

SUPPLEMENTARY

Auxiliary Tasks for Efficient Learning of Point-Goal Navigation

Saurabh Satish Desai
desaisau@oregonstate.edu
Oregon State University

Stefan Lee
leestef@oregonstate.edu
Oregon State University

1. Supplement Contents

In this supplementary document, we add the following -

- **Section 2** : Details the quantized depth prediction network architecture from section 3.2 of main paper.
- **Section 3** : Provides SPL and success curves for analyzing performance of agents as an extension to Table 1 of main paper.
- **Section 4** : Provides SPL curves for a comprehensive comparison between DD-PPO agents and our task-equipped agents.
- **Section 5** : Provides the SPL curves for comparison with existing auxiliary techniques in addition to main results presented in Table 3 of main paper.
- **Section 6** : Provides table showing how far are the predictions of position encoders from the actual position.
- **Section 7** : Discusses how well do the task-equipped agents perform on auxiliary objectives.

2. Quantized Depth Prediction

The original depth map produced by Habitat simulator has dimensions 256×256 with depth being a floating point value in range 0 and 1 i.e $\in (0, 1)^{256 \times 256}$. In order to obtain central field of view of agents, we center-crop this map to 128×128 . Earlier, we had experimented with full sized depth map and did not find any serious drop in agent performance on account of cropping. This makes sense intuitively as predicting the depth of floor and ceiling might not very useful from navigation perspective. As we only care about the general structure of scene geometry, we avoid pixel-wise predictions. Rather, we aggregate non-overlapping, neighbouring patches of pixels to obtain a depth target map. These aggregated values are further quantized into

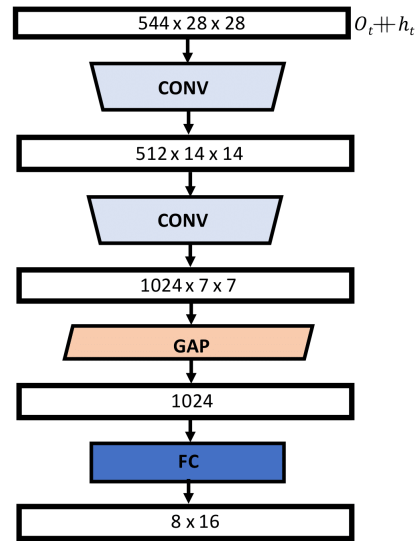


Figure 1. A schematic of auxiliary depth prediction network. It consumes tiled block $(O_t + h_t)$ to predict distribution over pixel classes.

8 bins - $\{0, 0.05, 0.175, 0.3, 0.425, 0.55, 0.675, 0.8, 1\}$ where 0 - 0.05 is class 0, 0.05-0.175 is class 1 and so on. The smaller depths are emphasized to better differentiate between close-by and nearly-colliding objects. We carry out an average pooling with kernel size 32 of the cropped map followed by quantization to produce the output target map $D_t \in [0, \dots, 7]^{4 \times 4}$.

The auxiliary network shown in Fig. 1 consumes the tiled convolutional block $O_t + h_t$. Here CONV block represents series of operations - {Conv 2d 3×3 , GroupNorm, ReLU} in order. We have used GroupNorm consistently with group size of 32 following [4] in order to exploit the correlation among layer channels. GAP is global average pooling layer. The layer FC is a fully connected layer which maps vector of size 1024 to that of size 128. This is reshaped to 8×16

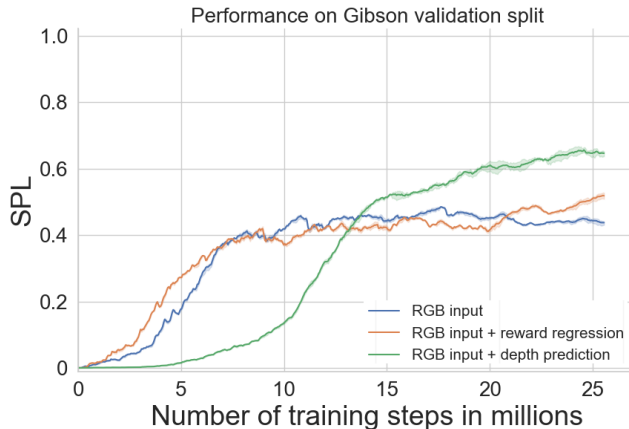


Figure 2. Table 2 in main paper compares the baseline with agents equipped with auxiliary tasks from [2, 1] over the intervals 5, 10, 15, 20, 25 million steps. Here we provide complete SPL curves for the same over the entire training cycle.

to calculate cross-entropy loss.

