OPTIMIZING SEMANTIC MAPS PREDICTION FOR INDOOR NAVIGATION

SIDDHARTH GOEL [SIGOEL@SEAS]

ABSTRACT. In this work we deep dive into one of the novel approaches of object goal navigation - Learning2Map (L2M) algorithm. We explore the various sub-components of training an autonomous agent to navigate to different objects in an indoor environment. The major components are predicting a semantic map of the environment, goal selection policy, and navigating to the goal using the map. We investigate the semantic map prediction module of L2M in detail and propose changing the loss function to Focal Loss, using the weighted loss functions, and incorporating temporal information by using an LSTM layer in the network to improve upon the existing results of L2M. Finally, we conduct experiments to validate our hypothesis and demonstrate improved results across all metrics.

1. INTRODUCTION

An autonomous agent in an indoor environment will make life much easier for elderly and disabled people if they could just ask an autonomous agent to go and fetch the laptop from the bed or check if the gas stove is on.

To perform such tasks an autonomous agent needs to be good at (i) Object Recognition, (ii) SLAM (Simultaneous Localization and Mapping), and (iii) navigating to that object by taking the shortest route without running into obstacles (navigation). Advanced object recognition models such as ResNet have been doing a decent job of identifying objects. A novel framework *Learning to Map* (L2M) proposes a unique approach for predicting a semantic map of the environment using raw visual inputs and navigating to an object goal in that environment. The L2M algorithm comprises of two main parts -

(i) an ensemble of hierarchical segmentation models to predict occupancy and semantic regions
(ii) an uncertainty (information gain) based goal selection policy to navigate to the target class object.

In this work we focus on improving the semantic map prediction results of the L2M paper. We perform three modifications to the existing L2M algorithm -

(i) Modify the existing loss function to Focal Loss[5] for improved object detection
(ii) Tune the weights for spatial loss and object loss, and
(iii) Modify the network architecture to incorporate temporal information.

We simulate the above experiments on the AI Habitat simulator [9] for the object navigation task in scenes from the Matterport3D dataset [1] and observe better performance of the L2M algorithm across different metrics.

The rest of the paper is organized as follows. In section 2, we briefly describe the key points of the original L2M paper. Section 3 presents a brief survey on the some of the other popular object goal navigation techniques. Then we present our modifications to the L2M approach and share the experiments and the corresponding results in Section 4 and 5. Finally, we explain our results and mention some potential future work.

2. BACKGROUND

The L2M algorithm can be divided into three sub-components namely - (i) semantic map predictor, (ii) goal-selection policy, and (iii) point-goal local navigation policy.

2.1. *Semantic map prediction*

The semantic map prediction is achieved through a hierarchical segmentation as shown in Figure 1. The semantic-map of the environment m_t is predicted by passing the RGB-D observations through an ensemble of models $\mathbf{G} = \{G_0, ...G_n\}$. Each model in the ensemble $\mathbf{G}$ is a combination of two Unets F_i^o and F_i^s. The model F_i^o predicts the occupancy space $\{$*occupied, free, unknown*$\}$ and the model F_i^s predicts the top-down egocentric semantics of unobserved areas $\hat{m}_t$. Predicting the semantic map is a two-step procedure. In the first step, a UNet [8] F_i^o is used to predict if a particular pixel in space is *occupied, free,* or *unknown* using the ground projection of the depth observations at time t. This gives the occupancy prediction $\hat{p}_t \in \mathbf{R}^{|\mathbf{C^o}| \times \mathbf{h} \times \mathbf{w}}$ as the output.

$$\hat{p}_t = F_i^o(p_t; \theta_i^o) \tag{1}$$

$$\hat{m}_t = F_i^s(\hat{p}_t \oplus \hat{s}_t; \theta_i^s) \tag{2}$$

In the next step, the occupancy prediction $\hat{p}_t$ is concatenated with a ground projection of the semantic segmentation $\hat{s}_t$ of the RGB observation at time t. The concatenated result is then provided as input to another UNet F_i^s to predict the final semantic map crop $\hat{m}_t$ at time t. The segmentation model to predict $\hat{s}_t$ is pre-trained separately.

