Modality-invariant Visual Odometry for Embodied Vision

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

Abstract

Effectively localizing an agent in a realistic, noisy setting is crucial for many embodied vision tasks. Visual Odometry (VO) is a practical substitute for unreliable GPS and compass sensors, especially in indoor environments. While SLAM-based methods show a solid performance without large data requirements, they are less flexible and robust w.r.t. to noise and changes in the sensor suite compared to learning-based approaches. Recent deep VO models, however, limit themselves to a fixed set of input modalities, e.g.*, RGB and depth, while training on millions of samples. When sensors fail, sensor suites change, or modalities are intentionally looped out due to available resources,* e.g.*, power consumption, the models fail catastrophically. Furthermore, training these models from scratch is even more expensive without simulator access or suitable existing models that can be fine-tuned. While such scenarios get mostly ignored in simulation, they commonly hinder a model's reusability in real-world applications. We propose a Transformer-based modality-invariant VO approach that can deal with diverse or changing sensor suites of navigation agents. Our model outperforms previous methods while training on only a fraction of the data. We hope this method opens the door to a broader range of real-world applications that can benefit from flexible and learned VO models.*

1. Introduction

Artificial intelligence has found its way into many commercial products that provide helpful digital services. To increase its impact beyond the digital world, personal robotics and embodied AI aims to put intelligent programs into bodies that can move in the real world or interact with it [15]. One of the most fundamental skills embodied agents must learn is to effectively traverse the environment around them, allowing them to move past stationary manipulation tasks and provide services in multiple locations instead [40]. The ability of an agent to locate itself in an environment is vital to navigating it successfully [12, 64]. A common setup is to equip an agent with an `RGB-D` (`RGB` and `Depth`) camera and a `GPS+Compass` sensor and teach it to navigate to goals in unseen environments [2]. With extended data access through simulators [28, 39, 40, 47, 57], photo-realistic scans of 3D environments [7, 28, 46, 56, 58], and large-scale parallel training, recent approaches reach almost perfect navigation results in indoor environments [55]. However, these agents fail catastrophically in more realistic settings with noisy, partially unavailable, or failing `RGB-D` sensor readings, noisy actuation, or no access to `GPS+Compass` [6, 64].

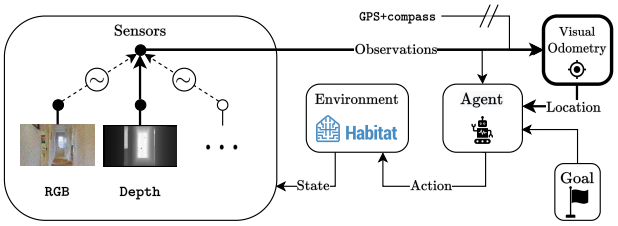


Figure 1. An agent is tasked to navigate to a goal location using `RGB-D` sensors. Because `GPS+Compass` are not available, the location is inferred from visual observations only. Nevertheless, sensors can malfunction, or availability can change during test-time (indicated by $\sim$), resulting in catastrophic failure of the localization. We train our model to react to such scenarios by randomly dropping input modalities. Furthermore, our method can be extended to learn from multiple arbitrary input modalities, *e.g.*, surface normals, point clouds, or internal measurements.

Visual Odometry (VO) is one way to close this performance gap and localize the agent from only `RGB-D` observations [2], and deploying such a model has been shown to be especially beneficial when observations are noisy [12, 64]. However, those methods are not robust to any sensory changes at the test-time, such as a sensor failing, underperforming, or being intentionally looped out. In practical applications [43], low-cost hardware can also experience serious bandwidth limitations, causing `RGB` (3 channels) and `Depth` (1 channel) to be transferred at different rates. Furthermore, mobile edge devices must balance battery usage by switching between passive (*e.g.*, `RGB`) and active (*e.g.*, `LIDAR`) sensors depending on the specific episode. Attempting to solve this asymmetry by keeping

*Work done on exchange at EPFL

separate models in memory, relying on active sensors, or using only the highest rate modality is simply infeasible for high-speed and real-world systems. Finally, a changing sensor suite represents an extreme case of sensor failure where access to a modality is lost during test-time. These points demonstrate the usefulness of a certain level of modality invariance in a VO framework. Those scenarios decrease the robustness of SLAM-based approaches [32] and limit the transferability of models trained on `RGB-D` to systems with only a subset or different sensors.

We introduce *"optional" modalities* as an umbrella term to describe settings where input modalities may be of limited availability at test-time. Figure 1 visualizes a typical indoor navigation pipeline, but introduces uncertainty about modality availability (*i.e*. at test-time, only a subset of all modalities might be available). While previous approaches completely neglect such scenarios, we argue that explicitly accounting for "optional" modalities already *during training* of VO models allows for better reusability on platforms with different sensor suites and trading-off costly or unreliable sensors during test-time. Recent methods [12, 64] use Convolution Neural Network (ConvNet) architectures that assume a constant channel size of the input, which makes it hard to deal with multiple "optional" modalities. In contrast, Transformers [51] are much more amenable to variable-sized inputs, facilitating the training of models that can optionally accept one or multiple modalities [4].

Transformers are known to require large amounts of data for training from scratch. Our model's data requirements are significantly reduced by incorporating various biases: We utilize multi-modal pre-training [4, 17, 30], which not only provides better initializations but also improves performance when only a subset of modalities are accessible during test-time [4]. Additionally, we propose a token-based action prior. The action taken by the agent has shown to be beneficial for learning VO [35, 64] and primes the model towards the task-relevant image regions.

We introduce the Visual Odometry Transformer (VOT), a novel modality-agnostic framework for VO based on the Transformer architecture. Multi-modal pre-training and an action prior drastically reduce the data required to train the architecture. Furthermore, we propose explicit modality-invariance training. By dropping modalities during training, a single VOT matches the performance of separate uni-modal approaches. This allows for traversing different sensors during test-time and maintaining performance in the absence of some training modalities.

We evaluate our method on point-goal navigation in the *Habitat Challenge 2021* [1] and show that VOT outperforms previous methods [35] with training on only 5% of the data. Beyond this simple demonstration, we stress that our framework is modality-agnostic and not limited to `RGB-D` input or discrete action spaces and can be adapted to various modalities, *e.g*., point clouds, surface normals, gyroscopes, accelerators, compass, etc. To the best of our knowledge, VOT is the first widely applicable modality-invariant Transformer-based VO approach and opens up exciting new applications of deep VO in both simulated and real-world applications. We make our code available at github.com/memmelma/VO-Transformer.

