Floor Plan Segmentation with Boundary Aware Context Fusion Deep Learning Network

Uzair Sipra[1*]

[1*]Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, Canada.

Corresponding author(s). E-mail(s): sipra@ualberta.ca;

Abstract

Semantic segmentation of architectural floor plans is vital for automating building analysis, supporting applications such as 3D reconstruction and design automation. While Convolutional Neural Networks (CNNs) have significantly advanced this field, capturing long-range spatial dependencies inherent in complex layouts remains a challenge. This paper proposes an enhanced dual-branch encoder-decoder architecture, building upon the work of [1], to improve floor plan segmentation accuracy. We introduce key innovations: (1) a Global Context (GC) module integrated at the encoder bottleneck, employing cross-attention with learnable tokens to capture comprehensive spatial relationships; (2) an efficient variant of Multi-Head Self-Attention (MHSA) incorporating spatial reduction to ensure computational feasibility within decoder stages; and (3) Feature Fusion Attention (FFA) modules within the room decoder that leverage boundary decoder features, using the efficient MHSA to refine room predictions. We evaluate our proposed architecture against the baseline model by [1], on the R3D dataset using standard metrics, including Intersection over Union (IoU) and pixel accuracy. This work aims to demonstrate the effectiveness of targeted attention mechanisms and feature fusion strategies in enhancing the semantic understanding of architectural floor plans.

Keywords: Semantic Segmentation, Floor Plans, Neural Networks, Attention Mechanism

1 Introduction

Floor plans serve as essential visual communication tools within the architecture, engineering, and construction (AEC) industries, facilitating the effective design, planning, and coordination of building projects [1]. These schematic representations contain critical spatial and structural information, but extracting such information manually is a labor-intensive and time-consuming task. The complexity and increasing intricacy of modern architectural designs further increase the need for automated solutions capable of delivering consistent and scalable analysis.

Recent advances in deep learning, particularly the emergence of convolutional neural networks (CNNs), have demonstrated substantial promise in automating floor plan interpretation and segmentation tasks [2]. CNN-based models excel at learning local features from visual data but face challenges when modeling long-range dependencies due to their inherently limited receptive fields [3]. This limitation has prompted exploration into alternative architectures capable of capturing global contextual relationships. Transformers, originally proposed for sequence modeling tasks in natural language processing, have shown exceptional potential in computer vision applications through the introduction of self-attention mechanisms [4]. These mechanisms enable a model to dynamically assess the relevance of different input regions to one another, effectively modeling both local and global dependencies. Vision Transformer (ViT) and subsequent hybrid architectures have successfully adapted these principles to image-based tasks [3, 5].

This paper proposes an enhanced semantic segmentation model specifically tailored for architectural floor plans, extending the dual-branch encoder-decoder architecture introduced by [1]. Our primary contribution lies in the integration of advanced attention and feature fusion mechanisms designed to capture both granular local details and overarching global contextual information more effectively. Specifically, we introduce three key enhancements:

- A Global Context (GC) Module positioned at the encoder bottleneck [6]. This module utilizes an efficient cross-attention mechanism involving learnable global tokens to aggregate and distribute comprehensive spatial relationship information from the entire feature map.
- An Efficient Multi-Head Self-Attention (MHSA) mechanism [6]. This variant employs spatial reduction techniques before the attention calculation, making the computational cost of self-attention manageable for integration within the decoder stages, even at relatively higher feature map resolutions.
- A Feature Fusion Attention (FFA) Module at each decoding stage. These modules leverage feature representations from the dual decoders and applies the proposed Efficient MHSA to the fused features. This allows boundary cues, processed through an attention mechanism, to explicitly refine room segmentation predictions.

Our model preserves the multi-task learning strategy of the baseline, simultaneously predicting room boundaries and room types, exploiting the inherent synergistic relationship between these two tasks. We hypothesize that the explicit modeling of global context at the bottleneck, combined with computationally feasible self-attention and targeted feature fusion within the decoders, will yield improved segmentation accuracy compared to the baseline architecture, particularly for floor plans exhibiting complex layouts. We rigorously evaluate our proposed approach against the original work by [1] on the publicly available R3D dataset. This research contributes towards the development of more context-aware and accurate deep learning models for automated architectural drawing analysis.

