Supplementary Material for MoDA: Map style transfer for self-supervised Domain Adaptation of embodied agents

Eun Sun Lee[1 [0000−0003−1731−5714]], Junho Kim[1 [0000−0002−5947−2147]], SangWon Park[1 [0000−0002−9735−1303]], and Young Min Kim[1 [0000−0002−6735−8539]]

Seoul National University, Seoul, 08826, Republic of Korea
{ eunsunlee, 82magnolia, paulmoguri, youngmin.kim } @snu.ac.kr

In supplementary material, we describe the details of our agent training setup and baseline implementation. Additionally, we explain the experiment setup along with the list of scenes used for each experiment scenario, show the experimental result of an additional baseline and describe the supplementary video.

1 Agent Training Setup

We implement our domain adaptation method, MoDA, on the visual navigation agent from Active Neural SLAM [2]. The agent navigates with a discrete action selected from { move forward by 25cm, turn right by 10 °, turn left by 10 °}. The pretrained agent is trained in a noiseless environment with ground-truth supervision using Adam optimizer [6] with a batch size of 4. We use the learning rate of 1×10^{-4}. The pretrained agent then explores on 10 trajectories for 50 steps each to collect the ground-truth map dataset D_{gt}. When the agent is deployed in the new environment with visual and dynamics corruptions, it similarly collects the noisy map dataset D_{noisy}. Given the two collections, we leverage CycleGAN [11] and train our map style transfer networks with a batch size of 1 and Adam optimizer. The network is quickly trained within 5 epochs using the learning rate of 2×10^{-4}.

During visual domain adaptation, our agent is fine-tuned on 30 unseen trajectories where the agent explores for 100 steps each. We train our agent with visual domain loss $\mathcal{L}_V$ and set the loss weight $\lambda_{st}^{\text{ego}} = 0.8$ and $\lambda_{fc} = 0.2$. For dynamics domain adaptation, our visually adapted agent is further fine-tuned on 50 trajectories exploring for 50 steps. The dynamics domain loss $\mathcal{L}_D$ consists of the loss weight $\lambda_{st}^{\text{global}} = 1.0$ and $\lambda_{tc} = 0.2$. We use Adam optimizer with a batch size of 1 in all our adaptation setups. The learning rate of 1×10^{-6} is used for visual domain adaptation and 1×10^{-4} for dynamics domain adaptation.

2 Baseline Implementation Details

In this section, we explain the implementation details of baselines; "Domain Randomization (**DR**)", "Policy Adaptation during Deployment (**PAD**)" and "Global Map Consistency (**GMC**)".

The Domain Randomization agent is trained on the domain randomly exposed to various visual corruptions and dynamics corruptions. We randomly select a visual variation among speckle noise, light variation, and defocus blur as suggested in [3, 9]. For each noise type, we also variate the parameters to train the agent for wide varieties of visual corruptions. For dynamics corruptions, we implement the actuation noise model suggested in RobustNav [3]. However, the magnitudes of noise parameters are modified from the proposed values based on our agent's action space. The actuation noise is generated from three types of noise models; motion drift and stochastic/constant motion bias. The motion drift is selected from [No drift, Drift right by 10°, Drift left by 10°]. The stochastic motion bias is selected from $\mathcal{N}(\mu_s, \Sigma_s)$. Specifically, μ_s is $\mathbf{0}$ and Σ_s is drawn from $\left(\begin{smallmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 3.3 \end{smallmatrix}\right)$, where the diagonal values correspond to pose (x, y), and orientation ϕ. Additionally, the constant motion bias for forward step and turn actions is randomly chosen from [±0.05m, ±0.1m, ±0.15m] and [±1.7°, ±3.3°, ±5°].

As suggested in [5], we train PAD model using a visual domain adaptation method utilizing an auxiliary self-supervision task. Specifically, the separate rotation prediction task is implemented on our agent. The RGB observation given at each step is randomly rotated by an angle selected from [0°, 90°, 180°, 270°]. The agent then learns to predict the rotation angle during training. Once deployed, the agent only fine-tunes the self-supervision task to adapt to the new visual domain. However, when we investigate the PAD model's adaptation performance only with the suggested visual domain adaptation, we observe no improvement over the pretrained agent's degraded performance. We thus additionally adapt the agent with our proposed dynamics domain adaptation. PAD results in the lowest pose estimation error on the logged trajectories, but it fails to improve the pretrained agent's performance in mapping and final navigation tasks. Nonetheless, the result shows that our proposed dynamics domain adaptation can be easily combined with other visual domain adaptation methods to enhance a visual navigation agent's pose estimation performance.