2. Related Work

SLAM- vs Learning-based Navigation: Simultaneous Localization and Mapping (SLAM) approaches decompose the navigation task into the components of mapping, localization, planning, and control [49]. These methods rely on explicit visual feature extraction and, therefore, fail in realistic settings with noisy observations [64], while learning-based methods are more robust to noise, ambiguous observations, and limited sensor suites [27, 32]. However, learning-based methods require an order of magnitude more data, *e.g*., available through simulation [40]. To deal with the large data requirements, SLAM- and learning-based methods can be combined [5, 8, 9, 11, 48, 61, 63].

Visual Odometry for Realistic Indoor Navigation: While most VO methods estimate an agent's pose change from more than two frames [52, 53] or optical flow [66], subsequent frames in indoor environments share almost no overlap and contain many occlusions due to the large displacement caused by the discrete action space [64]. Datta *et al*. [12] propose to estimate the pose change from consecutive frames via a ConvNet architecture and decouple learning the VO from the task-specific navigation policy to allow for retraining modules when dynamics change or the actuation experiences noise. Zhao *et al*. [64] improve the model's robustness to observation and actuation noise through geometric invariance losses [54], separate models for moving and turning, pre-process observations, and introduce dropout [44]. Finally, Partsey *et al*. [35] explore the need for explicit map building in autonomous indoor navigation. They apply train- and test-time augmentations and concatenate an action embedding similar to Zhao *et al*. [64] to the extracted visual features. A trend is to exploit simulators to gather large datasets (1M [64], 5M [35]). While this is a reasonable progression, it is infeasible to re-train the VO model whenever dynamics or sensor configurations change.

Multi-modal Representation Learning: The availability of multi-modal or pseudo-labeled [4] data [13, 16, 34, 38, 59, 65], *e.g*., depth, video, and audio, makes it possible to learn feature-rich representations over multiple modalities. Together with Transformer's [51] ability to process a token sequence of arbitrary length, this leads to general-purpose architectures that can handle various modalities [23] like video, images, and audio [30] or single-view 3D geometry [17]. In particular, Multi-modal Multi-task Masked Autoencoder (MultiMAE) [4] is a multi-modal pre-training

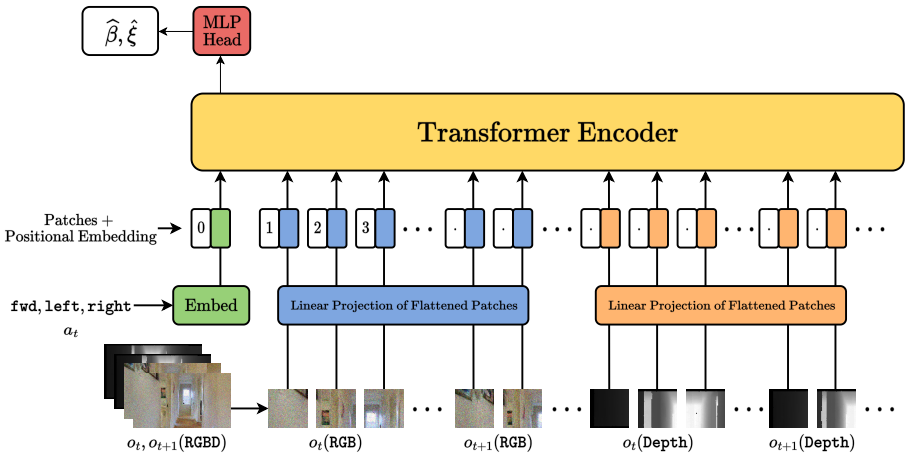


Figure 2. The Visual Odometry Transformer architecture for RGB-D input. Image patches are turned into tokens through modality-specific linear projections before a fixed positional embedding is added to them. We pass an action token that embeds the action taken by the agent. An MLP-head then estimates the VO parameters $\widehat{\beta}, \widehat{\boldsymbol{\xi}}$ from the output token corresponding to the input action token. By randomly dropping either RGB or Depth during training, the Transformer backbone becomes modality-agnostic, allowing it to deal with a subset of these input modalities during test-time without losing performance. When more modalities are available during training, other modality-specific linear projections can be added to process the additional information.

strategy that performs masked autoencoding [19] with RGB, Depth, and Semantic Segmenation (SemSeg). We show that fine-tuning a pre-trained MultiMAE model can significantly increase VO performance using only 5% of the training data amount of previous methods [35].

3. Proposed Method

3.1. Preliminaries

In the realistic PointGoal Navigation task [2], an agent spawns at a random position in an unseen environment and is given a random goal location $\boldsymbol{g}_t$ relative to its starting position. At each time step t of an episode, the agent perceives its environment through observations $\boldsymbol{o}_t$ and executes an action a_t from a set of discrete actions (move fwd $0.25m$, turn left and right by 30°). The stop action indicates the agent's confidence in having reached the goal. Because the relative goal position $\boldsymbol{g}_t$ is defined at the beginning of each episode, it has to be updated throughout the episode as the actions change the agent's position and orientation. Following [12,64], we update $\boldsymbol{g}_t$ through an estimate of the agent's coordinate transformation. With access to GPS+Compass, computing this transformation is trivial. However, since those sensors are unavailable, we estimate the transformation from the agent's subsequent observations $\boldsymbol{o}_t, \boldsymbol{o}_{t+1}$ and update the estimated relative goal position $\widehat{\boldsymbol{g}}_t$. When taking an action a_t, the agent's coordinate system C_t transforms into C_{t+1}. Because the agent can only navigate planarly in the indoor scene, we discard the 3rd dimension for simplicity. We define the estimated transformation as $\widehat{\boldsymbol{H}} \in SE(2)$, with $SE(2)$ being the group of rigid transformations in a 2D plane and parameterize it by the estimated rotation angle $\widehat{\beta} \in \mathbb{R}$ and estimated translation vector $\widehat{\boldsymbol{\xi}} \in \mathbb{R}^2$:

$$\widehat{\boldsymbol{H}} = \begin{bmatrix} \widehat{R} & \widehat{\boldsymbol{\xi}} \\ 0 & 1 \end{bmatrix}, \quad \widehat{R} = \begin{bmatrix} \cos(\widehat{\beta}) & -\sin(\widehat{\beta}) \\ \sin(\widehat{\beta}) & \cos(\widehat{\beta}) \end{bmatrix} \in SO(2). \quad (1)$$

We then learn a VO model f_ϕ with parameters ϕ predicting $\widehat{\beta}, \widehat{\boldsymbol{\xi}}$ from observations $\boldsymbol{o}_t, \boldsymbol{o}_{t+1}$: $\widehat{\beta}, \widehat{\boldsymbol{\xi}} = f_\phi(\boldsymbol{o}_t, \boldsymbol{o}_{t+1})$ Finally, we transform $\widehat{\boldsymbol{g}}_t$ in coordinate system C_t to the new agent coordinate system C_{t+1} by $\widehat{\boldsymbol{g}}_{t+1} = \widehat{\boldsymbol{H}} \cdot \widehat{\boldsymbol{g}}_t$.

