increased from ×1 (default) to ×2 and ×5 (*cf*. Fig. C.1).

D.2. Changing Sensor Suites During Test-time

We showcase our model's ability to deal with changing sensor suites by uniformly dropping modalities at test-time with probability p in Tab. D.2, effectively interpolating between an available and missing sensor, *i.e.*, modality. VOT-B trained to be modality-invariant (*w/ inv.*) does not suffer

Method	Drop	p	$S\uparrow$	SPL↑	SSPL↑	$d_g\downarrow$
VOT-B *w/ & w/o inv.*	RGB	0.00	92.6 / 88.2	70.6 / 67.9	71.3 / 71.3	40.7 / 42.1
VOT-B *w/ & w/o inv.*	RGB	0.25	92.2 / 84.9	70.4 / 65.2	71.8 / 70.8	40.7 / 44.8
VOT-B *w/ & w/o inv.*	RGB	0.50	92.0 / 83.8	70.4 / 65.0	71.8 / 71.4	37.7 / 47.6
VOT-B *w/ & w/o inv.*	RGB	0.75	92.3 / 82.0	70.5 / 63.4	71.5 / 70.9	43.8 / 50.4
VOT-B *w/ & w/o inv.*	RGB	1.00	91.0 / 75.9	69.4 / 58.5	71.2 / 69.9	37.0 / 59.5
VOT-B *w/ & w/o inv.*	Depth	0.00	92.6 / 88.2	70.67 / 67.9	71.3 / 71.3	40.7 / 42.1
VOT-B *w/ & w/o inv.*	Depth	0.25	81.1 / 63.3	61.9 / 48.8	69.4 / 67.8	54.9 / 69.3
VOT-B *w/ & w/o inv.*	Depth	0.50	75.1 / 47.2	57.8 / 36.5	69.2 / 65.0	58.0 / 98.5
VOT-B *w/ & w/o inv.*	Depth	0.75	66.8 / 34.1	51.2 / 25.6	68.1 / 60.5	61.4 / 122.6
VOT-B *w/ & w/o inv.*	Depth	1.00	60.9 / 26.1	47.2 / 20.0	67.7 / 58.7	72.1 / 148.1

Table D.2. Evaluation of changing sensors at test-time. Models *cf*. Tab. 2 - 13 & 14. Metrics reported as e^{-2}.

Resolution	$S\uparrow$	SPL↑	SSPL↑	$d_g\downarrow$
160×80	88.2	67.9	71.3	42.1
224×112	75.0	57.1	67.6	62.0

Table D.3. Results for training VOT (*cf*. Table 2, 16) on different input observation resolutions (width × height). A lower resolution (160×80) trains faster while performing better than the resolution the MultiMAE was pre-trained on (224×112). Losses $\mathcal{L}$, Success S, SPL, SSPL, and d_g reported as e^{-2}.

Method	Drop	$S\uparrow$	SPL↑	SSPL↑	$d_g\downarrow$
Batch drop					
VOT w/ *inv.*	–	92.6	70.67	71.3	40.7
VOT w/ *inv.*	RGB	91.0	69.4	71.2	37.0
VOT w/ *inv.*	Depth	60.9	47.2	67.7	72.1
Sample drop					
VOT w/ *inv.*	–	72.1	54.0	66.4	62.1
VOT w/ *inv.*	RGB	80.7	61.6	68.8	54.0
VOT w/ *inv.*	Depth	61.3	47.2	67.4	71.3

Table D.4. Results for dropping modalities on a batch vs. sample level during training. Losses $\mathcal{L}$, Success S, SPL, SSPL, and d_g reported as e^{-2}.

from catastrophic failure in contrast to training without it.

D.3. Input Resolution

We note that the MultiMAE is trained on an input resolution of 224×224 per modality, *i.e.*, 196 tokens with a patch size of 16×16. Since the Transformer's computation scales quadratically with the input sequence length [8], a higher resolution, therefore, means higher computational expenses. With multiple modality inputs at the same time, this becomes especially problematic. Bachmann *et al*. [1] remedy this issue by encoding only a subset of all input tokens, similar to MAE [5].