Apart from the class probabilities predicted by each model G_i and the mean of these probabilities over the ensemble, the per class uncertainty is also estimated using the ensemble predictions.

$$u_t = \text{Var}_E G(p_t, \hat{s}_t) := \frac{1}{n} \sum_{i=0}^{n} (G_i(p_t, \hat{s}_t) - G(p_t, \hat{s}_t)^2) \tag{3}$$

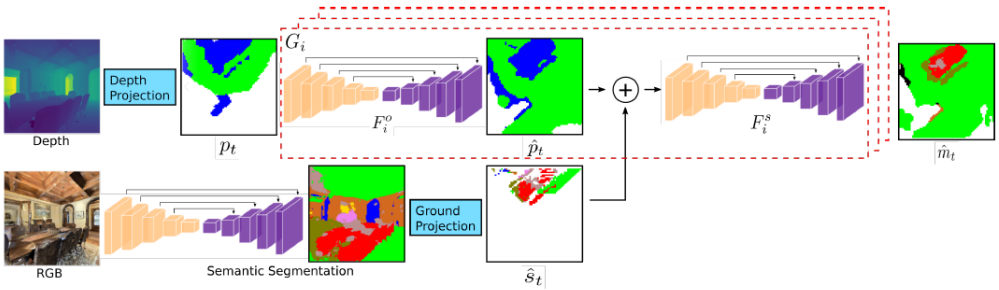

FIGURE 1. Predicting semantic maps from RGB-D images using an ensemble of models. Each model in the ensemble consists of two encoder decoder models for predicting occupancy and object semantics.

Each model G_i in the ensemble G is initialized with random weights θ_i^o and θ_i^s and trained end-to-end for both occupancy and semantic prediction using pixel-wise cross-entropy losses.

$$L^o = -\sum_c^{C^o} p_{t,c} \log \hat{p_{t,c}} \qquad L^s = -\sum_c^{C^s} m_{t,c} \log \hat{m_{t,c}} \tag{4}$$

where $p_{t,c}$ and $m_{t,c}$ are the ground-truth labels and $\hat{p}_{t,c}$ and $\hat{m}_{t,c}$ are the predicted classes for a particular pixel. The overall loss function for each G_i is defined as

$$L_{sem} = \lambda^o L^o + \lambda^s L^s \tag{5}$$

where λ^o and λ^s are the hyperparameters to weight the component loss functions. Finally, the above intermediate results are accumulated in the gloabl maps for occupancy $M^o \in R^{|C^o| \times H \times W}$, semantics $M^s \in R^{|C^s| \times H \times W}$ and uncertainty $M^u \in R^{|C^u| \times H \times W}$ over the episode.

$$M_t^o = \gamma^o(M_{t-1}^o, p_t, x_t); \qquad M_t^s = \gamma^s(M_{t-1}^s, \hat{m}_t, x_t); \qquad M_t^u = \gamma^u(M_{t-1}^u, u_t, x_t) \tag{6}$$

where each γ function transforms the local egocentric region to geocentric with respect to the relative pose x_t of the agent, and aggregates the information into the respective map. The observed regions of the map are updated with zero uncertainty.

2.2. *Goal Selection Policy*

After accumulating the RGB-D observations of the agent in the global semantic map, the next step is to select the goal location of the agent at each time step t. The results of the original paper suggest that we want to select the goal location at which the model can predict the target class with high confidence *i.e.* low variance.

Therefore, we need to consider the following two quantities. Firstly, the probability estimate $G^c(p_t, \hat{s}_t)$ of the target class c for each map location - $M_{t,c}^s$. Secondly, the uncertainty estimate $\text{Var}_E G^c(p_t, \hat{s}_t)$ of the target class c at each map location $M_{t,c}^u$.

Different combinations of the probability and the uncertainty estimate (shown in table 1) are considered in the original paper. The best results were found for the strategy which selects goals (location on the map) based on high probability estimate $G^c(p_t, \hat{s}_t)$ and low uncertainty $\text{Var}_E G^c(p_t, \hat{s}_t)$. This combination indicates that the model has high confidence of the presence of target class at the selected location.