Lastly, Lee et al. [7] suggests a self-supervised domain adaptation method which encourages global map consistency loss on round trip trajectories to learn dynamics corruptions. We thus let the agent explore for 50 steps, turn around 180°, and execute the previous action sequence in reverse to generate round trip trajectories. The agent is fine-tuned for 160 trajectories, inefficiently learning only during its backward path. The proposed method from [7] adapts the agent in an offline manner and only suggests a dynamics domain adaptation method. As suggested in [7], we use the Adam optimizer with the learning rate 5×10^{-5}. To conclude, we summarize the comparison between our model and the baselines in Table 1.

3 Experiment Setup

Our pretrained agent is trained on the standard train split [8] of Gibson dataset [10]. We use the unseen scenes of Gibson and Matterport3D [1] to evaluate the suggested domain adaptation method. For the generalization experiment setting,

Table 1: We compare our agent adapted with MoDA with the implemented baselines in terms of essential aspects in domain adaptation method

	Visual Domain Adaptation	Dynamics Domain Adaptation	Online/ Offline	Self-Supervision
DR	✓	✓	-	
PAD	✓		Online	✓
GMC		✓	Offline	✓
MoDA	✓	✓	Online	✓

we transfer the pretrained agent to the scenes in adaptation split and evaluate the transferred agent on the scenes in evaluation split. We evaluate on the new unseen scenes to show that our agent is transferred to the unknown corruptions rather than the scenes seen during adaptation. However, in the specialization experiment, the pretrained agent is fine-tuned and evaluated on the same scenes in adaptation split. This setup shows a more practical scenario as the embodied agents may be continuously deployed in the same scenes.

We randomly assign the scenes into either adaptation or evaluation split. The list of scenes for each split is shown in Table 2. The scenes in Gibson are tested for the visual corruptions with speckle noise and low-lighting condition, while the scenes in Matterport3D are used without modification to test the scene-scale change. Note that all scenes additionally contain dynamics corruptions. In all our experiments, we evaluate our agents on 50 trajectories. For mapping and localization, the agent is evaluated on 50 trajectories with 500 step lengths. In exploration and PointNav, the agent executes the given task for 1000 steps on 50 trajectories.

Table 2: List of scenes used for adaptation and evaluation split

Gibson Scenes with Speckle Noise or Low-Lighting Condition and Dynamics Corruptions	
Adaptation	Sisters, Ribera, Denmark, Mosquito, Elmira, Cantwell, Pablo
Evaluation	Eastville, Edgemere, Eudora, Greigsville, Sands, Scioto, Swormville
Matterport3D Scenes with Large Scene Scale and Dynamics Corruptions	
Adaptation	pRbA3pwrgk9, ZMojNkEp431, r1Q1Z4BcV1o, 5LpN3gDmAk7, VVfe2KiqLaN, mJXqzFtmKg4, GdvgFV5R1Z5, 1pXnuDYAj8r, 17DRP5sb8fy, 82sE5b5pLXE, e9zR4mvMWw7, aayBHfsNo7d, VLzqgDo317F, kEZ7cmS4wCh, PuKPg4mmafe, JmbYfDe2QKZ, 1LXtFkjw3qL, HxpKQynjfin, sKLMLpTHeUy, D7G3Y4RVNrH, ULsKaCPVFJR, XcA2TqTSSAj, b8cTxDM8gDG, 5q7pvUzZiYa, dhjEzFoUFzH, YmJkqBEsHnH, rPc6DW4iMge, sT4fr6TAbpF, Uxmj2M2itWa, PX4nDJXEHrG
Evaluation	29hnd4uzFmX, 2n8kARJN3HM, 759xd9YjKW5, 7y3sRwLe3Va, 8WUmhLawc2A, B6ByNegPMKs, D7N2EKCX4Sj, E9uDoFAP3SH, EDJbREhghzL, JF19kD82Mey, JeFG25nYj2p, Pm6F8kyY3z2, S9hNv5qa7GM, SN83YJsR3w2, V2XKFyX4ASd, VFuaQ6m2Qom, Vvot9Ly1tCj, Vzqfbhrp DEA, ac26ZMwG7aT, cV4RVeZvu5T, gTV8FGcVJC9, gZ6f7yhEvPG, i5noydFURQK, jh4fc5c5qoQ, p5wJjkQkbXX, qoiz87JEwZ2, r47D5H71a5s, s8pcmisQ38h, uNb9QFRL6hY, ur6pFq6Qu1A, vyrNrziPKCB

4 Additional Baseline: SECANT

We additionally evaluated a baseline "SECANT [4]" which generalizes the pretrained agent with a representation learning method. Given an expert mapper which is trained in a noiseless scenes, we train a student mapper with strong augmentation methods from [4]. We also evaluate the agent which further adapts SECANT with our dynamics adaptation method, referred as "SECANT+MoDA" in Table 3. The mapping result demonstrates that our adaptation method outperforms both SECANT and SECANT+MoDA, as similarly shown in Fig 1. Furthermore, SECANT+MoDA baseline shows that our dynamics adaptation method can easily provide an integrated solution for various powerful representation learning methods.