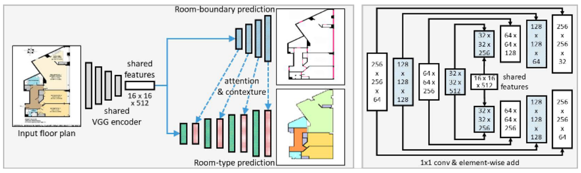


Fig. 1: Diagram of the baseline Architecture[1]

2 Background

The introduction of the self-attention mechanism by [4] marked a pivotal shift in the capabilities of deep learning models, particularly in modeling long-range dependencies within data. This breakthrough laid the foundation for the development of transformer-based architectures, which were subsequently adapted to the computer vision domain through notable contributions such as the Vision Transformer [3] and MaskFormer [5]. Inspired by these advancements, we hypothesized that incorporating a multi-head self-attention mechanism into the task of floor plan analysis—building upon the multi-task network proposed by [1]—could yield substantial performance improvements due to the mechanism's ability to model global contextual relationships.

The application of transformer-based models to floor plan segmentation offers several potential advantages. Unlike traditional convolutional neural networks (CNNs), which rely on local receptive fields, transformers can attend to relationships across the entire image. This is particularly beneficial for floor plans, where spatial relationships between architectural elements—such as the adjacency between a kitchen and a dining area—can span large distances and are critical to accurate segmentation. Furthermore, transformers inherently possess greater flexibility in modeling variations in visual style, making them well-suited for handling the wide diversity of floor plan representations, which often lack standardized notations.

By dynamically weighing the relevance of features across spatial locations, the self-attention mechanism facilitates adaptive learning of meaningful spatial dependencies, even in the presence of occlusions or visual noise. This global modeling capability has the potential to enhance segmentation performance in both structured and cluttered layouts, thereby improving robustness and generalization across similar datasets.

2.1 Deep Learning

Deep learning has emerged as a transformative field within machine learning, enabling significant breakthroughs across domains such as computer vision, natural language processing, and speech recognition. Its foundations lie in artificial neural networks, an idea dating back to the 1940s, but it was not until the late 2000s that deep learning began to achieve widespread success due to the confluence of increased computational power, large-scale labeled datasets, and algorithmic advancements [7–9].

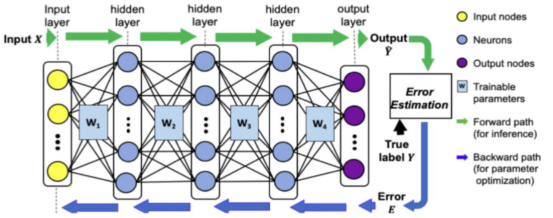


Fig. 2: Multilayer Perceptron Architecture

One of the earliest milestones was the development of multilayer perceptrons (MLPs), also known as feed-forward neural networks, which allowed the modeling of non-linear functions by stacking multiple layers of computational units. However, due to limitations in training algorithms and computational hardware, these networks were initially constrained in depth and effectiveness. The resurgence of interest in deep neural networks began with improvements in backpropagation and the introduction of more effective activation functions, such as the rectified linear unit (ReLU), which helped mitigate vanishing gradient issues [10].

A major turning point occurred in 2012, when a deep convolutional neural network (CNN) known as AlexNet dramatically outperformed traditional methods on the ImageNet Large Scale Visual Recognition Challenge [11]. This result signaled the start of a deep learning revolution, with CNNs becoming the dominant architecture for tasks involving spatially structured data, such as images and video.

In parallel, advancements were also made in sequence modeling. Recurrent neural networks (RNNs) and their variants—such as long short-term memory networks (LSTMs)—enabled learning from sequential and time-dependent data, laying the groundwork for breakthroughs in language modeling and machine translation [12].