TABLE 1. Goal selection strategy options

$G^c(p_t, \hat{s}_t)$	$\text{Var}_E G^c(p_t, \hat{s}_t)$	**Interpretation**
High	**Low**	Found target with high confidence
High	High	Found target with low confidence
Low	High	Not certain that the target is found
Low	Low	Target not found with high confidence

2.3. *Local Policy*

Once the goal is selected, a deep reinforcement learning, state-of-the-art algorithm for point-goal navigation task, called DD-PPO [11] is used to navigate the agent from its current location to the selected goal location in the environment by the most optimal route. The DD-PPO model takes as input the egocentric depth observation and the current goal converted from a 2D map coordinate to a relative distance and heading and outputs the next navigation action for the agent at each time-step t.

3. RELATED WORK

Modular learning approach: The methods [2, 3] perform the overall navigation tasks by using separate modules for semantic map prediction and goal selection policy. This allows for using existing pre-trained models for object detection and object segmentation for semantic map prediction. These maps are then used to explore the environment efficiently based on the goal object category. The modular approach has been shown to perform better than end-to-end approaches as it leverages structural

regularities of indoor environments, is robust to state-estimation errors, and is more sample-efficient. L2M also comprises of a modular structure with separate semantic map prediction and goal navigation policy modules.

Semantic Mapping: A good survey on visual SLAM techniques can be found in [4]. Some of the more relevant works which use learning-based computer vision models are [12, 6]. The L2M algorithm leverages a pre-trained image segmentation model along with an active exploration strategy enabling it to predict in both observed and unobserved areas of the map.

Uncertainty Estimation: One of the key approaches of L2M is to exploit the uncertainty of the model for active exploration during training and target driven exploration at test time. Estimating model uncertainty from the variance between between the output of an ensemble of models and using it as an objective in vision based reinforcement learning was motivated from [7, 10].

4. Approach

In this work, we attempt to improve the existing L2M approach in the following ways.

(i) *Use Focal Loss in place of Cross-Entropy Loss*

The L2M paper uses Cross Entropy (CE) Loss as the loss function for both the occupancy prediction F_i^o and semantic map prediction F_i^s network. Intuitively, we know that we have a class imbalance in the observations of the agent. Majority of the frames observed by the agent contain objects such as floors and walls whereas the number of images containing objects to be detected such as chair, table, sofa, fridge, bed etc is comparatively very small.

As a result, even a small CE loss such as 0.2 for the correctly identified walls, floors, and free space (easy negatives) overwhelm the loss function and the network is barely able to train on frames containing actual objects (hard positives).

To counter the effect of extreme class imbalance in the presence of exponentially more number of easy negatives and lesser hard positives, we replace the CE loss with the focal loss. The focal-loss achieves this by multiplying a modulating factor $(1 - p_t)^\gamma$ to the CE loss as shown in (8).

The cross-entropy loss is defined in (7) and the focal-loss is defined in (8) where N is the number of classes and γ is a tunable hyper-parameter. Also, notice that when $\gamma = 0$ the focal-loss is equal to cross-entropy loss.

$$\text{CE Loss} = -\sum_{i=1}^{N} y_i \log(p_i) \tag{7}$$

$$\text{Focal Loss} = -\sum_{i=1}^{N} y_i (1 - p_i)^\gamma \log(p_i) \tag{8}$$

As a result, if an easy example is correctly predicted with probability $p_i = 0.9$ the loss is weighted down by multiplying $(1 - 0.9)^\gamma$ by the focal loss. For example, if $\gamma = 2$ then this loss will be 100 times smaller than the cross-entropy loss. On the other hand, a hard example with a predicted probability of 0.2 is given a weight 0.64. In summary, the correct easy example was multiplied by a factor of 0.01 whereas a mis-classified hard example was multiplied by a factor of 0.64. Therefore, countering the effect of disproportionate number of easy and hard examples.

