

# Fast SceneScript: Fast and Accurate Language-Based 3D Scene Understanding via Multi-Token Prediction

## Supplementary Material

Sec. A provides additional comparisons, along with extended ablation studies, distribution analyses, and failure cases. Additional implementation details are provided in Sec. B. Additional related work is discussed in Sec. C. Details of the ASE dataset splits are given in Sec. D.

### A. Additional Results

#### A.1. Results with More Heads

Tab. S1 illustrates the impact of using more heads in our Fast SceneScript for layout estimation. We observe a drop in accuracy and no obvious latency advantage when using 12 or 16 heads in Fast SceneScript.

#### A.2. Qualitative Results on Structured3D

Fig. S1 shows qualitative results on Structured3D [20]. Compared to SceneScript and SceneScript + MTP, our Fast SceneScript yields more complete and accurate layouts.

#### A.3. Ablation Study for Layout Estimation

This section presents additional ablation study on ASE [2].

**$\epsilon$  for the scoring-based methods:** Tab. S2 reports results of scoring-based token filtering methods under different thresholds  $\epsilon$  during inference. The results indicate minor variations across different thresholds. To balance accuracy and speed, we set the default thresholds to 0.90, 0.30, and 0.50 for CGD, ProductThre, and SoftmaxThre, respectively.

**$\tau$  for the supervision of confidence branch:** Tab. S3 illustrates the results of CGD in our Fast SceneScript trained with different values of  $\tau$ , *i.e.*, 0, 2, 5. Among these, Fast SceneScript (CGD) with  $\tau = 2$  achieves the highest F1-Score and is therefore selected as the final configuration.

**Softmax score distribution:** Fig. S2 provides detailed softmax score distribution for 3 classes: wall, window, and door. In (a), for *window*, many tokens generated by the MTP model have softmax scores below 0.5. As a result, the method with *SoftmaxThre* rejects these tokens. In contrast, the SSD in (b) and CGD in (c) also accept some of these tokens. This shows the effectiveness of the proposed CGD strategy compared to naive decoding using softmax probabilities as the reliability of a token.

**Supervision strategies for confidence branch:** Tab. S4 investigates the impact of different training losses for confidence branch in CGD. Two settings are considered: (1) Generating the confidence label based on the consistency with the token  $t_{k+i}^1$  from the first head (our default setting, *Fast SceneScript (CGD)*), (2) Generating the confidence

label based on the consistency with its ground truth, *i.e.*, *Fast SceneScript (CGD, GT)*. While both strategies achieve comparable accuracy, the default setting is faster than *Fast SceneScript (CGD, GT)*. We argue that our default setting not only learns token confidence but also captures the inherent uncertainty among different heads. By comparing with predictions rather than with ground truth, the confidence branch is encouraged to model internal agreement and disagreement, which also represents the model uncertainty. This makes confidence estimation more robust and informative. As a result, the network accept more tokens during inference, reaching faster speed.

#### A.4. Analyses for Object Detection

This section provides additional analysis for object detection.

**Distribution of accepted tokens per inference:** Fig. S3 reports the distribution of the number of accepted tokens per decoder inference with our Fast SceneScript for object detection. It shows that our Fast SceneScript with CGD or SSD accepts all 8 tokens in  $\sim 70\%$  of decoder runs. This aligns with our observation on the layout estimation task.

**Softmax score distribution:** Fig. S4 shows the softmax score distribution of the accepted tokens in Fast SceneScript. Only  $\sim 40\%$  of accepted tokens have scores above 0.9, compared to  $\sim 70\%$  in layout estimation (see main paper). We attribute the high softmax confidence in layout estimation to inherent token redundancy and predictable layout structure, *e.g.*, neighboring walls often share a corner. In contrast, there is no redundancy in 3D object detection tokens, yielding lower softmax scores.

#### A.5. Failure Cases

Fig. S5 presents failure cases of our method on the synthetic ASE [2] and real-world SceneCAD [1] datasets. It can be seen that Fast SceneScript (SSD) may struggle in regions where the point cloud is incomplete and in producing accurate layouts for certain non-Manhattan scenes.

#### A.6. SceneScript with Longer Training

Tab. S5 illustrates the results of the baseline SceneScript [2] trained for 60 and 90 epoches on layout estimation. Longer training yields marginal improvement in accuracy.