This might be a promising direction for reducing the computational requirements for training VOT but there is no clear strategy yet on which tokens to select. When actions are discrete, one could first train a VOT and use the attention maps to identify important regions. This bias could then be injected back into training another VOT but now with masking out, *i.e.*, removing tokens from the input that were not attended to by the first model. While this approach would clearly not be generally applicable, we decided to reduce the computation by reducing the resolution of the input observations. As our experiments show, the model is indeed able to adapt to the change in resolution by fine-tuning on the downstream VO task.

We evaluate the quantitative difference of input resolutions by comparing our proposed observation resolution (width × height) of 80×160 (stacked: 160×160) to 112×224 (stacked 224×224), matching the training resolution of MultiMAE. We train the latter with a batch size of 48 on 2 NVIDIA V100-SXM4-40GB GPUs due to the larger number of input tokens. Both models were trained for 50 epochs. Table D.3 shows that using the original, *i.e.*, higher resolution turns out to be harmful to the VO task.

D.4. Sample- vs. Batch-wise Invariance Training

We investigate how dropping modalities on a batch (*i.e.* all samples in the batch drop the same modality) vs. sample (*i.e.* multi-modal masking) level affects performance. Intuitively, sample-based dropping should lead to more stable training as the model gets updated on all modalities and does not drift toward overfitting a single one which would be the case on a batch level. We compare a modality-invariant VOT (*w/ inv.*) (*cf*. Table 2 17) to the sample-based invariance training in Table D.4 and find that sample-based dropping performs worse than the batch-wise approach.

E. Additional Visualizations

Figure E.1: **Displacement and rotation error** of VOT (*cf*. Table 2 17). We aggregate estimation errors for every action over 150 trajectories and plot them against the magnitude of the ground truth displacement $\boldsymbol{\xi}$ and rotation β.

Figure E.2 **Navigation paths** show the "imaginary" path and the VO estimate of navigation agent using VOT (*cf*. Table 2 16,17).

Figure E.3 **Attention maps** of an estimate w.r.t. different action embeddings. We find that the action type, *i.e*., turning or moving, has a high impact on the resulting attention maps. In contrast, the turning direction has a low impact on the attended regions. We hypothesize that in most cases the displacement is large enough such that it can be inferred by the model. However, knowing that the agent rotates helps to deal with ambiguous cases, *e.g*., a noisy `fwd` action colliding with a wall might be difficult to distinguish from a noisy `left` turning less than 30° [12].

Figure E.3 **Overlayed attention maps** show that VOT ignores artifacts and focuses on distinct visual features in the observations.

References

[1] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *European Conference on Computer Vision*, 2022. 3

[2] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, 1992. 1

[3] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Conference on Computer Vision and Pattern Recognition*, 2015. 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, 2009. 1, 2

[5] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Computer Vision and Pattern Recognition*, 2022. 3

[6] Ruslan Partsey, Erik Wijmans, Naoki Yokoyama, Oles Dobosevych, Dhruv Batra, and Oleksandr Maksymets. Is mapping necessary for realistic pointgoal navigation? In *Conference on Computer Vision and Pattern Recognition*, 2022. 1

[7] Facebook AI Research. Redwood Depth Noise Model [link], 2021. 2

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Advances in Neural Information Processing Systems*, 2017. 3

[9] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *IEEE International Conference on Robotics and Automation*, 2017. 2

[10] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Conference on Computer Vision and Pattern Recognition*, 2018. 6

[11] Xiaoming Zhao. PointNav-VO. `https://github.com/Xiaoming-Zhao/PointNav-VO`, 2021. 1

[12] Xiaoming Zhao, Harsh Agrawal, Dhruv Batra, and Alexander G Schwing. The surprising effectiveness of visual odometry techniques for embodied pointgoal navigation. In *International Conference on Computer Vision*, 2021. 1, 2, 4