3. Learning Curves

Table 1 in main paper discusses agent performance on two navigation metrics - SPL and success for RGB and Depth agents. Fig. 3 and Fig. 4 show the corresponding SPL and success curves respectively on Gibson validation split. We find that our auxiliary tasks contribute more towards improving agent’s path efficiency rather than increasing the number of successful episodes. They also are more effective for RGB agents than Depth.

4. Comparison with state-of-the-art

For presentational clarity in introduction section of main paper, Figure 1 only compares our best task-equipped agents with state of the art DD-PPO agents for RGB input. Fig. 5 shows a more comprehensive comparison between our task-equipped agents and DD-PPO agents for RGB and Depth inputs. Our best auxiliary tasks for RGB and depth achieves a $> 4\times$ speed up in terms of sample efficiency. For RGB input, all our agents outperform DD-PPO agents by a good margin while gains for depth input are a bit lower. As discussed in main paper, DD-PPO agents employ much complex models and are trained using a distributed RL framework. Our simpler task-equipped depth agents still perform as well as DD-PPO agents which makes this an interesting result.

5. Existing work on auxiliary tasks

The main paper compares agents equipped with tasks from [1, 2] with Habitat baseline agents in Table 2. We have provided the complete SPL validation curves over the entire

	% of predictions within range (\uparrow)		
	< 4m	< 2m	< 1m
Baseline agent	57.2	51.0	48.4
+ Depth Pred.	71.3	66.9	62.5
+ Inv. Dyn	59.1	52.3	50.6
+ Remain. Path.	80.1	76.6	74.5

Table 1. Percent of samples for which the decoder predictions fall within a radius of 4m, 2m and 1m. The decoder is most accurately able to extract the localization information using the encoder trained with remaining path length prediction task.

training cycle in Fig. 2. This provides a fine-grained understanding of how learning is affected by including these auxiliary tasks.

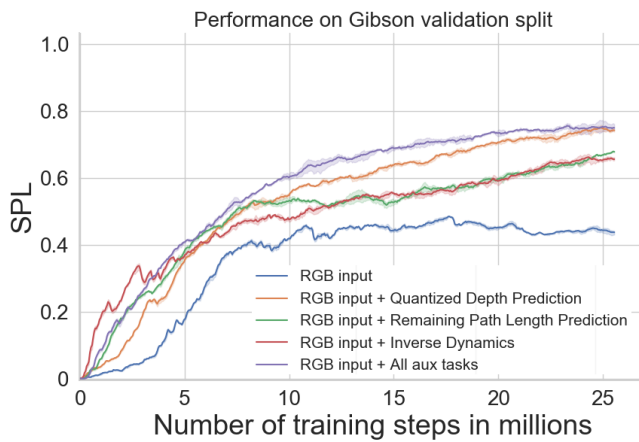
6. Decoding Position from Agent Representations

Tab. 1 shows the performance of position decoders utilizing the state representation h_t of various frozen encoders to predict the current coordinates.