(ii) ***Tune the weights for spatial loss and object loss***

The total loss for the semantic map prediction model is the sum of the weighted losses corresponding to the occupancy prediction model (L^o) and the semantic prediction model (L^s) and can be denoted as — $\mathbf{L_{sem}} = \lambda^{\mathbf{o}}\mathbf{L^o} + \lambda^{\mathbf{s}}\mathbf{L^s}$.

Though the L2M paper proposes the idea of weighting the two losses differently but the values for λ^o and λ^s are kept as 1 for all the experiments. We fine-tune these hyper-parameters to improve the overall performance of the model. Going by the same intuition as for using focal-loss, we believe that the observations comprising objects are much less in number as compared to their counter observations. Therefore, the weight for the object/semantic loss should be much higher than the weight for the spatial/occupancy loss. Using a higher weight for L^s *i.e.* ($\lambda^s >> \lambda^o$) makes the model put more emphasis on identifying objects and does not let the high frequency samples dominate the loss function.

(iii) ***Incorporate temporal information in the map prediction model by using LSTM layer***

A Recurrent Neural Network (RNN) is known to be good at modelling sequence data. In the object goal navigation scenario, each episode is a sequence of observations. An agent can do a much better job if it can store information such as objects table and chairs are usually placed together or a refrigerator is mostly closer to an oven as opposed to a sofa.

The L2M paper only uses a ResNetUnet architecture for both occupancy and semantic map prediction. We hypothesize that introducing an LSTM layer in both these networks will help the network to remember the past observations in an episode and thereby improve its performance.

5. Experimental Results

We implemented the L2M code along with our proposed changes. The implementation details of the network are mentioned in table 4 along with the habitat simulator configuration in table 5 in the appendix section.

Both training and validation were performed on the Matterport3D (MP3D) [1] dataset using the Habitat [9] simulator. MP3D contains reconstructions of real indoor scenes with large appearance and layout variation, and Habitat provides continuous control for realistic agent movement. The standard train/val split, which contains 56 scenes for training and 11 for validation, is used for experimentation. During testing, the model is evaluated on 17900 test examples collected in the 11 validation scenes which had not been observed during training and the results are compared to the original L2M paper results.

The performance of semantic map prediction model is evaluated using the three common image segmentation metrics - F1 score, Intersection over Union (IoU), and Accuracy. F1 score keeps a check of the true performance since there is a huge class

imbalance across class labels. IoU is one of the standard metrics for the semantic segmentation task. Lastly, accuracy is used as the baseline metric.

A qualitative example of a predicted semantic map of a bedroom scene is shown in figure 2.

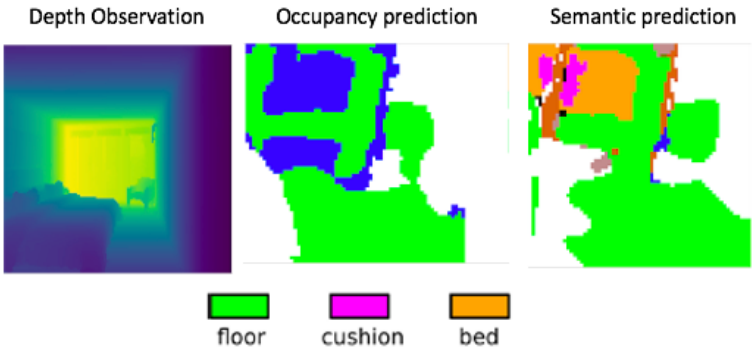

FIGURE 2. Semantic map prediction results of a bedroom scene

Experiment 1: *Selecting the optimal weights for occupancy and object detection loss function*

We conducted experiments by training different models for different values of λ^o and λ^s. Initially, we started with both the weights λ^o and λ^s equal to 1 for the purpose of replicating the L2M paper results. This also served as a baseline to compare the results using different weights.

In the following experiments, we started to increase the weight corresponding to the semantic map prediction model in factors of 10. In other words, the ratio of the two losses - $\frac{\lambda^s}{\lambda^o}$ was increased in the order 1, 10, 100, and 1000, putting this much times more weight on the semantic loss.