[13] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 1

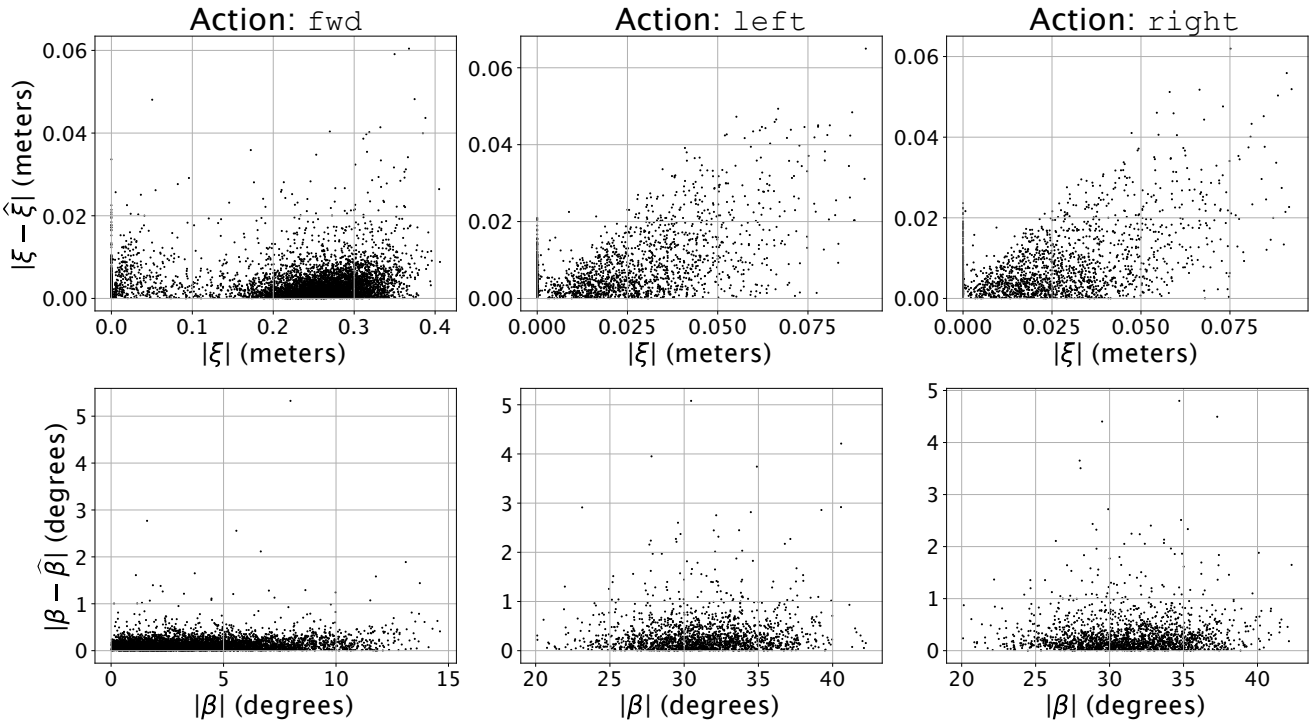
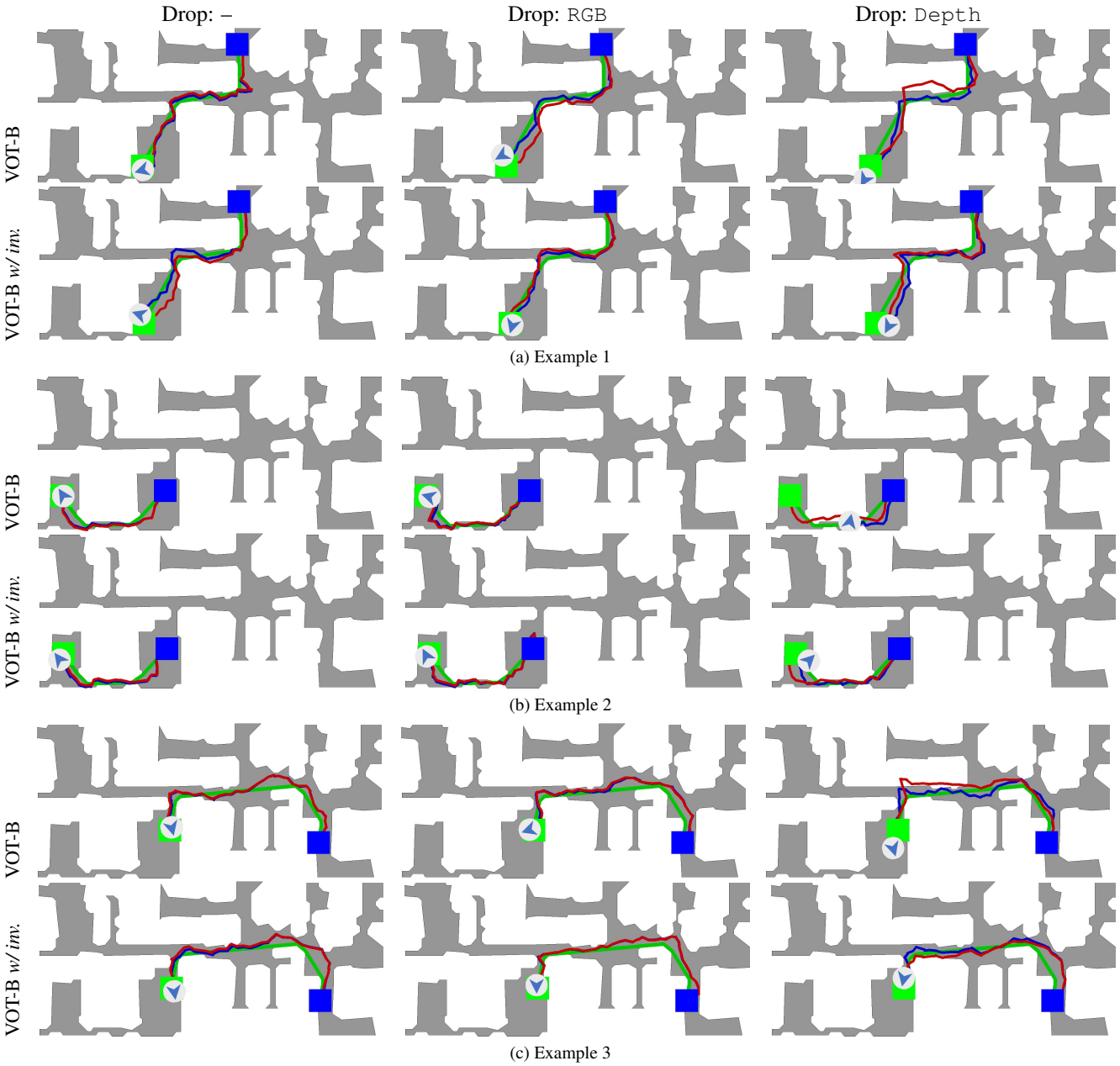


