Modality-invariant Visual Odometry for Embodied Vision
Supplementary Material

Marius Memmel\textsuperscript{1*}, Roman Bachmann\textsuperscript{2}, Amir Zamir\textsuperscript{2}
\textsuperscript{1}University of Washington \hspace{0.8cm} \textsuperscript{2}Swiss Federal Institute of Technology (EPFL)

https://vo-transformer.github.io

A. Dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>fwd</th>
<th>left</th>
<th>right</th>
<th>Collisions</th>
<th>Rollouts</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>training (VOT)</td>
<td>143697</td>
<td>53140</td>
<td>53163</td>
<td>28255</td>
<td>4891</td>
<td>250000</td>
</tr>
<tr>
<td>training [12]</td>
<td>150000</td>
<td>49870</td>
<td>50130</td>
<td>27908</td>
<td>9782</td>
<td>250000</td>
</tr>
<tr>
<td>validation (VOT)</td>
<td>13990</td>
<td>5542</td>
<td>5468</td>
<td>2809</td>
<td>599</td>
<td>25000</td>
</tr>
<tr>
<td>validation [12]</td>
<td>15000</td>
<td>4999</td>
<td>5001</td>
<td>2770</td>
<td>1184</td>
<td>25000</td>
</tr>
</tbody>
</table>

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 $L^2$-norm between the ground truth VO parameters $\xi$, $\beta$, and their estimated counterparts $\hat{\xi}$, $\hat{\beta}$:

$$\mathcal{L}_{\text{norm}} = \| \xi - \hat{\xi} \|^2_2 + \| \beta - \hat{\beta} \|^2_2$$

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

$$\mathcal{L}_{\text{rot}} = \| \hat{\beta}_{C_t \rightarrow C_{t+1}} + \hat{\beta}_{C_{t+1} \rightarrow C_t} \|^2_2$$

$$\mathcal{L}_{\text{trans}} = \| \hat{\xi}_{C_t \rightarrow C_{t+1}} + \hat{R} \hat{\xi}_{C_{t+1} \rightarrow C_t} \|^2_2$$

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

The final loss can then be written as

$$\mathcal{L} = \mathcal{L}_{\text{norm}} + \lambda_1 \mathcal{L}_{\text{trans}} + \lambda_2 \mathcal{L}_{\text{rot}}$$

with hyperparameters $\lambda_1, \lambda_2$ 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 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 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 $\bar{H}$. We initialize $\bar{H}$ 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 ($\pm 30^\circ$) causes too much deviation in consecutive point clouds, leading to drift and poor
Table C.1. Comparison of baselines (w/ and w/o supervised pre-training on ImageNet [4]) and VOT (cf. Table 2.13). Success $S$, SPL, SSPL, and $d_g$ reported as $e^{-2}$.

† results from [12]

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

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

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

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 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 Redwood Noise Model [3, 7]. However, we show VOT’s robustness to noisy Depth even if the model’s noise multiplier is increased from $\times 1$ (default) to $\times 2$ and $\times 5$ (cf. Fig. C.1).

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 Redwood Noise Model [3, 7]. However, we show VOT’s robustness to noisy Depth even if the model’s noise multiplier is increased from $\times 1$ (default) to $\times 2$ and $\times 5$ (cf. Fig. C.1).
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 $224 \times 224$ per modality, i.e., 196 tokens with a patch size of $16 \times 16$. Since the Transformer’s 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 $\times$ height) of $80 \times 160$ (stacked: $160 \times 160$) to $112 \times 224$ (stacked $224 \times 224$), 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 $\xi$ and rotation $\beta$.

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 $\text{fwd}$ action colliding with a wall might be difficult to distinguish from a noisy $\text{left}$ turning less than $30^\circ$ [12].

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

References


[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.


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

(a) Ground truth action: left

(b) Ground truth action: fwd
Figure E.4. Attention maps of the last MHA-layer of VOT-B. From top to bottom: observation, attention map, both overlayed.