		Gibson Speckle Noise			Gibson Low-Lighting			Matterport3D Large Scene Scale		
		(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
Pose	$\theta(^\circ)$	**2.61**	3.74	15.02	2.87	**2.41**	15.78	**2.34**	3.00	14.42
	x, y(m)	0.04	**0.03**	**0.03**	0.05	**0.03**	0.04	0.05	**0.03**	0.04
Map (MSE)	ego	**1.08**	1.11	1.11	**0.89**	0.90	0.90	**1.02**	1.07	1.07
	global	**0.25**	0.27	0.34	**0.25**	0.27	0.34	**0.31**	0.35	0.42
Exploration	area	28.63	**29.50**	27.73	31.56	**32.99**	30.31	**63.68**	61.66	52.16
	ratio	0.82	**0.84**	0.79	0.91	**0.93**	0.87	**0.54**	0.51	0.45
	collision	0.36	**0.32**	0.43	**0.26**	0.30	0.34	**0.28**	0.31	0.44
Point Nav	success	**0.56**	0.40	0.16	**0.56**	**0.56**	0.16	**0.22**	0.18	0.02
	SPL	**0.47**	0.33	0.13	0.45	**0.46**	0.14	**0.18**	0.14	0.02

Table 3: Generalization result of (a)MoDA (b)SECANT+MoDA and (c)SECANT in the new environments with visual variations and dynamics corruptions.

5 Description on the Supplementary Video

We provide a video which shows our agent, trained with MoDA, exploring the three scenarios suggested in experiments from the main paper. For each scenario, we compare the exploration sequence of our agent and the pretrained agent. Both agents begin exploring from the same initial position in the same scene for 500 steps. At each frame, we show a visually corrupted RGB observation seen by the agent at each step. The predicted map is overlaid on the ground-truth map to show its mapping performance. Lastly, the localization performance is shown by the predicted trajectory drawn along with the ground-truth trajectory. The result in the video proves that our agent is successfully transferred to the novel

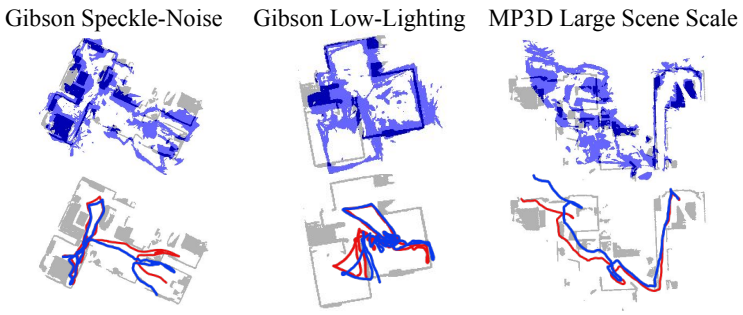

Fig. 1: Mapping (top) and Localization (bottom) result of SECANT+MoDA comparable to Fig 5 of the main paper

environments with both visual and dynamics corruptions, outperforming the pretrained agent deployed without any adaptation.

References

1. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. arXiv preprint arXiv:1709.06158 (2017)
2. Chaplot, D.S., Gandhi, D., Gupta, S., Gupta, A., Salakhutdinov, R.: Learning to explore using active neural slam. arXiv preprint arXiv:2004.05155 (2020)
3. Chattopadhyay, P., Hoffman, J., Mottaghi, R., Kembhavi, A.: Robustnav: Towards benchmarking robustness in embodied navigation. ArXiv **abs/2106.04531** (2021)
4. Fan, L., Wang, G., Huang, D.A., Yu, Z., Fei-Fei, L., Zhu, Y., Anandkumar, A.: Secant: Self-expert cloning for zero-shot generalization of visual policies. arXiv preprint arXiv:2106.09678 (2021)
5. Hansen, N., Jangir, R., Sun, Y., Alenyà, G., Abbeel, P., Efros, A.A., Pinto, L., Wang, X.: Self-supervised policy adaptation during deployment. arXiv preprint arXiv:2007.04309 (2020)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
7. Lee, E.S., Kim, J., Kim, Y.M.: Self-supervised domain adaptation for visual navigation with global map consistency. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1707–1716 (January 2022)
8. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., Batra, D.: Habitat: A Platform for Embodied AI Research. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
9. Truong, J., Chernova, S., Batra, D.: Bi-directional domain adaptation for sim2real transfer of embodied navigation agents. IEEE Robotics and Automation Letters **6**(2), 2634–2641 (2021)
10. Xia, F., Zamir, A.R., He, Z., Sax, A., Malik, J., Savarese, S.: Gibson env: Real-world perception for embodied agents. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9068–9079 (2018)
11. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Computer Vision (ICCV), 2017 IEEE International Conference on (2017)