From figure 3 we observe the following things. Initially, there is no considerable improvement in the mean F1 scores for semantic/object and occupancy/spatial as the weights ratio — $\frac{\lambda^s}{\lambda^o}$ is increased to 10. The mean F1 score metric is almost same as the baseline for both semantic and occupancy models. Upon gradually increasing the weights ratio to 100 a drastic improvement in performance of the semantic map model is observed along with a very small improvement in the occupancy model mean F1 score. Finally, as the weights ratio is increased beyond 100 we see a dip in the mean F1 score for both semantic and occupancy models.

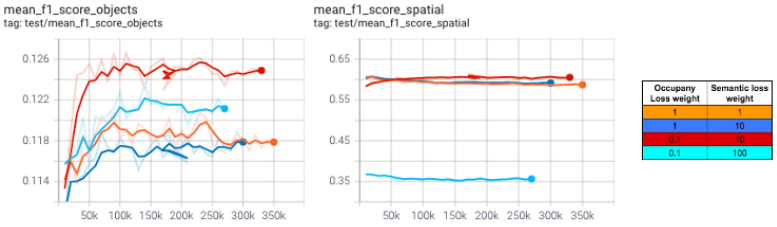

FIGURE 3. Mean F1 score for spatial and object prediction for different loss weights

The occupancy mean F1 score remains almost same for weights ratio 1, 10, and 100. This indicates that the occupancy prediction is a relatively easier problem to solve and putting a 100 times lower weight on the occupancy loss than its counterpart doesn't affect the performance of the model on predicting occupancy.

These observations are in line with our intuition. Increasing the weight for the semantic loss makes the model learn object detection better since it is a relatively harder problem than occupancy prediction. At the same time, increasing the weight beyond 100 leads to a dip in performance as the model is focusing entirely on object detection and the learning corresponding to occupancy prediction is barely taking place. We must recall that the semantic map model takes the output of occupancy prediction as input. Therefore, it is important for the end-to-end model to maintain a balance and do well on both the sub-components.

For the purpose of ablation studies, we also trained models by giving more weights to occupancy prediction loss λ^o than semantic prediction loss λ^s. As expected, we observed worse performance than the baseline for these models.

Experiment 2: *Selecting the best loss function for occupancy prediction and semantic prediction models*

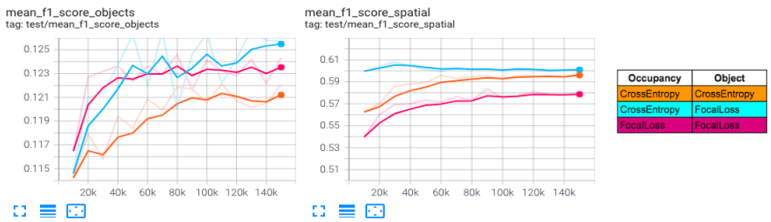

FIGURE 4. Mean F1 score for spatial and object prediction for different loss functions

We replaced the CE loss for semantic prediction model by focal-loss and observed a considerable improvement in the mean F1 scores for semantic/object and occupancy/spatial as compared to the baseline L2M model results.

The focal-loss prevents the high number of easy examples from overwhelming the loss function and propagates a balance in training of easy negatives and hard positives. This is very well confirmed by our results in figure 4.

Next, we also replaced the CE loss with focal-loss for the occupancy prediction model along with semantic prediction model. Again an improvement in the performance of the mean F1 score for objects was observed as compared to the L2M baseline. But, the performance was not optimal for the mean F1 score for spatial/occupancy. We believe the following reasons account for these observations.

The spatial space has just three labels - {*free, occupied, unknown*}. Also, the three labels are quite proportionate to each other in the frames. The modulating factor of the focal-loss weighs down the spatial space loss more than required and hurts the performance of the model.

In figure 4 we observe that the mean F1 score across models is best for the combination of CE loss (occupancy prediction) and focal loss (semantic prediction). The weights of the two losses were fine-tuned in a separate experiment and the values $\lambda^o = 0.1$ and $\lambda^s = 10$ give the best performance metrics.