Table S1. Quantitative comparisons for layout estimation on ASE dataset [2].  $n$  denotes the number of MTP heads.  $\alpha_{val}$  and  $\alpha_{test}$  refer to the average number of tokens accepted per decoder inference on *val* and *test* sets.

Method	$n$	Param ↓	Latency ↓	$\alpha_{val}$ ↑	F1-Score of <i>val</i> set ↑				$\alpha_{test}$ ↑	F1-Score of <i>test</i> set ↑			
					wall	window	door	mean		wall	window	door	mean
SceneScript [2]	1	14.00 M	382 ms	1	0.918	0.880	0.940	0.913	1	0.921	0.881	0.942	0.915
SceneScript [2] + MTP [6]	8	23.67 M	62 ms	8	0.831	0.804	0.885	0.840	8	0.836	0.804	0.886	0.842
Fast SceneScript (SSD)	8	15.05 M	81 ms	7.46	0.914	0.882	0.939	0.912	7.45	0.919	0.882	0.939	0.913
Fast SceneScript (CGD)	8	16.10 M	92 ms	6.29	0.912	0.883	0.938	0.911	6.30	0.918	0.883	0.938	0.913
SceneScript [2] + MTP [6]	10	26.43 M	54 ms	10	0.805	0.776	0.863	0.815	10	0.808	0.774	0.861	0.814
Fast SceneScript (SSD)	10	15.05 M	75 ms	8.97	0.910	0.879	0.937	0.909	8.99	0.915	0.880	0.940	0.912
Fast SceneScript (CGD)	10	16.10 M	89 ms	7.27	0.909	0.880	0.936	0.908	7.27	0.912	0.879	0.938	0.910
SceneScript [2] + MTP [6]	12	29.20 M	49 ms	12	0.792	0.754	0.840	0.795	12	0.800	0.752	0.842	0.798
Fast SceneScript (SSD)	12	15.05 M	75 ms	10.21	0.902	0.877	0.934	0.904	10.11	0.907	0.876	0.937	0.907
Fast SceneScript (CGD)	12	16.10 M	93 ms	7.71	0.902	0.877	0.933	0.904	7.86	0.905	0.877	0.937	0.906
SceneScript [2] + MTP [6]	16	34.72 M	42 ms	16	0.717	0.681	0.794	0.731	16	0.721	0.688	0.794	0.734
Fast SceneScript (SSD)	16	15.05 M	73 ms	12.48	0.898	0.870	0.932	0.900	12.38	0.902	0.870	0.933	0.902
Fast SceneScript (CGD)	16	16.10 M	99 ms	8.60	0.896	0.874	0.934	0.901	8.65	0.898	0.871	0.934	0.901

Table S2. Ablation experiments for layout estimation on the ASE *val* set [2].  $\epsilon$  is the threshold used to determine where to stop in the scoring-based methods. The default setting in our paper is underlined.

Method	$\epsilon$	$n$	Param ↓	Latency ↓	$\alpha_{val}$ ↑	F1-Score ↑			
						wall	window	door	mean
Fast SceneScript (CGD)	0.80	8	16.10 M	85 ms	6.77	0.905	0.878	0.935	0.906
Fast SceneScript (CGD)	0.85	8	16.10 M	88 ms	6.55	0.910	0.879	0.937	0.909
Fast SceneScript (CGD)	<u>0.90</u>	8	16.10 M	91 ms	6.29	0.912	0.883	0.938	0.911
Fast SceneScript (CGD)	0.95	8	16.10 M	97 ms	5.98	0.914	0.883	0.939	0.912
Fast SceneScript (ProductThre [9, 15])	0.20	8	16.10 M	97 ms	5.42	0.905	0.878	0.934	0.907
Fast SceneScript (ProductThre [9, 15])	0.25	8	16.10 M	105 ms	5.04	0.907	0.880	0.938	0.908
Fast SceneScript (ProductThre [9, 15])	<u>0.30</u>	8	16.10 M	112 ms	4.67	0.908	0.881	0.938	0.909
Fast SceneScript (ProductThre [9, 15])	0.35	8	16.10 M	120 ms	4.34	0.909	0.882	0.938	0.910
Fast SceneScript (SoftmaxThre [13])	0.40	8	16.10 M	96 ms	5.49	0.903	0.875	0.931	0.903
Fast SceneScript (SoftmaxThre [13])	0.45	8	16.10 M	106 ms	5.03	0.906	0.878	0.935	0.906
Fast SceneScript (SoftmaxThre [13])	<u>0.50</u>	8	16.10 M	116 ms	4.53	0.908	0.880	0.937	0.908
Fast SceneScript (SoftmaxThre [13])	0.55	8	16.10 M	131 ms	3.57	0.912	0.882	0.939	0.911