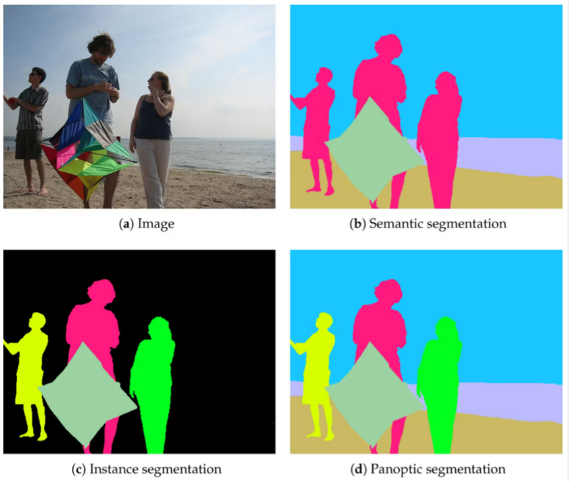
More recently, the introduction of the self-attention mechanism and the transformer architecture by [4] marked another paradigm shift. Transformers replaced recurrence with attention, enabling models to capture global dependencies more effectively and scale to massive datasets. This architecture has since been adopted across a wide range of applications, leading to the development of powerful models such as BERT [13], GPT [14], and Vision Transformer (ViT) [3].

The trajectory of deep learning continues to evolve rapidly, driven by community collaboration, open-source frameworks, and continual innovation in model architecture and training techniques. As a result, deep learning has transitioned from a niche academic interest to a cornerstone of modern artificial intelligence.

2.2 Semantic Segmentation

Semantic segmentation is a subset of image segmentation within the realm of computer vision, aimed at assigning a categorical label to every pixel in an image, thus facilitating a detailed understanding of the image contents. Unlike object detection or classification, semantic segmentation not only recognizes objects but also delineates their precise boundaries, which is essential in various applications such as autonomous driving, medical imaging, and architectural analysis [15].

Deep learning, particularly Convolutional Neural Networks (CNNs), has significantly advanced semantic segmentation. Architectures like Fully Convolutional Networks (FCNs), U-Net, and DeepLab have emerged as powerful tools due to their ability to learn hierarchical



(a) Image

(b) Semantic segmentation

(c) Instance segmentation

(d) Panoptic segmentation

Fig. 3: Categories of Image Segmentation [9]

representations from raw pixel data. FCNs transformed the approach to segmentation by replacing fully connected layers with convolutional layers, enabling pixel-level predictions and efficient computation [15]. Subsequently, architectures like U-Net introduced skip connections to preserve spatial information lost during downsampling, significantly improving segmentation accuracy for complex structures [2].

Despite these advancements, CNN-based approaches often struggle to effectively model long-range spatial dependencies due to their intrinsic local receptive fields. Recent developments in transformer architectures, notably the Vision Transformer (ViT) [3] and transformer-based segmentation frameworks such as MaskFormer [5], Mask2Former [5], and SegFormer [16], address these limitations by integrating self-attention mechanisms that capture global contextual information. These self-attention mechanisms allow the network to dynamically weigh the importance of different spatial locations, enhancing its capacity to understand intricate spatial relationships within the image.

2.3 Convolutional Neural Nets vs Transformers

The fundamental difference between CNNs and transformers lies in their approach to context capture and their inherent inductive biases. CNNs, with their convolutional layers, are inherently designed to process local features, gradually building a global understanding through hierarchical layers. This local focus, coupled with inductive biases like translation equivariance, often allows CNNs to perform well even with relatively smaller datasets. However, their ability to model long-range dependencies can be limited. In contrast, transformers, particularly ViTs, excel at capturing global context from the outset through their self-attention mechanisms. By treating an image as a sequence of patches and allowing each patch to attend to all other patches, transformers can effectively model relationships across the entire image.