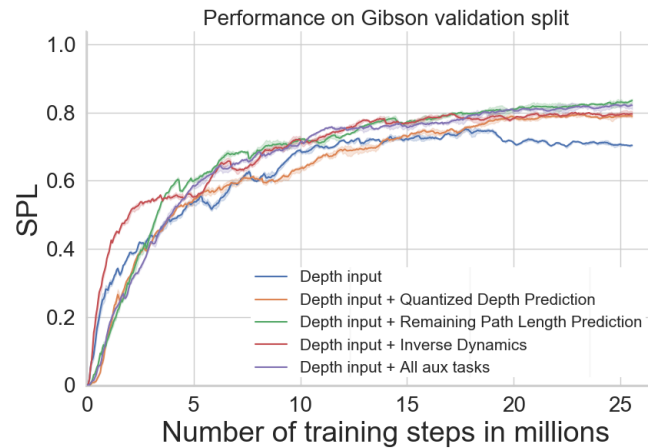
7. Performance on Auxiliary Objectives

So far, we have seen the improvement in navigation skill, in terms of SPL and success metrics, due to additional auxiliary tasks. Further, we analyze how well the agents performed on these objectives themselves (see Fig. 6). We note that learning curves for inverse dynamics and auxiliary path length prediction share a common shape where loss drops quickly before increasing, then reducing again over time. As both these losses depend on the actions the agent takes, this reflects a shift in agent behavior over training – initially performing random actions and eventually navigating competently. For example, when the agent does not make significant progress to the goal early in training, path length prediction is a somewhat easy task (low loss) based on dataset priors about path complexity. Later when the agent is better able to move around the world and towards the goal, this task becomes significantly harder.

We compare reduction in these auxiliary objectives for task-equipped RGB and Depth agents. Perhaps unsurprisingly, we find that Depth agents are better at solving these objectives than RGB agents. Depth agents obviously have an advantage at depth prediction, but also perform better at inverse dynamics and path length prediction. Given that both these tasks build on navigational skills, this result is in line with depth agents’ higher baseline performance. Despite performing worse on the auxiliary losses themselves, we do find that RGB agents get more improvement from optimizing them (as in Fig. 3).

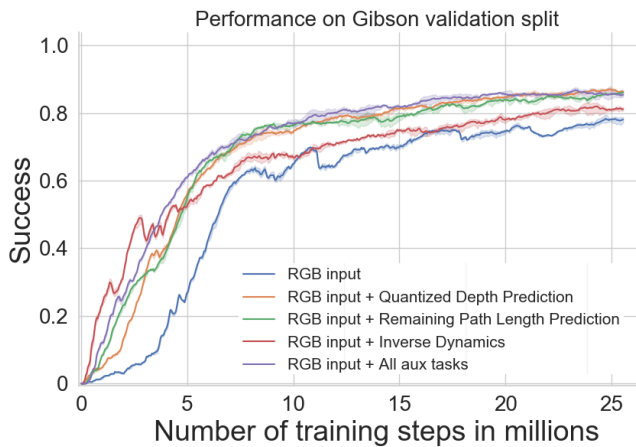


(a) Validation SPL over training for RGB agent

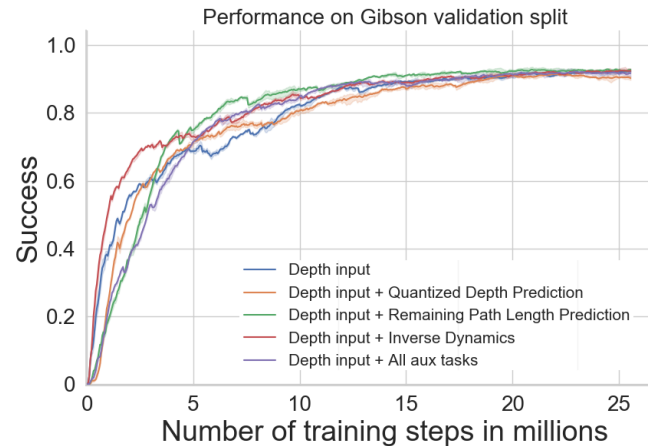


(b) Validation SPL over training for Depth agent

Figure 3. SPL on Gibson validation set for RGB (left) and Depth (right) agents as a function of number of training steps. We compare baseline agents with our auxiliary task-equipped agents. For RGB input, our agents consistently outperform the baseline from the start of training. Our depth agents have a slower start but outpace the baseline model after 20 million steps.