3.2. Visual Odometry Transformer

Model Architecture: When facing "optional" modalities, it is not yet clear how systems should react. Options range from constructing an alternative input, *e.g.*, noise [29], to falling back on a model trained without the missing modalities, to training the network with placeholder inputs [31]. Besides these, recent approaches depend on a fixed set of modalities during train- and test-time due to their ConvNet-based backbone. Transformer-based architectures can process a variable number of input tokens and can be explicitly trained to accept fewer modalities during test-time while observing multiple modalities throughout training [4,51]. Furthermore, the Transformer's global receptive field could be beneficial for VO, which often gets solved with correspondence or feature matching techniques [41]. We, therefore, propose the Visual Odometry Transformer (VOT), a multimodal Transformer-based architecture for VO.

Visual Odometry Estimation: To estimate the VO parameters, we pass the encoded Action Token ($[ACT]$) token to a prediction head. We use a two-layer Multi-layer Perceptron (MLP) with learnable parameters ψ composed into $\boldsymbol{W_0} \in \mathbb{R}^{d \times d_h}, \boldsymbol{b}_0 \in \mathbb{R}^{d_h}$, and $\boldsymbol{W_1} \in \mathbb{R}^{d_h \times 3}, \boldsymbol{b}_1 \in \mathbb{R}^3$

with token dimensions $d = 768$, and hidden dimensions $d_h = d/2$. A Gaussian Error Linear Unit (GELU) [21] acts as the non-linearity between the two layers. The VO model can then be defined as a function $f_{\phi,\psi}(\boldsymbol{o}_t, \boldsymbol{o}_{t+1}, a_t)$ taking as input the action a_t and the observations $\boldsymbol{o}_t, \boldsymbol{o}_{t+1}$ corresponding to either `RGB`, `Depth`, or `RGB-D` and predicting the VO parameters $\widehat{\beta}, \widehat{\boldsymbol{\xi}}$. Simplifying the backbone as $b_\phi(\boldsymbol{o}_t, \boldsymbol{o}_{t+1}, a_t)$ that returns extracted visual features $\boldsymbol{v}_{t\to t+1} \in \mathbb{R}^{1\times d}$, and governed by parameters ϕ, the resulting model is:

$$\begin{aligned} b_\phi(\boldsymbol{o}_t, \boldsymbol{o}_{t+1}, a_t) &= \boldsymbol{v}_{t\to t+1} \\ \mathrm{MLP}_\psi(\boldsymbol{v}) &= \mathrm{GELU}(\boldsymbol{v}\boldsymbol{W_0} + \boldsymbol{b}_0)\boldsymbol{W_1} + \boldsymbol{b}_1 \\ f_{\phi,\psi}(\boldsymbol{o}_t, \boldsymbol{o}_{t+1}, a_t) &= \mathrm{MLP}_\psi(b_\phi(\boldsymbol{o}_t, \boldsymbol{o}_{t+1}, a_t)) = \widehat{\beta}, \widehat{\boldsymbol{\xi}} \end{aligned} \tag{2}$$

Action Prior: The action a_t taken by the agent to get from $\boldsymbol{o}_t$ to $\boldsymbol{o}_{t+1}$ is a powerful prior on the VO parameters. To provide this information to the model, we embed the action using an embedding layer [36]. This layer acts as a learnable lookup for each action, mapping it to a fixed-size embedding. With the embedding size equal to the token dimensions, we can create an $[ACT]$ and pass the information directly to the model (*cf*. Figure 2). In contrast to [35, 64], we pass the token directly to the encoder instead of concatenating it to the extracted features. This practice conditions the visual feature extraction on the action and helps ignore irrelevant parts of the image. Note that this approach is not limited to discrete actions but tokens could represent continuous sensor readings like accelerometers, gyroscopes, and compasses, allowing for flexible deployment, *e.g*., in smartphones or autonomous vehicles [43].

Explicit Modality-invariance Training: Explicitly training the model to be invariant to its input modalities is one way of dealing with missing sensory information during test-time. To enforce this property, we drop modalities during training to simulate missing modalities during test-time. Furthermore, this procedure can improve training on less informative modalities by bootstrapping model performance with more informative ones. For example, `RGB` is more prone to overfitting than `Depth` because the model can latch onto spurious image statistics, *e.g*. textures. Training on `RGB`-only would likely cause the model to latch onto those and converge to local minima, not generalizing well to unseen scenes. By increasing the amount of `Depth` observations seen during training, the model learns to relate both modalities, acting as regularization. We model this notion as a multinomial distribution over modality combinations (here: `RGB`, `Depth`, `RGB-D`) with equal probability. For each batch, we draw a sample from the distribution to determine on which combination to train.

4. Experimental Evaluation

4.1. Setup

Simulation: We use the AI Habitat (Habitat) simulator for data collection and model evaluation, following the Habitat PointNav Challenge 2020 [1] specifications. The guidelines define an action space of `fwd` (move forward $0.25m$), `left` (turn left by 30°), `right` (turn right by 30°), and `stop` (indicate the agent reached its goal), and include a sensor suite of `RGB-D` camera, and `GPS+Compass` (not used in the realistic PointGoal navigation task). The `RGB` observations get returned into a $[0, 255]$ range while the `Depth` map is scaled to $[0, 10]$. Both sensors are subject to noise, *i.e*., noisy actuations [33] and observations [10]. Furthermore, collision dynamics prevent *sliding*, a behavior that allows the agent to slide along walls on collision. Cosmetic changes bring the simulation closer to the LoCoBot [18], a low-cost robotic platform with an agent radius of $0.18m$ and height of $0.88m$. An optical sensor resolution of 341×192 (width $\times$ height) emulates an Azure Kinect camera. An episode is successful if the agent calls `stop` in a radius two times its own, *i.e*., $0.36m$, around the point goal and does so in $T = 500$ total number of time steps. By specification, the 3D scenes loaded into Habitat are from the Gibson [57] dataset, more precisely Gibson-4+ [40], a subset of 72 scenes with the highest quality. The validation set contains 14 scenes, which are not part of the training set.