However, this flexibility comes at the cost of requiring larger datasets to learn these relationships effectively, as transformers possess minimal inherent inductive biases about image structure. Furthermore, the self-attention mechanism can be computationally intensive, especially for high-resolution images, due to its quadratic complexity with respect to the number of input tokens (patches). In the context of floor plan segmentation, while CNNs have demonstrated strong performance, their limitations in capturing long-range dependencies might hinder their ability to accurately segment complex layouts where the relationships between distant rooms or architectural elements are crucial. Transformers, with their global context awareness, hold the potential to overcome these limitations and achieve improved accuracy on such complex floor plans. The performance of each architecture may also vary depending on the specific type of floor plan, with CNNs potentially being sufficient for simpler brochure-type plans, while transformers could offer a significant advantage for more intricate architectural drawings. The trade-off between these architectures often necessitates exploring hybrid approaches that combine the strengths of both.

2.4 Self-Attention Mechanism

Attention mechanisms empower neural networks to dynamically focus on the most relevant parts of the input data when generating an output. Self-attention, a cornerstone of the Transformer model [4], calculates attention scores between different elements within the same input sequence

Table 1: CNN vs Transformer

Aspect	CNN	Transformer
Context Capture	Primarily local, captures global context through deeper layers	Primarily global, excels at capturing long-range dependencies via self-attention
Data Requirements	Generally, performs well with relatively smaller datasets due to strong inductive biases	Typically requires large datasets for optimal performance due to minimal inductive bias
Computational Efficiency	Generally, more computationally efficient for image tasks due to localized operations	Can be computationally intensive, especially the self-attention mechanism
Performance on Complex Floor Plans	Strong performance but may struggle with intricate layouts requiring long-range understanding	Potential for improved accuracy due to global context awareness
Inductive Bias	Strong bias towards local features, translation equivariance, and locality	Minimal inductive bias, learns relationships directly from the data

(or feature map). This allows the model to explicitly capture long-range dependencies by assessing the relevance of every other element to a specific element.

The self-attention mechanism operates on an input sequence $X = [x_1, x_2, \ldots, x_n]$, where x_i represents the feature vector of the i-th token. The self-attention module computes the attention score for each pair of tokens, allowing the model to attend to relevant input parts.

Key Components:

- **Query (Q):** Represents the token for which we compute the attention scores.
- **Key (K):** Represents the tokens against which the attention scores are computed.
- **Value (V):** Represents the tokens contributing to the final output based on the attention scores.

Mathematical Formulation:

- **Linear Transformations:** The input feature vectors are linearly transformed into Query, Key, and Value vectors using learned weight matrices W_Q, W_K, and W_V:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V$$

- **Scaled Dot-Product Attention:** The attention scores are computed as the dot product of Query and Key vectors, scaled by the square root of the dimensionality d_k to maintain stable gradients:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- **Multi-Head Attention:** Instead of using a single attention mechanism, multiple heads are used to capture different aspects of the input. The outputs of multiple attention heads are concatenated and linearly transformed:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1,\ head_2,\ \ldots,\ head_h) \cdot W_O \tag{1}$$

where

$$head_i = \text{Attention}(QW_{Q_i},\ KW_{K_i},\ VW_{V_i}) \tag{2}$$

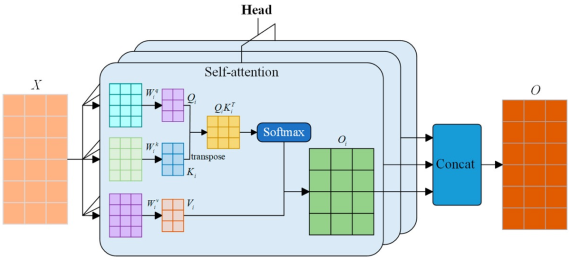


Fig. 4: This diagram illustrates the self-attention mechanism, where input X is projected into query (Q), key (K), and value (V) vectors using learnable weight matrices W^q, W^k, and W^v. The attention score is computed as the dot product of Q and K, normalized using a SoftMax function. These scores are used to weight the values (V) O_i. Outputs from multiple attention heads are concatenated to form the final output O [17].

2.5 Global Context

Standard self-attention has a computational complexity quadratic in the number of elements (e.g., pixels in a feature map), making it computationally expensive for high-resolution inputs common in segmentation tasks. Global Context (GC) Networks [8] proposed an efficient alternative for modeling global context. They combined non-local blocks (a form of self-attention) with channel attention mechanisms like Squeeze-and-Excitation. GC blocks aggregate context from the entire feature map into a compact representation, which is then used to modulate the original features, enhancing their contextual awareness without the quadratic complexity of full self-attention across spatial dimensions.