## B. Additional Implementation Details

### B.1. Training Details of CGD

Fig. S6 provides details of generating confidence labels during training. Since teacher-forcing is applied during training, multiple heads can predict and supervise each token. For any given token, the first head’s prediction is generally considered more reliable. Therefore, the confidence label  $\hat{c}_{k+i}^j$  for the  $(k+i)$ -th token  $t_{k+i}^j$ , generated by the  $j$ -th head ( $j \in [2, n]$ ), is computed based on its consistency with the token  $t_{k+i}^1$  from the first head. If absolute difference satisfies  $|t_{k+i}^j - t_{k+i}^1| \leq \tau$ , the token  $t_{k+i}^j$  is regarded as

correct and  $\hat{c}_{k+i}^j = 1$ , otherwise  $\hat{c}_{k+i}^j = 0$ .<sup>1</sup>

### B.2. Training Settings

Training employs the AdamW optimizer [10] with an initial learning rate of 0.001 and a batch size of 64. A multi-step linear learning rate scheduler is used. The loss weights  $\lambda_h$  and  $\lambda_c$  are set to 0.8 and 0.5, respectively.

For layout estimation on the ASE dataset [2], the NTP model is trained for 60 epochs. Following [3, 17], the MTP model is initialized from the NTP model, excluding the projection blocks and confidence heads, and trained for

<sup>1</sup>In the main paper, we omit the head index  $j$  for simplicity. The token  $\tilde{t}_{k+i}$  mentioned in the main paper corresponds to  $t_{k+i}^1$  in the supplementary material.

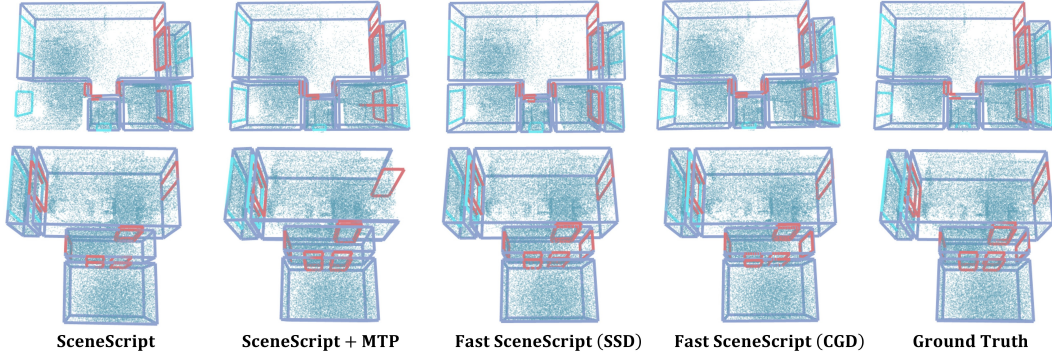


Figure S1. Qualitative results on Structured3D *test* set [20]. The number of MTP heads  $n$  is set to 8. The scene layout generated by Fast SceneScript demonstrates superior accuracy compared to SceneScript [2] + MTP [6].