Dataset: For training VOT, we collect a training- and a validation dataset. Each set consists of samples containing the ground truth translation $\boldsymbol{\xi}$ and rotation parameters β retrieved from a perfect `GPS+Compass` sensor, observations $\boldsymbol{o}_t, \boldsymbol{o}_{t+1}$, and taken action a_t. We keep samples where the agent collides with its environment as the transformations strongly differ from standard behavior [64]. The collection procedure follows Zhao *et al*. [64] and is performed as: 1) initialize the Habitat simulator and load a scene from the dataset, 2) place the agent at a random location within the environment with a random orientation, 3) sample a navigable PointGoal the agent should navigate to, 4) compute the shortest path and let the agent follow it, and 5) randomly sample data points along the trajectory. We collect $250\,\mathrm{k}$ observation-transformation pairs from the training and $25\,\mathrm{k}$ from the validation scenes of Gibson-4+, which is significantly less than comparable methods ($1\,\mathrm{M}$ [64], $5\,\mathrm{M}$ [35]). Furthermore, we apply data augmentation during training to the `left` and `right` actions by horizontally flipping the observations and computing the inverse transformation.

Loss Function: Our loss function is the $L2$-norm between the ground truth VO parameters and their estimated counterparts. We further add the geometric invariance losses $\mathcal{L}_{inv}$ proposed by Zhao *et al*. [64] and use the Adam [26] optimizer ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e^{-8}$) to minimize the resulting loss function $\mathcal{L} = \|\boldsymbol{\xi} - \widehat{\boldsymbol{\xi}}\|_2^2 + \|\beta - \widehat{\beta}\|_2^2 + \mathcal{L}_{inv}$.

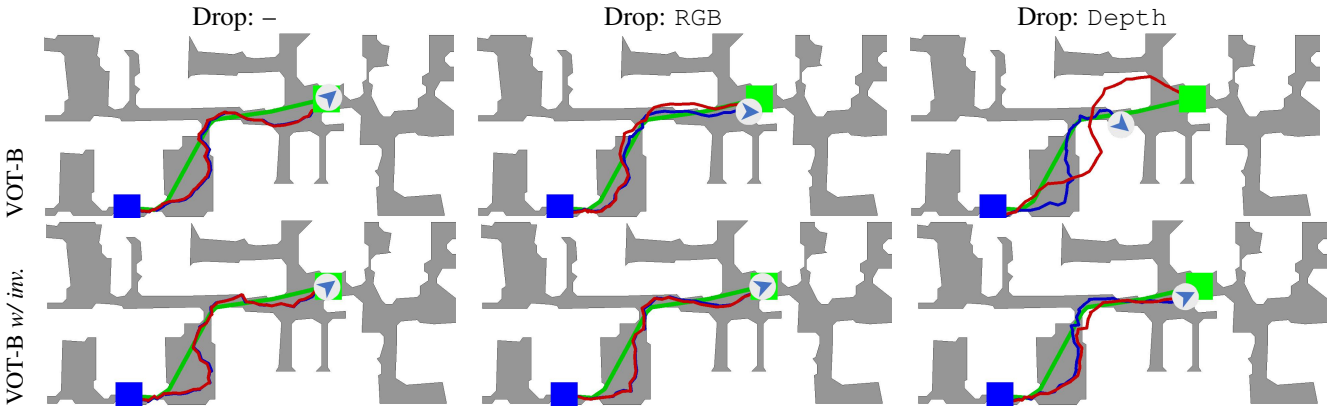


Figure 3. Top-down map of the agent navigating the *Cantwell* scene [58] from start (■) to goal (■). The plot shows the shortest path (■), the path taken by the agent (■), and the "imaginary" path the agent took, *i.e.*, its VO estimate (■). We evaluate the model without `RGB` or `Depth` (*Drop*) to determine performance when modalities are missing. As expected, the VOT relies heavily on both modalities, causing the estimation to drift when either `RGB` or `Depth` is unavailable. The localization error accumulates over the course of the trajectory and causes the true and imaginary path to diverge, resulting in failure to complete the episodes. Training a VOT to be modality-invariant (VOT *w/ inv.*) removes those reliances and leads to success even when modalities are missing.

Method	Drop	$S\uparrow$	SPL↑	SSPL↑	$d_g\downarrow$
VOT `RGB`	–	59.3	45.4	66.7	66.2
VOT `Depth`	–	93.3	71.7	72.0	38.0
[12]	–	64.5	48.9	65.4	85.3
VOT	–	88.2	67.9	71.3	42.1
VOT w/ *inv.*	–	**92.6**	**70.6**	**71.3**	**40.7**
[12]	`RGB`	0.0	0.0	5.4	398.7
VOT	`RGB`	75.9	58.5	69.9	59.5
VOT w/ *inv.*	`RGB`	**91.0**	**69.4**	**71.2**	**37.0**
[12]	`Depth`	0.0	0.0	5.4	398.7
VOT	`Depth`	26.1	20.0	58.7	148.1
VOT w/ *inv.*	`Depth`	**60.9**	**47.2**	**67.7**	**72.1**

Table 1. Results for dropping modalities during test-time. Training a VOT to be modality-invariant (*w/ inv.*) leads to no performance drop in comparison to a VOT trained on a single modality (VOT `RGB`, VOT `Depth`). This shows that a single VOT *w/ inv.* can replace multiple modality-dependent counterparts. Previous approaches [12,35,64] become inapplicable, converging to a `Blind` behavior. Metrics reported as e^{-2}. **Bold** indicates best results.

Pre-training: Pre-training is a well-known practice to deal with the large data requirements of Vision Transformers (ViTs) [14, 60], especially in a VO setting where data is scarce [14,25,45]. We use the pre-trained MultiMAE (`RGB` + `Depth` + `SemSeg`) made publicly available by Bachmann *et al*. [3]. Since `SemSeg` is unavailable in our setting, we discard the corresponding projection layers.