Capturing global context is particularly pertinent for floor plan analysis, where understanding the overall building layout and the relationships between potentially distant rooms is essential for accurate segmentation. Efficient variants of self-attention, often involving techniques like spatial reduction or localized attention windows, aim to provide similar benefits with lower computational overhead, making attention more viable within deeper network stages or on larger feature maps.

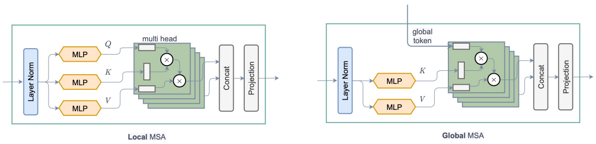


Fig. 5: Local and global attention blocks. Global attention block does not compute a query vector and reuses the global query computed via Global Token Generation [6].

3 Methodology

This section details the architecture of our proposed floor plan segmentation network.

3.1 Network Architecture Overview

Our proposed model architecture introduces significant enhancements through advanced feature processing mechanisms to effectively capture both local and global contextual information critical for accurate floor plan analysis. The model architecture consists of the following core components:

- **Shared Encoder:** A hierarchical convolutional encoder extracts multi-scale features, facilitating detailed structural representation [1].
- **Global Context (GC) Module:** An efficient self-attention-based module integrated at the bottlenecks to aggregate global spatial relationships via cross-attention [4, 8].
- **Dual Decoders (RBD and RTD):** Two parallel decoding paths built upon the encoded features, predicting room and boundary types.
- **Efficient Multi-Head Self-Attention (EMHSA):** A computationally optimized version of MHSA utilizing spatial reduction, an integral component within the FFA module [6].
- **Feature Fusion Attention (FFA) Module:** Utilizing the efficient MHSA, this module combines the contextual information from the dual decoders into a unified feature map.

The combination of the GC, EMHSA, and FFA modules form a novel component that we have denoted as **Room Boundary Aware Context Fusion Module (RBACFM)**.

3.2 Shared Encoder

The encoder utilizes a VGG-16 based backbone [18]. It consists of five stages, each comprising multiple convolutional layers followed by Rectified Linear Unit (ReLU) activations. Max-pooling layers are employed between stages to progressively down sample the spatial resolution of the feature maps (by factors of 2, 4, 8, 16, and 32) while increasing the feature dimensionality. This hierarchical structure allows the encoder to capture features at multiple scales, from local textures and edges to more abstract structural components. At the encoder's deepest stage—the bottleneck—the **Global Context (GC) Module** is introduced.

3.3 Global Context (GC) Module

Standard CNN operations are inherently local. Floor plan understanding, however, often necessitates reasoning about the global layout – for instance, identifying a room based on its position relative to distant entries or the overall building shape. Explicitly and efficiently capturing such long-range dependencies is critical for accurate