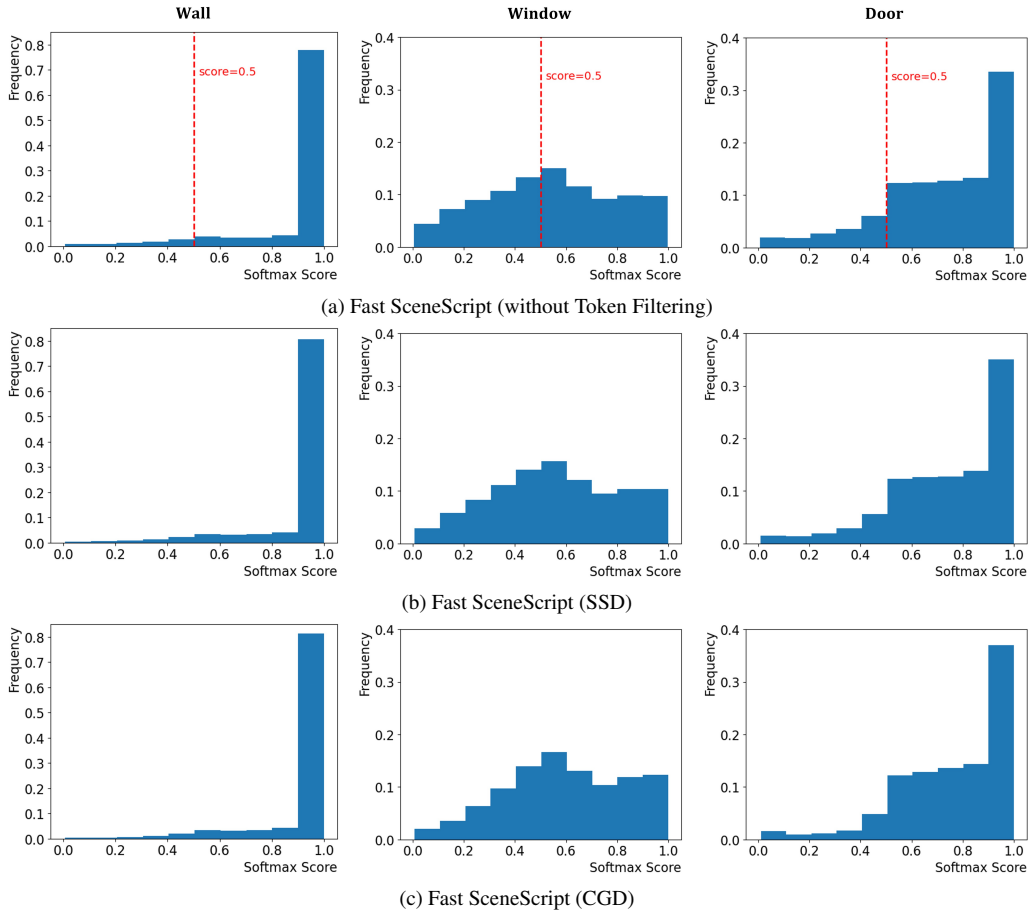


Figure S2. Softmax score distribution of 3 classes. (a): softmax score distributions of all tokens without filtering. (b) - (c): softmax score distributions for accepted tokens. In (a), the red dashed line marks the position where the softmax score equals 0.5. Tokens to the left of this line are rejected by *SoftmaxThre*. In contrast, The SSD in (b) and CGD in (c) can accept tokens with low softmax confidence.

30 epochs. On the Structured3D dataset [20], models are initialized with weights pretrained on ASE [2] and further trained for 600 epochs. Finally, the SceneCAD model is fine-tuned on SceneCAD [1] for 600 epochs.

For object detection on ASE dataset [2], the NTP model is trained for 150 epochs, while the MTP model is trained for 120 epochs using the pretrained NTP model. On SceneCAD [1], models are initialized with weights trained

Table S3. Ablation study for layout estimation on ASE *val* set [2]. The default setting in our paper is underlined.

Method	$\tau$	$n$	Param $\downarrow$	Latency $\downarrow$	$\alpha_{val} \uparrow$	F1-Score of <i>val</i> set $\uparrow$			
						wall	window	door	mean
Fast SceneScript (CGD)	0	8	16.10 M	97 ms	5.94	0.908	0.881	0.935	0.908
Fast SceneScript (CGD)	<u>2</u>	8	16.10 M	91 ms	6.29	0.912	0.883	0.938	0.911
Fast SceneScript (CGD)	5	8	16.10 M	90 ms	6.44	0.912	0.882	0.936	0.910

Table S4. Ablation experiments for layout estimation on the ASE *val* set [2].