Training Details: We follow prior work [12, 35, 64] and train our navigation policy and VO model separately before jointly evaluating them on the validation set. In contrast to [12, 64], we do not fine-tune the navigation policy on the trained VO models as it has shown minimal navigation performance gains in [64] and was abandoned in [35].

We train all models, including baselines, for 100 epochs with 10 warm-up epochs that increase the learning rate linearly from 0.0 to $2e^{-4}$, and evaluate the checkpoints with the lowest validation error. We further find gradient norm clipping [62] (max gradient norm of 1.0) to stabilize the training of VOT but to hurt the performance of the ConvNet baselines. The training was done with a batch size of 128 on an NVIDIA V100-SXM4-40GB GPU with automatic mixed-precision enabled in PyTorch [36] to reduce memory footprint and speed up training. Our backbone is a ViT-B [14] with a patch size of 16×16 and 12 encoder blocks with 12 Multi-head Attention (MHA) heads each, and token dimensions 768. To encode the input into tokens, we use a 2D sine-cosine positional embedding and separate linear projection layers for each modality. Note that if additional modalities are available, our model can be extended by adding additional linear input projections or fine-tuning existing ones [4]. Finally, we pass all available tokens to the model and resize each observation to $160 \times 80 \times c$ (width $\times$ height $\times$ channels c) and concatenate modalities along their height to $160 \times 160 \times c$ to reduce computation. We keep a running mean and variance to normalize `RGB` and `Depth` to zero mean and unit variance.

Evaluation Metrics: Anderson *et al*. [2] propose the Success weighted by (normalized inverse) Path Length (SPL) to evaluate agents in a PointGoal or ObjectGoal navigation setting. A crucial component of this metric is the success of an episode (success $S = 1$, failure $S = 0$). With l the shortest path distance from the starting position and p the length of the path taken by the agent, the SPL over N episodes is defined as $\text{SPL} = \frac{1}{N}\sum_{i=0}^{N-1} S^{(i)} \frac{l^{(i)}}{\max(p^{(i)}, l^{(i)})}$.

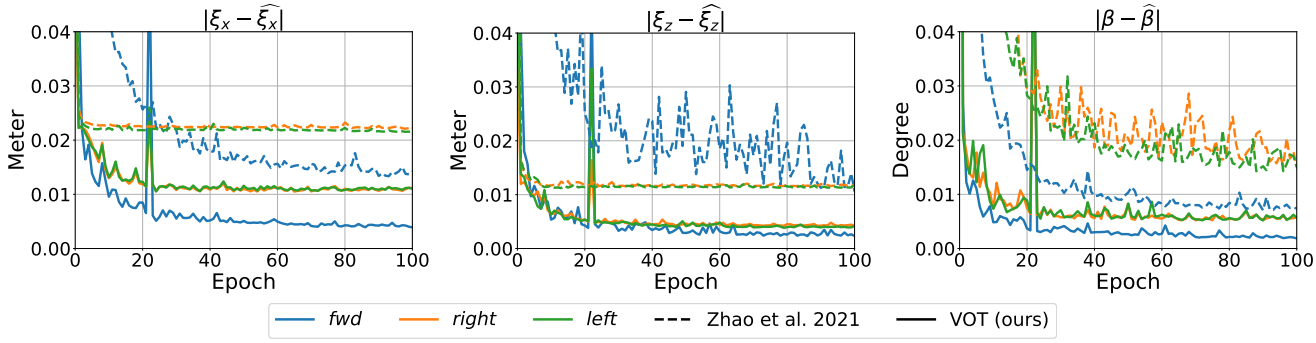


Figure 4. Absolute difference between ground truth translations $\boldsymbol{\xi_x}, \boldsymbol{\xi_z}$ and rotation angle $\boldsymbol{\beta}$ to their estimated counterparts $\widehat{\cdot}$. We compare *Zhao et al.* [64] (Table 2, 2) to the VOT (Table 2, 13). Our model estimates `fwd` translation along the z-axis (*middle*), `left`, `right` along z-, x-axis (*left, middle*), and the turning angle β (*right*) more accurately than the baseline. We successfully capture the displacements caused by the noisy actuation with an average error (over both axis x, z) of 0.25 cm (`fwd`), 0.7 cm (`right`), and 0.65 cm (`left`).

While SPL depends on the success of an episode, [12] propose the Soft Success Path Length (SSPL) that provides a more holistic view of the agent's navigation performance. The authors replace the binary success S of an episode with a soft value consisting of the ratio between the (geodesic) distances to target upon start d_{init} and termination of an episode d_g. The resulting metric is then SSPL $= \frac{1}{N}\sum_{i=0}^{N-1}\left(1 - d_g^{(i)}/d_{init}^{(i)}\right)\frac{l^{(i)}}{\max\left(p^{(i)}, l^{(i)}\right)}$ The closer the agent gets to the goal, the higher the SSPL, even if the episode is unsuccessful. This softening allows distinguishing agents that fail to complete a single or multiple episodes but move significantly close to the goal from ones that move away from it. Without access to `GPS+Compass`, SSPL becomes significantly more important as an agent might call `stop` prematurely due to inaccurate localization. We report the SPL, SSPL, success S, and (geodesic) distance to goal on termination d_g on the validation scenes of Gibson-4+ with decimals truncated.

Navigation Policy: Similar to prior work [12, 35, 64], we replace the `GPS+Compass` with our VO model to estimate the relative goal position, which serves as the input to a pre-trained navigation policy. We use the same pre-trained policy as Zhao *et al.* [64] for our experiments, which was trained using a goal position updated by ground truth localization. The policy architecture consists of a Long Short-Term Memory (LSTM) [22] with two recurrent layers that process 1) a 512-dimensional encoding of the agent's observations $\boldsymbol{o_t}$ (here: `Depth`), 2) a 32-dimensional embedding of the previous action, and 3) a 32-dimensional embedding of the updated relative goal position. The observation encoding gets obtained by passing the observations $\boldsymbol{o}_t$ through a ResNet-18 [20] backbone, flattening the resulting feature map to dimensionality 2052, and projecting it to dimensionality 512 with a fully-connected layer. Finally, the output of the LSTM is fed through another fully-connected layer to produce a distribution over the action space and a value function estimate. The policy was trained using DDPO [55], a distributed version of Proximal Policy Optimization (PPO) [42].