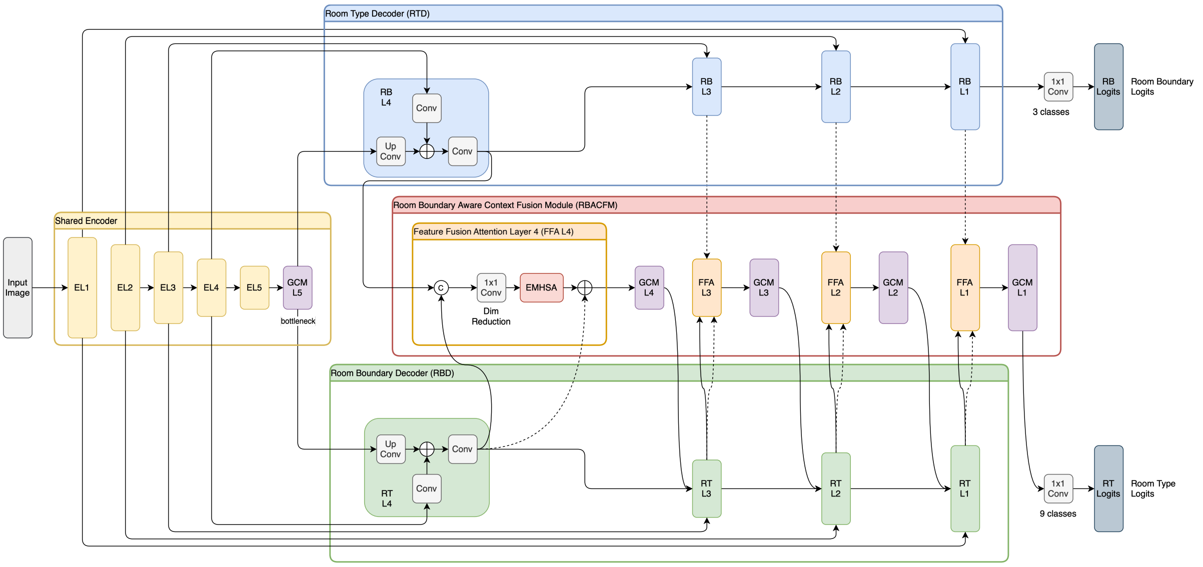


Fig. 6: This diagram depicts the complete proposed architecture for floor plan segmentation. A shared VGG-16 encoder extracts multi-scale features, which are globally enriched by a Global Context Module (GCM) at the bottleneck. The network then splits into two decoders: the Room Boundary Decoder (RBD, green) and the Room Type Decoder (RTD, blue), which predict structural boundaries and room classes, respectively. Feature Fusion Attention (FFA) modules within the Room Boundary Aware Context Fusion Module (RBACFM, red) integrate features from RBD into RTD using 1x1 convolutions and Efficient Multi-Head Self-Attention (EMHSA). Concatenation operations ("C") merge features from both decoders, and elementwise addition ($\oplus$) applies residual connections to preserve original RTD information. The final 1x1 convolution layers output logits representing unnormalized per-pixel scores for boundary (3 classes) and room type (9 classes) segmentation.

semantic interpretation, especially in complex layouts. Inspired by recent developments in vision transformers and global context networks [4, 6, 8], we introduce a two-step cross-attention GC module:

- **Context Aggregation:** A small, fixed set of learnable *global query tokens* are initialized. Cross-attention is performed where these global tokens act as queries, attending to all spatial locations in the bottleneck feature map (which provide the keys and values). This step effectively summarizes the global spatial context from the entire feature map into these compact, learnable tokens.
- **Context Distribution:** A second cross-attention step is performed. Here, the spatial features from the bottleneck map act as queries, attending back to the context-enriched global tokens (which now provide the keys and values). This broadcasts the aggregated global context information back to each spatial location, enriching the original features with global awareness.

This module enables the network to model long-range spatial interactions effectively, enhancing the feature representation for subsequent layers.

3.4 Decoder Branches (RBD and RTD)

Both decoder branches mirror the encoder's structure symmetrically. The feature maps are progressively upsampled using learnable transposed convolutions (up-convolutions) combined with features from corresponding encoder stages via skip connections. Each decoder stage typically involves:

- Upsampling the features from the previous, lower-resolution stage.

Table 2: Size of feature maps for each layer from Figure 6.

Stage	Layer	Output Shape
Input Image	N/A	[512, 512, 3]
Shared Encoder	EL1	[256, 256, 64]
	EL2	[128, 128, 128]
	EL3	[64, 64, 256]
	EL4	[32, 32, 512]
	EL5	[16, 16, 512]
	GCM L5	[16, 16, 512]
Room Boundary Decoder (RBD)	RB L4	[32, 32, 256]
	RB L3	[64, 64, 128]
	RB L2	[128, 128, 64]
	RB L1	[256, 256, 32]
	RB Logits	[512, 512, 3]
Room Type Decoder (RTD)	RT L4	[32, 32, 256]
	RT L3	[64, 64, 128]
	RT L2	[128, 128, 64]
	RT L1	[256, 256, 32]
	RT Logits	[512, 512, 9]
RBACFM	FFA L4 / GCM L4	[32, 32, 256]
	FFA L3 / GCM L3	[64, 64, 128]
	FFA L2 / GCM L2	[128, 128, 64]
	FFA L1 / GCM L1	[256, 256, 32]

- Concatenating or adding the upsampled features with features from the corresponding encoder stage (via skip connection).
- Applying convolutional layers to refine the combined features.