Figure E.1. Displacement and rotation error analysis of VOT (*cf*. Table 2 17).

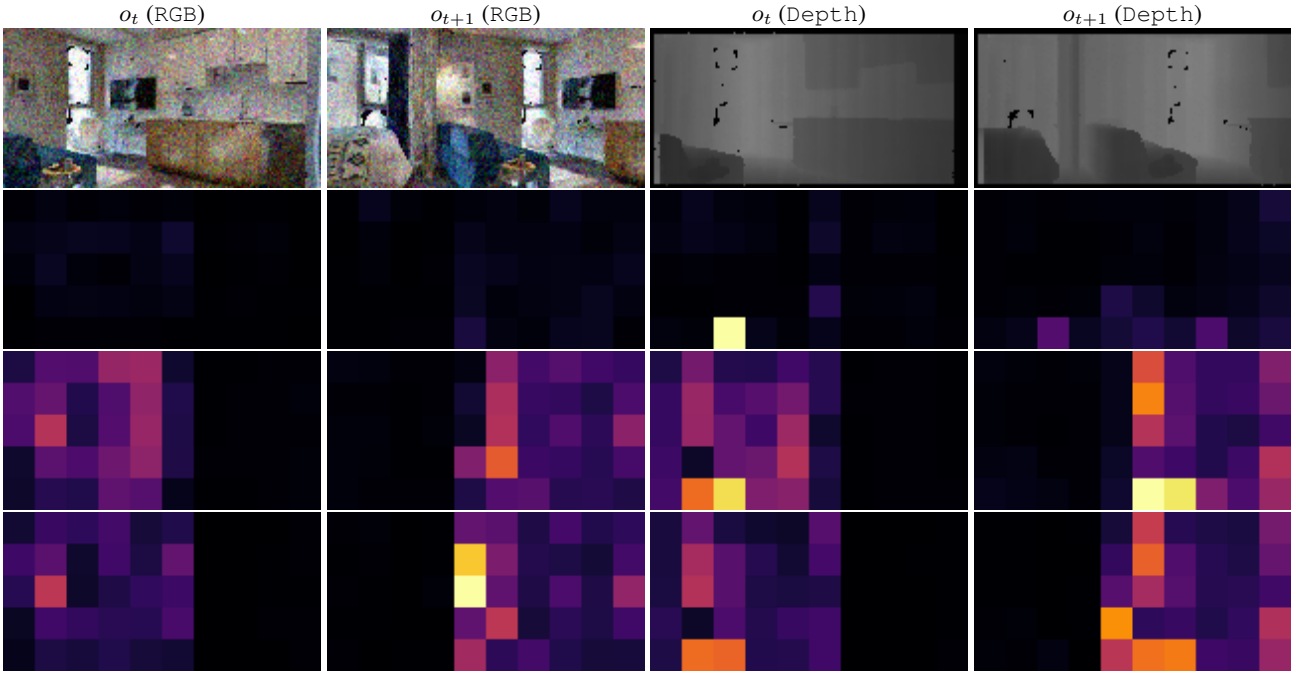


(a) Example 1

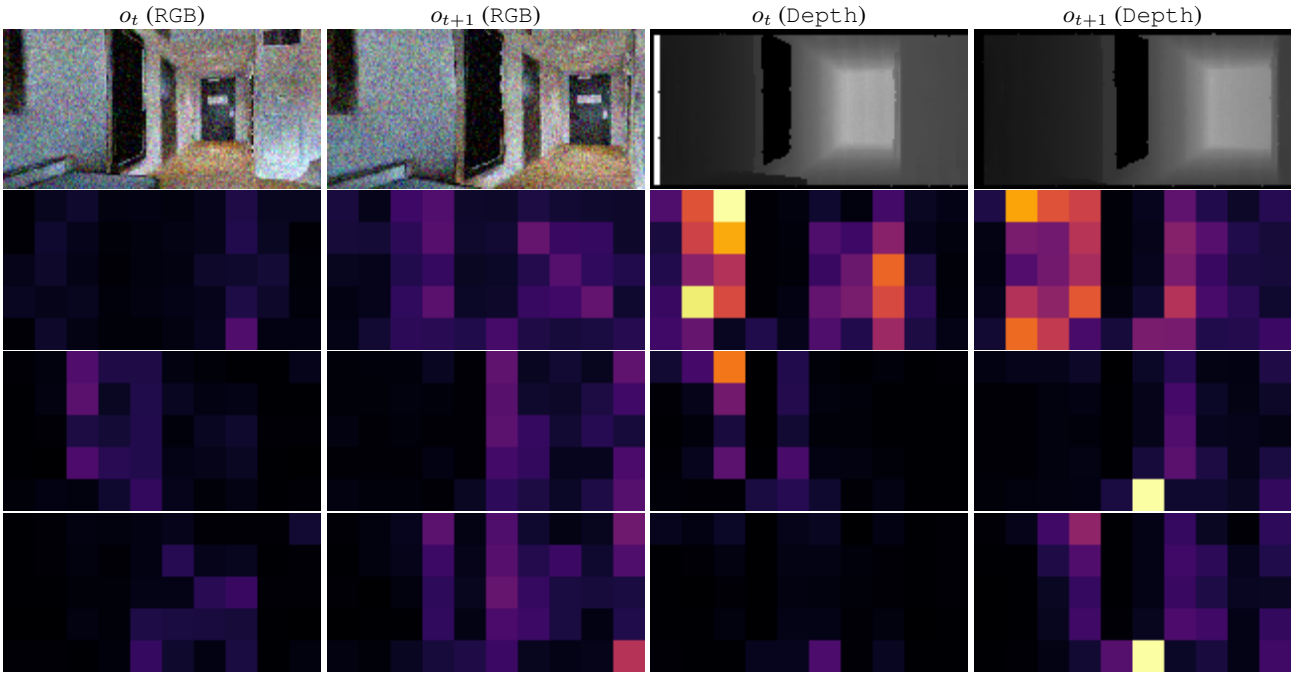
(b) Example 2

(c) Example 3

Figure E.2. Top-down map of the agent navigating the *Cantwell* scene [10] from start (■) to goal (■). The plot shows the shortest path (▬), the path taken by the agent (▬), and the "imaginary" path the agent took, *i.e.*, its VO estimate (▬). We evaluate the model without RGB or Depth (*Drop*) to determine performance when modalities are missing.