4.2. Dealing With Optional Modalities

We evaluate the models' robustness to missing modalities by randomly dropping access to one of the training modalities. This setup probes VOT for dependencies on the input modalities, which directly influence the downstream performance under limited access. In case of sensor malfunctioning, *e.g.*, only a single modality might be available, a ConvNet's failure is predetermined as it requires a fixed-size input. If not given, the system converges to a `Blind` behavior, exemplified in Table 1. Limiting access to modalities reveals VOT's dependency on `Depth`. Dropping `RGB` barely decreases performance, while dropping `Depth` causes the localization to fail more drastically. Comparing the true agent localization and its "imaginary", *i.e.*, VO estimate, it becomes clear why. Figure 3 shows how the errors accumulate, causing the true location to drift away from the estimate. While the effect is less drastic when dropping `RGB`, the agent still fails to reach the goal.

Training VOT with the proposed invariance training (*w/ inv.*), *i.e.*, sampling `RGB` for 20%, `Depth` for 30%, and `RGB-D` for 50% of the training batches, eliminates this shortcoming. Removing `RGB` now only decreases the success rate by 1.6%, while removing `Depth` also leads to a stronger performance. This observation suggests that `RGB` is less informative for the VO task than `Depth`. Especially when navigating narrow passages, `RGB` might consist of uniform observations, *e.g.*, textureless surfaces like walls, making it hard to infer the displacement, unlike `Depth` which would still provide sufficient geometric information (*cf*. Figure 3). However, this information asymmetry only leads to a decline in the metrics that are sensitive to subtle inconsistencies in the localization, *i.e.*, S, and SPL. Inspecting the SSPL, the drop of -3.5 is less drastic. Explicit modality-invariance training keeps VOT-B (`RGB-D`)

	Method	Observations	Pre-train	$[ACT]$	$S\uparrow$	SPL↑	SSPL↑	$d_g\downarrow$	$\mathcal{L}_{train}\downarrow$	$\mathcal{L}_{val}\downarrow$
1	[64] (separate)	RGB-D			22.4	13.8	31.5	305.3	0.125	0.186
2	[64] (unified)	RGB-D		✔	64.5	48.9	65.4	85.3	0.264	0.420
3	Blind	–			0.0	0.0	5.4	398.7	48.770	47.258
4	VOT-B	RGB			27.1	21.2	57.7	177.0	0.735	1.075
5	VOT-B	Depth			43.2	32.0	59.3	122.5	**0.441**	**0.644**
6	VOT-B	RGB-D			**47.3**	**36.3**	**61.2**	**119.7**	1.256	1.698
7	Blind	–		✔	13.3	10.0	46.3	251.8	1.637	1.641
8	VOT-B	RGB		✔	42.0	32.3	62.7	107.0	0.043	0.571
9	VOT-B	Depth		✔	**76.1**	**58.8**	**69.2**	**60.7**	**0.017**	**0.113**
10	VOT-B	RGB-D		✔	72.1	55.6	68.5	64.4	0.019	0.129
11	VOT-B	RGB	✔		54.5	41.3	65.2	69.9	0.056	0.347
12	VOT-B	Depth	✔		83.2	63.4	69.1	**49.9**	0.079	0.205
13	VOT-B	RGB-D	✔		**85.7**	**65.7**	**69.7**	56.1	**0.021**	**0.060**
14	VOT-B	RGB	✔	✔	59.3	45.4	66.7	66.2	0.003	0.280
15	VOT-B	Depth	✔	✔	**93.3**	**71.7**	**72.0**	**38.0**	**0.004**	**0.044**
16	VOT-B	RGB-D	✔	✔	88.2	67.9	71.3	42.1	0.004	0.051
17	VOT-B *w/ inv.*	RGB-D	✔	✔	**92.6**	**70.6**	**71.3**	**40.7**	**0.008**	**0.094**
	oracle	GPS+Compass	–	–	97.8	74.8	73.1	29.9	–	–

Table 2. Ablation study of architecture design and input modalities. We further investigate pre-training with MultiMAE [4] in models 11-14. Losses $\mathcal{L}$, Success S, SPL, SSPL, and d_g reported as e^{-2}. **Bold** indicates best results.

Rank	Participant team	S	SPL	SSPL	d_g
1	**MultiModalVO (VOT)** (ours)	93	74	77	21
2	VO for Realistic PointGoal [35]	94	74	76	21
3	inspir.ai robotics	91	70	71	70
4	VO2021 [64]	78	59	69	53
5	Differentiable SLAM-net [24]	65	47	60	174

Table 3. **Habitat Challenge 2021.** Results for the Point Nav Test-Standard Phase (test-std split) retrieved on 05-Nov-2022.

from exploiting this asymmetry and matches the performance of VOT-B (RGB) when Depth is dropped during test-time Tab. 1.

4.3. Quantitative Results

We compare our approach to Zhao *et al*. [64] in terms of downstream navigation performance, *i.e*., the VO model as GPS+Compass replacement for a learned navigation agent. We use the same publicly available navigation policy for both approaches and the published VO models of the baseline [64]. Using only 25% of the training data, VOT improves performance by $S+12.3$, SPL+9.7, SSPL+2.0 (*cf*. Table 2 15) and $S+7.2$, SPL+5.7, SSPL+1.3 (*cf*. Table 2 16). When training the baseline on our smaller data set (*cf*. Table 2 2, unified, ResNet-50), this improvement increases to $S+29.8$, SPL+22.8, SSPL+6.6 (*cf*. Table 2 15) and $S+23.7$, SPL+19.0, SSPL+5.9 (*cf*. Table 2 16).

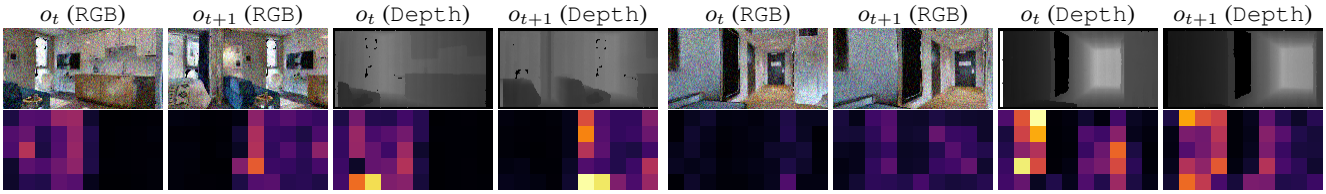
To capture the raw VO performance detached from the indoor navigation task, we inspect the absolute prediction error in Figure 4. We differentiate between translation $\boldsymbol{\xi}$ in x- and y- direction (ξ_x, ξ_y), and taken action. VOT is accurate up to 0.36 cm (fwd), 1.04 cm (right), 1.05 cm (left) in x- direction and 0.20 cm (fwd), 0.41 cm (right), 0.38 cm (left) in z-direction. Note how the baseline struggles to capture ξ_z, corresponding to the forward-moving direction z when taking the fwd action.