The RBD decoder outputs a segmentation map predicting boundary elements (walls, doors, windows, background), while the RTD decoder outputs a segmentation map predicting room types (bedroom, bathroom, kitchen, etc., background).

3.5 Efficient Multi-Head Self-Attention (EMHSA)

To combine the output feature maps of the dual decoders in a contextually relevant manner, we turned to a fundamental building block of transformer architecture, the Multi-Head Self-Attention mechanism [4]. This excels at modeling pairwise relationships between all feature locations, allowing the network to combine the most relevant feature information from the RBD and RTD. However, applying standard MHSA within decoder stages, especially at higher resolutions (e.g., 1/16th or 1/8th of input size), was often computationally prohibitive due to its quadratic complexity concerning the number of spatial locations (pixels). To mitigate this, an *efficient* variant of MHSA is employed [6]. Before computing the standard self-attention (projecting features to Query, Key, and Value matrices, calculating scaled dot-product attention, and combining results from multiple heads), a spatial reduction step is introduced. This involves applying average pooling with a specific stride to the input feature map, significantly reducing the number of tokens (spatial locations) involved in the attention calculation. The attention mechanism then operates on this reduced-resolution map. If necessary, the output is upsampled (e.g., via bilinear interpolation) back to the original feature map resolution before being combined with the input via a residual connection and normalized using layer normalization. This modification drastically reduces the

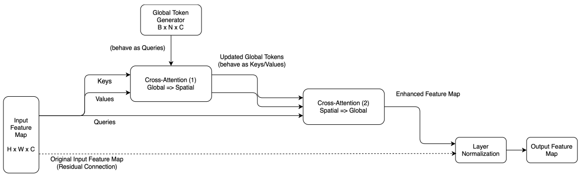


Fig. 7: This diagram illustrates the two-stage cross-attention mechanism used in the Global Context module. The input feature map is processed alongside a set of learnable global tokens generated per batch. In the first cross-attention stage, the global tokens serve as queries and attend to all spatial positions in the input feature map (which act as keys and values). This allows the tokens to aggregate information from the entire spatial layout. In the second cross-attention stage, the enriched global tokens are now used as keys and values, while the spatial features become the queries. This allows the aggregated global context to be redistributed back to each spatial location in the feature map. The output of this attention mechanism is added to the original input feature map via a residual connection to preserve low-level details, and the result is passed through layer normalization to stabilize training. The final output is a globally contextualized feature map ready for decoding.

computational and memory requirements of self-attention, making it feasible to incorporate within multiple decoder stages for feature refinement without incurring excessive overhead.

3.6 Feature Fusion Attention (FFA) Module

The tasks of room segmentation and boundary detection are highly correlated. Knowing the location of walls, doors, and windows provides strong cues for identifying room extents and types, and vice-versa. This module integrates features from the RBD into the RTD at each corresponding upsampling stage. The process involves:

- Concatenating the feature map from the current RTD stage with the feature map from the corresponding RBD stage.
- Applying a 1x1 convolution to the concatenated features to merge them and potentially adjust channel dimensions.
- Applying the **Efficient MHSA** module to the fused feature map. This allows the model to perform self-attention on features that combine both room and boundary information.
- Adding the output of the attention module back to the original RTD features via a residual connection.

This module explicitly encourages the model to leverage boundary information when making room predictions. By applying efficient self-attention *after* fusion, the model can learn complex relationships and dependencies between room types and their surrounding structural elements, leading to more contextually aware and accurate segmentation.