Method	$n$	Param $\downarrow$	Latency $\downarrow$	$\alpha_{val} \uparrow$	F1-Score $\uparrow$			
					wall	window	door	mean
Fast SceneScript (CGD, GT)	8	16.10 M	100 ms	5.61	0.913	0.884	0.935	0.911
Fast SceneScript (CGD)	8	16.10 M	91 ms	6.29	0.912	0.883	0.938	0.911

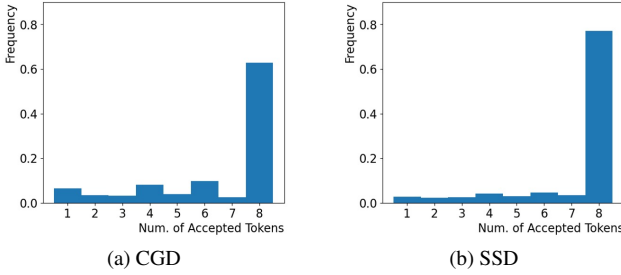


Figure S3. Distribution of the number of accepted tokens per decoder inference with our Fast SceneScript for object detection. It can be seen that both SSD and CGD accept a large number of tokens predicted by the network.

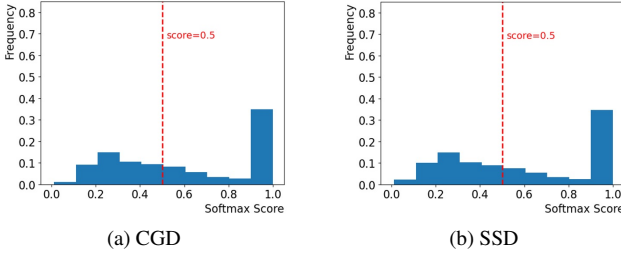


Figure S4. Analysis of softmax scores for the accepted tokens in Fast SceneScript for object detection. The  $x$ -axis shows the softmax scores and the  $y$ -axis indicates their frequency. Our Fast SceneScript also accepts tokens with low softmax scores.

on ASE [2] and trained for 600 epochs.

### B.3. Network Architecture

The architecture details of our shared Token / Confidence Head and Projection Block are presented in Fig. S7. The Token Head and Confidence Head have a similar architecture, including 3 linear layers and 2 ReLU layers. The Projection Block is built from 2 feed-forward blocks, following the Transformer FFN design [16]. Each block consists of 2

linear layers, 1 ReLU layer, 1 Layer Normalization (LN) layer, and shortcut connection between its input and output.

## C. Additional Related Works

This section reviews related work on 3D layout estimation from point clouds.

Scan2BIM [12] detects walls with conventional geometric analysis algorithms, such as One-Point RANSAC Model fitting [5]. FloorNet [8] uses an Integer Programming formulation [7] to convert pixel-wise predictions on layout geometry and semantics into vector-graphics layout. SceneCAD [1] predicts layout planes through a bottom-up pipeline, which generates corners, edges, and planes hierarchically. HEAT [4] detects corners and classifies edge candidates between corners with Transformer [16]. RoomFormer [19] uses Transformer [16] to predict a set of room polygons. Its Transformer decoder uses two-level queries: room-level queries and corner-level queries. SceneScript [2] and SpatialLM [11] reformulate layout estimation as a language modeling problem, achieving state-of-the-art accuracy with a simple and end-to-end manner. SceneScript [2] designs task-specific structured language and trains a specialized small language decoder, while SpatialLM [11] uses a general large language model [14, 18]. Our Fast SceneScript is designed to significantly accelerate inference compared to SceneScript [2], while preserving its accuracy.

## D. Dataset Splits for ASE

This section describes our data splits of the ASE dataset [2].

For layout estimation, the split is defined in *ase\_split/layout\_train.txt*, *ase\_split/layout\_val.txt*, and *ase\_split/layout\_test.txt*.

For object detection, the split is defined in *ase\_split/detection\_train.txt* and *ase\_split/detection\_test.txt*.