Given the results in Table 2, we advise using VOT trained on Depth-only when access is assumed, as the difference to using GPS+Compass is a mere $S-4.5$, SPL−3.1, SSPL−1.1. When "optional" modalities are needed, *e.g*., they are expected to change during test-time, invariance training should be used. Trained on RGB-D, this setup also reaches GPS+Compass like performance with differences of only $S-5.2$, SPL−4.2, SSPL−1.8.

4.4. Ablation Study

We identify the impact of different input modalities and model design choices in our ablation study (*cf*. Table 2). Without observations, the Blind VO model cannot update the goal position. This means the agent can only act without goal-related feedback, resulting in a 0% success rate.

Extending the model with our proposed $[ACT]$ token allows it to surpass the Blind performance. Able to up-



(a) Ground truth action: `left` (b) Ground truth action: `fwd`

Figure 5. 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. The VOT learns to focus on regions present in both time steps $t, t+1$, *i.e.*, outer image regions for turning `left`, and center regions for moving `fwd`. Artifacts of the Gibson dataset get ignored (*cf*. Figure 5b).

date the relative goal position, the agent reaches an SSPL of 46.3, but due to the actuation noise, it calls `stop` correctly only 13.3% of the time. Access to `RGB` or `Depth` allows the VO model to adjust to those unpredictable displacements. While the `RGB` and `Depth` observations correlate with the $[ACT]$ token, they also contain information about the noisy actuation. Vice versa, $[ACT]$ disambiguates corner cases where the visual observations do not provide explicit information about the underlying action. For instance, a `fwd` action colliding with a wall might be hard to distinguish from a noisy `left` turning less than 30° [64].

Our results show that MultiMAE pre-training provides useful multi-modal features for VO that fine-tuned outperform the ConvNet baselines. In addition, these features are complementary to the $[ACT]$ prior, together achieving state-of-the-art results. We conclude that the $[ACT]$ prior biases the model towards the mean of the corresponding transformation, while the pre-training supports the learning of the additive actuation noise.

Training separate models for each modality reveals that `Depth` is a more informative modality than `RGB` for VO. We assume this to be a direct result of its geometric properties, *i.e.*, the 3D structure of the scene. We find that training VOT on noisy `RGB` even hurts the localization. The model overfits the visual appearance of the scenes and is unable to generalize to unseen ones. In turn, `Depth` does not suffer from this issue as it only contains geometric information.

4.5. Action-conditioned Feature Extraction

We show what image regions the model attends to by visualizing the attention maps of the last MHA-layer (*cf*. Table 2 16) corresponding to the $[ACT]$ token in Figure 5. To reduce the dimensionality of the visualization, we fuse the heads' weights via the max operator and align the attention maps with the input images. We normalize the maps to show the full range of the color scheme.

We find that passing different actions to VOT primes it to attend to meaningful regions in the image. When passed turning actions `left` or `right`, VOT focuses on regions present at both time steps. This makes intuitive sense, as a turning action of 30° strongly displaces visual features or even pushes them out of the agent's field of view. A similar behavior emerges for a `fwd` action which leads to more attention on the center regions, *e.g.*, the walls and the end of a hallway (*cf*. Figure 5b). These results are particularly interesting as the model has no prior knowledge about the VO task but learns something about its underlying structure.

4.6. Habitat Challenge 2021 PointNav

We compare our approach (*cf*. Table 2 16) to several baselines submitted to the *Habitat Challenge 2021* benchmark in Table 3. Using the same navigation policy as Partsey *et al*. [35], VOT achieves the highest SSPL and on par SPL and d_g training on only 5% of the data. These results clearly show that reusability doesn't come with a price of lower performance and that scaling data requirements doesn't seem to be the answer to solving deep VO.

4.7. Limitations

In our work, we separate the VO model from the navigation policy and only focus on the modality-invariance of the former, neglecting that the navigation policy expects `Depth` as input [12, 35, 64]. Designing policies to be modality-invariant is subject to future research. Assuming an accurate sensor failure detection when dropping modalities, additionally, is an idealized setup. Furthermore, our experiments in the Habitat's simulator limit the available modalities to `RGB-D`. Even though `SemSeg` has shown to be beneficial for some VO applications [37, 50], there is no specific sensor for it. However, `SemSeg` could be estimated from `RGB`. While our experiments focus on discrete actions and `RGB-D`, our architecture could be adapted to continuous actions and other sensor types. However, training might become more difficult due to a lack of pre-trained weights.

5. Conclusions

We present Visual Odometry Transformers for learned Visual Odometry. Through multi-modal pre-training and action-conditioned feature extraction, our method is sample efficient and outperforms current methods trained on an order of magnitude more data. With its modality-agnostic design and modality-invariance training, a single model can deal with different sensor suites during training and can trade-off subsets of those during test-time.

References

[1] Abhishek Kadian*, Joanne Truong*, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Are We Making Real Progress in Simulated Environments? Measuring the Sim2Real Gap in Embodied Visual Navigation. In *arXiv preprint arXiv:1912.06321*, 2019. 2, 4

[2] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. In *arXiv preprint arXiv:1807.06757*, 2018. 1, 3, 5

[3] Roman Bachmann and David Mizrahi. Multimae: Multi-modal multi-task masked autoencoders. https://github.com/EPFL-VILAB/MultiMAE, 2022. 5

[4] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *European Conference on Computer Vision*, 2022. 2, 3, 5, 7

[5] Shehroze Bhatti, Alban Desmaison, Ondrej Miksik, Nantas Nardelli, N Siddharth, and Philip HS Torr. Playing doom with slam-augmented deep reinforcement learning. In *arXiv preprint arXiv:1612.00380*, 2016. 2

[6] Lukas Biewald. Habitat challenge 2020. https://eval.ai/web/challenges/challenge-page/580/leaderboard/1631, 2020. 1

[7] 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 *International Conference on 3D Vision*, 2017. 1

[8] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations*, 2020. 2

[9] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *Conference on Computer Vision and Pattern Recognition*, 2020. 2