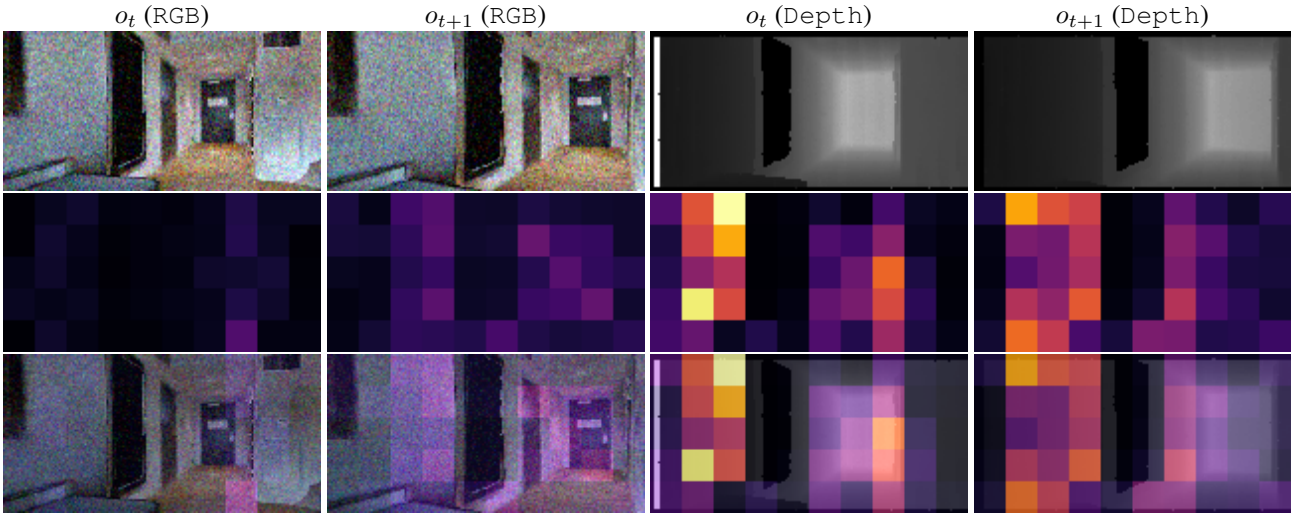


(a) Ground truth action: `left`

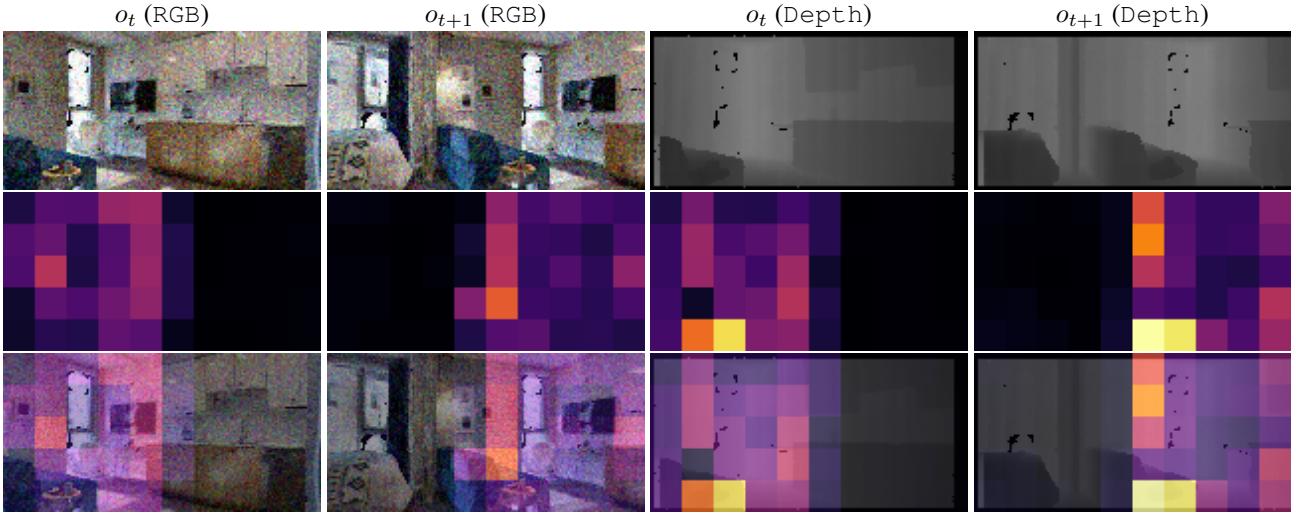


(b) Ground truth action: `fwd`

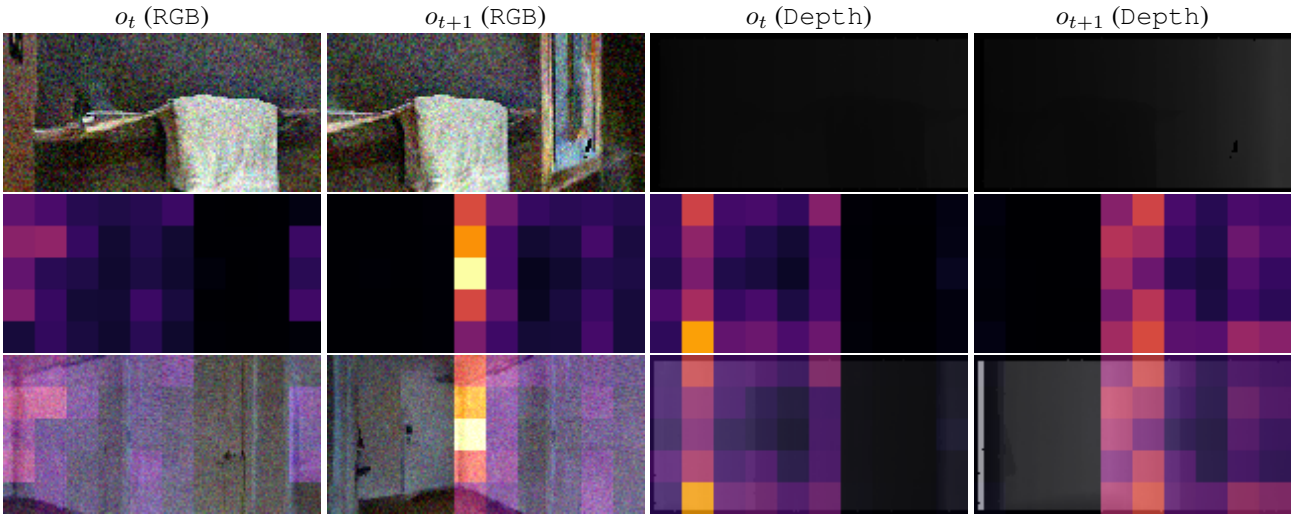
Figure E.3. Attention maps of the last attention layer of VOT (*cf*. Table 2 13). Brighter color indicates higher (■) and darker color lower (■) weighting of the image patch. Impact of different $[ACT]$ tokens on the attention. From top to bottom: observation, $[ACT]$ for `fwd`, `left`, `right`.



(a) Action: `fwd`



(b) Ground truth action: `left`



(c) Ground truth action: `right`

Figure E.4. Attention maps of the last MHA-layer of VOT-B. From top to bottom: observation, attention map, both overlayed.