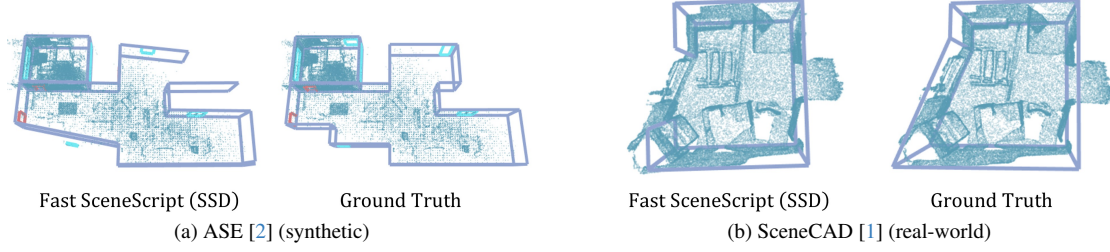


Figure S5. Failure cases on ASE [2] and SceneCAD [1] datasets.

Table S5. Results of the baseline SceneScript [2] with longer training on ASE dataset [2].

Method	Epoch	F1-Score of <i>val</i> set $\uparrow$				F1-Score of <i>test</i> set $\uparrow$			
		wall	window	door	mean	wall	window	door	mean
SceneScript [2]	60	0.918	0.880	0.940	0.913	0.921	0.881	0.942	0.915
SceneScript [2]	90	0.921	0.887	0.946	0.918	0.924	0.888	0.948	0.920

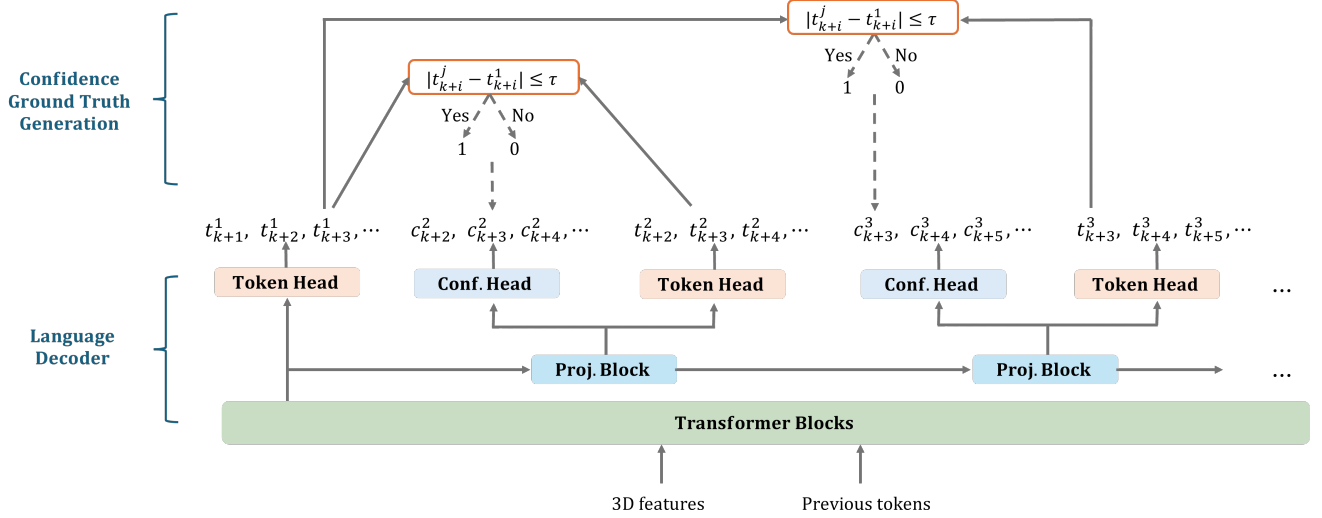


Figure S6. Confidence ground truth generation for CGD. During training, the confidence label  $\hat{c}_{k+i}^j$  for the  $(k+i)$ -th token  $t_{k+i}^j$ , generated by the  $j$ -th head ( $j \in [2, n]$ ), is computed as follows: (1) Calculate the absolute difference between the token from head  $j$  and the token from the first head, *i.e.*,  $|t_{k+i}^j - t_{k+i}^1|$ , (2) Apply thresholding: if  $|t_{k+i}^j - t_{k+i}^1| \leq \tau$ ,  $\hat{c}_{k+i}^j = 1$ , otherwise,  $\hat{c}_{k+i}^j = 0$ . In particular,  $\tau$  is a positive hyperparameter for numerical tokens and is 0 for non-numerical tokens. This figure illustrates examples of generating the confidence labels  $\hat{c}_{k+3}^2$  and  $\hat{c}_{k+3}^3$ .