(a) Validation Success over training for RGB agent

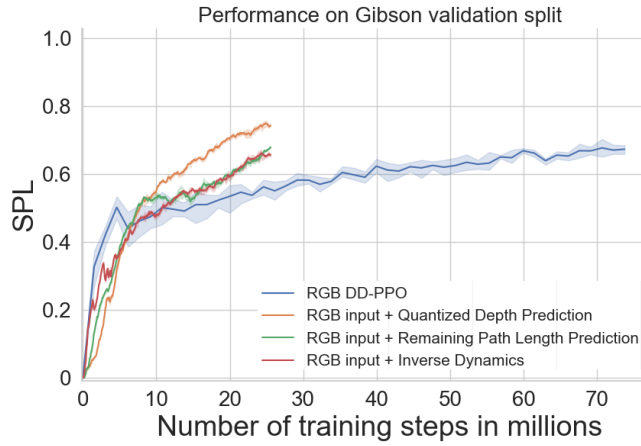


(b) Validation Success over training for Depth agent

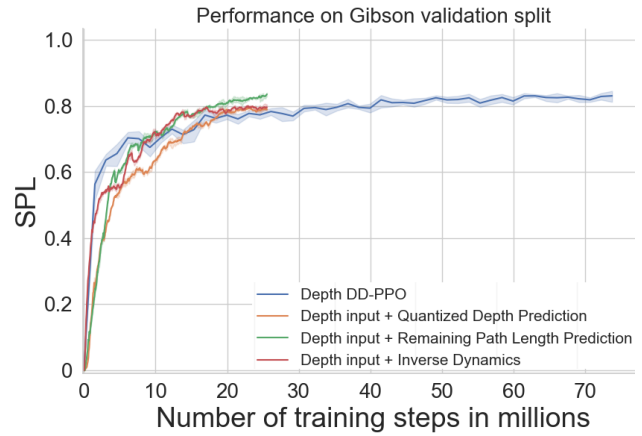
Figure 4. Success on Gibson validation set for RGB (left) and Depth (right) agents as a function of number of training steps. We compare baseline agents with our auxiliary task-equipped agents. For RGB input, our agents consistently outperform the baseline from the start of training but not necessarily with a very large margin. For depth input, the success curves seem to eventually converge to same value for all the agents.

References

- [1] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.
- [2] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*, 2016.
- [3] 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. 2020.
- [4] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

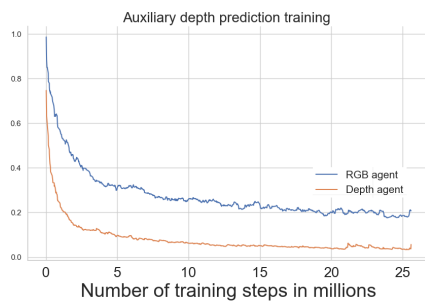


(a) Validation SPL over training for RGB agent

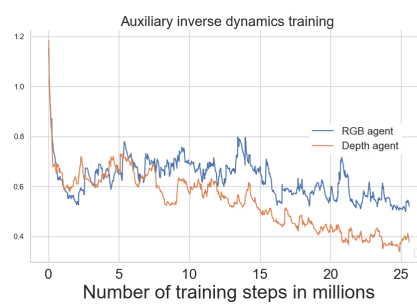


(b) Validation SPL over training for Depth agent

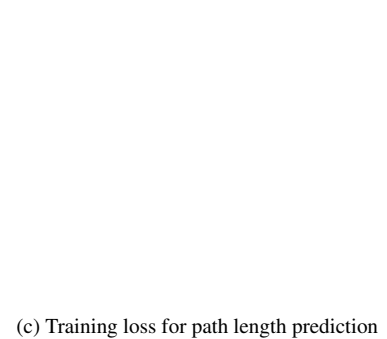
Figure 5. SPL on Gibson validation set for RGB (left) and Depth (right) agents as a function of number of training steps. We compare state-of-the-art DD-PPO[3] agents with our auxiliary task-equipped agents. For RGB input, our agents outperform DD-PPO agents convincingly. Our depth agents are on par with DD-PPO agent which uses complex model and is trained with distributed RL framework.



(a) Training loss for depth prediction



(b) Training loss for inverse dynamics



(c) Training loss for path length prediction

Figure 6. Training loss (y-axis) of task-equipped RGB and Depth agents as a function of training time. We examine how well are the RGB and Depth agents able to minimize the error on all three auxiliary tasks. It can be seen that Depth agent is a better learner of auxiliary tasks than its RGB counterpart.