[10] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Conference on Computer Vision and Pattern Recognition*, 2015. 4

[11] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. Deepfactors: Real-time probabilistic dense monocular slam. In *IEEE Robotics and Automation Letters*, 2020. 2

[12] Samyak Datta, Oleksandr Maksymets, Judy Hoffman, Stefan Lee, Dhruv Batra, and Devi Parikh. Integrating egocentric localization for more realistic point-goal navigation agents. In *Conference on Robot Learning*, 2021. 1, 2, 3, 5, 6, 8

[13] 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. 2

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. 5

[15] Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks. In *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2022. 1

[16] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *International Conference on Computer Vision*, 2021. 2

[17] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *Computer Vision and Pattern Recognition*, 2022. 2

[18] Abhinav Gupta, Adithyavairavan Murali, Dhiraj Prakashchand Gandhi, and Lerrel Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. In *Advances in Neural Information Processing Systems*, 2018. 4

[19] 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

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016. 6

[21] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). In *arXiv preprint arXiv:1606.08415*, 2016. 4

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. In *Neural Computation*, 1997. 6

[23] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2022. 2

[24] Peter Karkus, Shaojun Cai, and David Hsu. Differentiable slam-net: Learning particle slam for visual navigation. In *Computer Vision and Pattern Recognition*, 2021. 7

[25] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *International Conference on Computer Vision*, 2015. 5

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 4

[27] Noriyuki Kojima and Jia Deng. To learn or not to learn: Analyzing the role of learning for navigation in virtual environments. In *arXiv preprint arXiv:1907.11770*, 2019. 2

[28] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. In *arXiv preprint arXiv:1712.05474*, 2017. 1

[29] Yu-Jhe Li, Jinhyung Park, Matthew O'Toole, and Kris Kitani. Modality-agnostic learning for radar-lidar fusion in vehicle detection. In *Conference on Computer Vision and Pattern Recognition*, 2022. 3

[30] Valerii Likhosherstov, Anurag Arnab, Krzysztof Choromanski, Mario Lucic, Yi Tay, Adrian Weller, and Mostafa Dehghani. Polyvit: Co-training vision transformers on images, videos and audio. In *arXiv preprint arXiv:2111.12993*, 2021. 2

[31] Marius Memmel, Camila Gonzalez, and Anirban Mukhopadhyay. Adversarial Continual Learning for Multi-domain Hippocampal Segmentation. In *Domain Adaptation and Representation Transfer, and Affordable Healthcare and AI for Resource Diverse Global Health*, 2021. 3

[32] Dmytro Mishkin, Alexey Dosovitskiy, and Vladlen Koltun. Benchmarking classic and learned navigation in complex 3d environments. In *arXiv preprint arXiv:1901.10915*, 2019. 2

[33] Adithyavairavan Murali, Tao Chen, Kalyan Vasudev Alwala, Dhiraj Gandhi, Lerrel Pinto, Saurabh Gupta, and Abhinav Gupta. Pyrobot: An open-source robotics framework for research and benchmarking. In *arXiv preprint arXiv:1906.08236*, 2019. 4

[34] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, 2012. 2

[35] 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. 2, 3, 4, 5, 6, 7, 8

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019. 4, 5

[37] Noha Radwan, Abhinav Valada, and Wolfram Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. In *IEEE Robotics and Automation Letters*, 2018. 8

[38] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *International Conference on Computer Vision*, 2021. 2

[39] Manolis Savva, Angel X Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. Minos: Multimodal indoor simulator for navigation in complex environments. In *arXiv preprint arXiv:1712.03931*, 2017. 1

[40] 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 *Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 4

[41] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. In *IEEE Robotics & Automation Magazine*, 2011. 3

[42] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017. 6

[43] Siddhartha S Srinivasa, Patrick Lancaster, Johan Michalove, Matt Schmittle, Colin Summers, Matthew Rockett, Joshua R Smith, Sanjiban Choudhury, Christoforos Mavrogiannis, and Fereshteh Sadeghi. Mushr: A low-cost, open-source robotic racecar for education and research. In *arXiv preprint arXiv:1908.08031*, 2019. 1, 4

[44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. In *The Journal of Machine Learning Research*, 2014. 2

[45] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. In *arXiv preprint arXiv:2106.10270*, 2021. 5

[46] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. In *arXiv preprint arXiv:1906.05797*, 2019. 1

[47] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems*, 2021. 1

[48] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *Advances in Neural Information Processing Systems*, 2021. 2

[49] Sebastian Thrun. Probabilistic robotics. In *Communications of the ACM*, 2002. 2

[50] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. In *IEEE International Conference on Robotics and Automation*, 2018. 8

[51] 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. 2, 3

[52] 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

[53] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. In *The International Journal of Robotics Research*, 2018. 2

[54] Xiangwei Wang, Daniel Maturana, Shichao Yang, Wenshan Wang, Qijun Chen, and Sebastian Scherer. Improving

learning-based ego-motion estimation with homomorphism-based losses and drift correction. In *International Conference on Intelligent Robots and Systems*, 2019. 2

[55] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations*, 2020. 1, 6

[56] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment. In *International Conference on Learning Workshops*, 2018. 1

[57] Fei Xia, William B Shen, Chengshu Li, Priya Kasimbeg, Micael Edmond Tchapmi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese. Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments. In *IEEE Robotics and Automation Letters*, 2020. 1, 4

[58] 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. 1, 5

[59] Amir R. Zamir, Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2

[60] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Computer Vision and Pattern Recognition*, 2022. 5

[61] Huangying Zhan, Chamara Saroj Weerasekera, Jia-Wang Bian, and Ian Reid. Visual odometry revisited: What should be learnt? In *IEEE International Conference on Robotics and Automation*, 2020. 2

[62] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020. 5

[63] Jingwei Zhang, Lei Tai, Ming Liu, Joschka Boedecker, and Wolfram Burgard. Neural slam: Learning to explore with external memory. In *arXiv preprint arXiv:1706.09520*, 2017. 2

[64] 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, 3, 4, 5, 6, 7, 8

[65] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. In *International Journal of Computer Vision*, 2019. 2

[66] Ran Zhu, Mingkun Yang, Wang Liu, Rujun Song, Bo Yan, and Zhuoling Xiao. Deepavo: Efficient pose refining with feature distilling for deep visual odometry. In *Neurocomputing*, 2022. 2