Experiment 3: *Add an LSTM layer to the network architecture*

Finally, we performed the third set of experiments by modifying the network architecture (figure 6) to capture temporal information. We added an LSTM layer between the encoder and decoder of the UNet to achieve this.

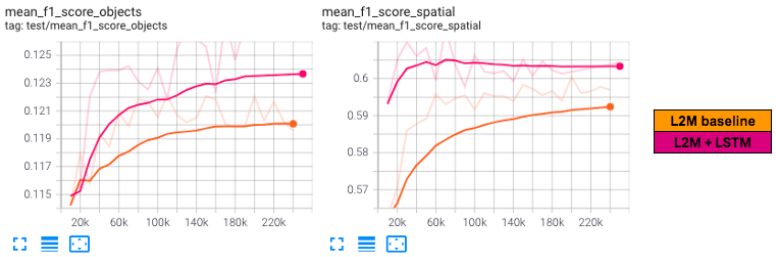

FIGURE 5. Mean F1 score for spatial and object prediction with LSTM layer

The results of this experiment in figure 5 indicate that adding an LSTM layer does improve the performance of the baseline L2M model. As per our earlier hypothesis, we believe that using an LSTM layer helps the model in building an episodic memory and retain structural information better *e.g.* pair of objects like table and chairs, sofa and cushion, or fridge and oven are usually placed together. In case of occupancy model, as well the spatial information *e.g.* the continuous nature of floor and walls are captured better. This led to an improved mean F1 score for both semantic and occupancy model as compared to the baseline.

It is also noted that the mean F1 score for occupancy prediction curve is almost a flat line across experiments after 60,000-70,000 steps. This indicates that occupancy prediction model is easier to train and reaches its optimal state faster as compared to the semantic prediction model.

Finally, we combined the results of our previous experiments and found the best performing model configuration as mentioned in table 2. The final metric scores for this configuration are recorded in table 3.

TABLE 2. Best performing model loss function and weights configuration

Parameter	Value
Occupancy Loss	Cross-entropy
Semantic Loss	Focal Loss
λ^o	0.1
λ^s	10
LSTM layer	yes

TABLE 3. Semantic map prediction results

Occupancy Prediction			
Method	**Acc(%)**	**IoU(%)**	**F1(%)**
L2M	65.2	45.5	61.9
L2M+FocalLoss+LSTM	**66.0**	**46.5**	**63.0**
Semantic Prediction			
L2M	**31.2**	20.1	30.5
L2M+FocalLoss+LSTM	29.0	**21.1**	**31.7**

6. DISCUSSION

In this work we improved the existing performance of the L2M algorithm for predicting semantic maps of an unseen indoor environment. We first fine-tuned the weights of the two loss functions to put more weight on the semantic prediction loss function. Intuitively, predicting whether a pixel is free or occupied is an easier task for a model as compared to predicting the right object class if the pixel is occupied. Putting more weight on the object detection loss function also countered the effect of class imbalance and balanced the learning of both the networks.

The next modification was to change the loss function from cross-entropy loss to focal-loss for semantic prediction model. This was done to counter the overwhelming effect of the high number of easy examples on the loss function and the gradients.

Finally, we incorporated temporal information in the model by adding an LSTM layer. As a result, we observed better performance across all the three relevant metrics by using focal loss for object detection, optimal weights for the two loss functions, and incorporating the temporal information by using LSTM layer in the network.

In future work we intend to add weights for object classes to the loss function. Including class weights will check the class imbalance further and should lead to a more improved model performance. The challenge in this case lies in identifying the appropriate weights corresponding to each object class in the train set. Further, the number of layers in LSTM can be increased and a biLSTM can also be experimented with.

REFERENCES

[1] Angel Chang et al. "Matterport3D: Learning from RGB-D Data in Indoor Environments". In: *International Conference on 3D Vision (3DV)* (2017).

[2] Devendra Singh Chaplot et al. "Learning to Explore using Active Neural SLAM". In: *CoRR* abs/2004.05155 (2020). arXiv: 2004.05155. URL: https://arxiv.org/abs/2004.05155.