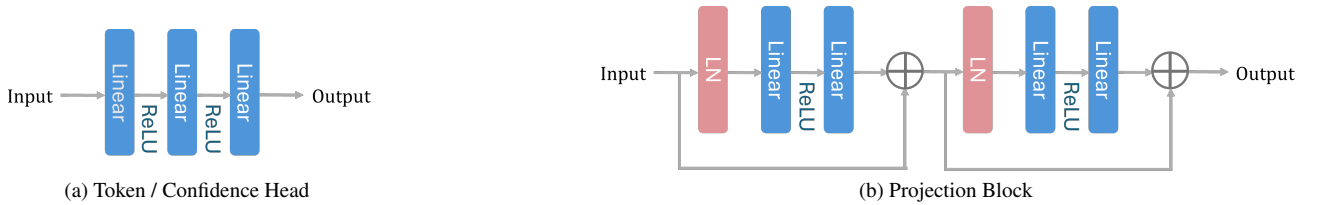


Figure S7. Network details for our shared Token / Confidence Head and Projection Block.

## References

- [1] Armen Avetisyan, Tatiana Khanova, Christopher Choy, Denver Dash, Angela Dai, and Matthias Nießner. Scenecad: Predicting object alignments and layouts in rgb-d scans. In *ECCV*, 2020. [1](#), [3](#), [4](#), [5](#)
- [2] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins,



- Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, et al. Scene-script: Reconstructing scenes with an autoregressive structured language model. In *ECCV*, 2024. [1](#), [2](#), [3](#), [4](#), [5](#)
- [3] Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads. In *ICML*, 2024. [2](#)
- [4] Jiacheng Chen, Yiming Qian, and Yasutaka Furukawa. HEAT: holistic edge attention transformer for structured reconstruction. In *CVPR*, 2022. [4](#)
- [5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981. [4](#)
- [6] Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Roziere, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In *ICML*, 2024. [2](#), [3](#)
- [7] Chen Liu, Jiajun Wu, Pushmeet Kohli, and Yasutaka Furukawa. Raster-to-vector: Revisiting floorplan transformation. In *ICCV*, 2017. [4](#)
- [8] Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floornet: A unified framework for floorplan reconstruction from 3d scans. In *ECCV*, 2018. [4](#)
- [9] Yangzhou Liu, Yue Cao, Hao Li, Gen Luo, Zhe Chen, Weiyun Wang, Xiaobo Liang, Biqing Qi, Lijun Wu, Changyao Tian, Yanting Zhang, Yuqiang Li, Tong Lu, Yu Qiao, Jifeng Dai, and Wenhai Wang. Sequential diffusion language models. *CoRR*, abs/2509.24007, 2025. [2](#)
- [10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019. [2](#)
- [11] Yongsan Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. Spatiallm: Training large language models for structured indoor modeling. In *NeurIPS*, 2025. [4](#)
- [12] Srivathsan Murali, Pablo Speciale, Martin R. Oswald, and Marc Pollefeys. Indoor scan2bim: Building information models of house interiors. In *IROS*, 2017. [4](#)
- [13] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *CoRR*, abs/2502.09992, 2025. [2](#)
- [14] Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. [4](#)
- [15] Shikhar Tuli, Chi-Heng Lin, Yen-Chang Hsu, Niraj K. Jha, Yilin Shen, and Hongxia Jin. Dynamo: Accelerating language model inference with dynamic multi-token sampling. In *NAACL*, 2024. [2](#)
- [16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. [4](#)
- [17] Yuhao Wang, Heyang Liu, Ziyang Cheng, Ronghua Wu, Qunshan Gu, Yanfeng Wang, and Yu Wang. Vocalnet: Speech LLM with multi-token prediction for faster and high-quality generation. *CoRR*, abs/2504.04060, 2025. [2](#)
- [18] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. [4](#)
- [19] Yuanwen Yue, Theodora Kontogianni, Konrad Schindler, and Francis Engelmann. Connecting the dots: Floorplan reconstruction using two-level queries. In *CVPR*, 2023. [4](#)
- [20] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *ECCV*, 2020. [1](#), [3](#)