[3] Devendra Singh Chaplot et al. *Object Goal Navigation using Goal-Oriented Semantic Exploration*. 2020. arXiv: 2007.00643 [cs.CV].

[4] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J.M. Rendón-Mancha. "Visual simultaneous localization and mapping: a survey". In: *Artif Intell* Rev 43, 55–81 (2015). URL: https://doi.org/10.1007/s10462-012-9365-8.

[5] Tsung-Yi Lin et al. *Focal Loss for Dense Object Detection*. 2018. arXiv: 1708.02002 [cs.CV].

[6] Lingni Ma et al. "Multi-View Deep Learning for Consistent Semantic Mapping with RGB-D Cameras". In: *CoRR* abs/1703.08866 (2017). arXiv: 1703.08866. URL: http://arxiv.org/abs/1703.08866.

[7] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. "Self-Supervised Exploration via Disagreement". In: *CoRR* abs/1906.04161 (2019). arXiv: 1906.04161. URL: http://arxiv.org/abs/1906.04161.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.

[9] Manolis Savva et al. "Habitat: A Platform for Embodied AI Research". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[10] H. S. Seung, M. Opper, and H. Sompolinsky. "Query by Committee". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1992, pp. 287–294. ISBN: 089791497X. DOI: 10.1145/130385.130417. URL: https://doi.org/10.1145/130385.130417.

[11] Erik Wijmans et al. *DD-PPO: Learning Near-Perfect PointGoal Navigators from 2.5 Billion Frames*. 2020. arXiv: 1911.00357 [cs.CV].

[12] Yuke Zhu et al. "Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning". In: *CoRR* abs/1609.05143 (2016). arXiv: 1609.05143. URL: http://arxiv.org/abs/1609.05143.

APPENDIX

TABLE 4. L2M semantic map prediction model details

Parameter	Value
Epoch	40
Batch Size	12
Episode Length	10
Initial Learning Rate	2e-4
Learning Rate decay	0.99
Optimizer	Adam
no. of spatial classes	3
no. of object classes	27
image crop size	64 x 64

TABLE 5. Habitat Simulator configuration

Parameter	Value
Max. episode steps	500
Sensors	['RGB', 'Depth', 'Semantic']
Height of agent(m)	1.5
Task type	ObjectNav-v1
Possible Actions	['Stop', 'Move Forward', 'Turn Left', 'Turn Right']
Move forward distance (cm)	25
Turn left/right angle	10°

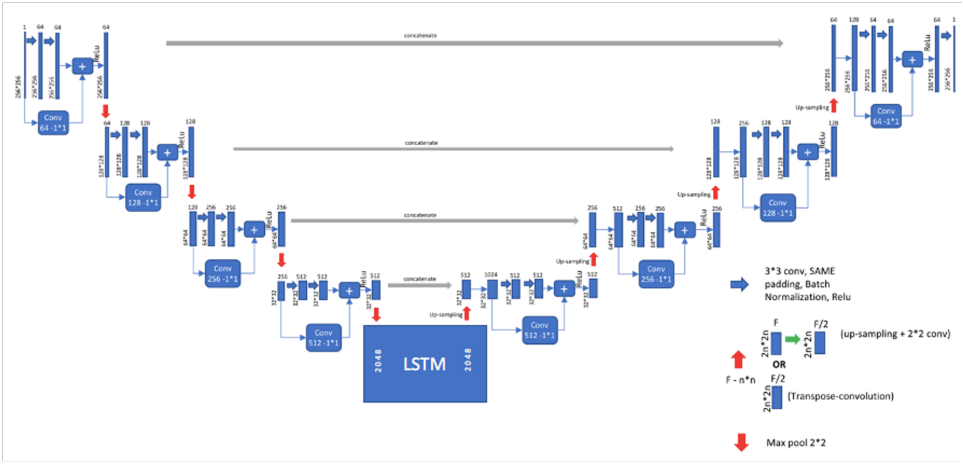

FIGURE 6. Model architecture with LSTM layer between the encoder and decoder