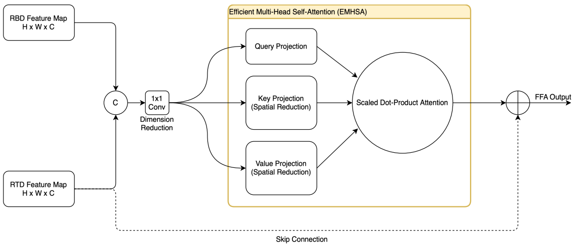


Fig. 8: This diagram illustrates the process by which the Feature Fusion Attention module integrates room type (RTD) and room boundary (RBD) feature maps at a given resolution level. The feature maps are first concatenated (denoted by “C”) and passed through a 1x1 convolution layer to reduce channel dimensionality. The resulting fused feature map is processed by the Efficient Multi-Head Self-Attention (EMHSA) block, which projects it into query, key, and value embeddings, applying spatial reduction for keys and values to reduce computational overhead. The scaled dot-product attention operation produces a context-aware representation. This refined output is then combined with the original fused features via element-wise addition (denoted by $\oplus$), forming a residual connection that helps preserve the original information while integrating enhanced context from the attention mechanism. The result is passed onward as the FFA output, improving segmentation performance through context-aware refinement.

3.7 Room Boundary Aware Context Fusion Module (RBACFM)

The Room Boundary Aware Context Fusion Module (RBACFM) highlights the effective synergy between its parts, leading to robust floor plan parsing. The Global Context (GC) module offers initial layout understanding, the Feature Fusion Attention module integrates boundary information at various resolutions within the Room-aware Top-Down (RTD) architecture, and the Efficient MHSA enables efficient spatial refinement within the combined features. The name RBACFM thus reflects the module's ability to fuse context in a way that is inherently "aware" of room boundaries due to this component collaboration. Reference Figure 6 for a visual diagram of the overall module.

4 Experimental Design

4.1 Dataset

We utilize the publicly available **R3D floor plan dataset** introduced by [1]. This dataset contains 232 rasterized floor plan images annotated with pixel-level labels for multiple architectural classes. These classes encompass both structural elements (e.g., Wall, Door/Window) relevant to the room boundary detection task and various room types (e.g., Living Room, Kitchen, Bedroom, Bathroom, Closet) relevant to the room type detection task. We strictly adhere to the standard train-test split defined by the dataset creators (typically 179 images for training and 53 for testing) to ensure fair and direct comparison with the baseline model and prior work using this benchmark.

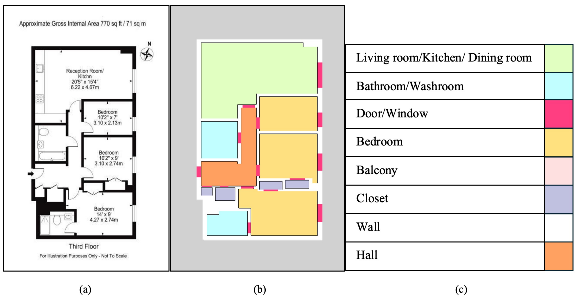

Approximate Gross Internal Area 770 sq ft / 71 sq m
Reception Room/ Kitchen
Bedroom
Bedroom
Bedroom
Third Floor
For Illustration Purposes Only - Not To Scale
Living room/Kitchen/ Dining room
Bathroom/Washroom
Door/Window
Bedroom
Balcony
Closet
Wall
Hall
(a)
(b)
(c)

Fig. 9: Floor plan layout from the R3D dataset. (a): Original, (b): Ground Truth, (c): Class Labels

4.2 Baseline Model

The primary benchmark for our comparison is the **original dual-branch floor plan segmentation architecture proposed by** [1]. This model serves as a strong baseline because it introduced the multi-task learning approach for this problem, established performance benchmarks on the R3D dataset, and forms the architectural foundation upon which our enhancements are built. Both our proposed model and the baseline implementation are trained using identical dataset splits, preprocessing steps, and evaluation metrics to isolate the impact of our proposed architectural modifications (GC module, Efficient MHSA, FFA).

4.3 Model Training

To guide the learning process for this two-headed architecture, a dual-purpose loss function was designed to accommodate the distinct goals of spatial region labeling and structural boundary delineation. The framework treats these learning objectives as complementary and integrates their supervision signals into a unified training routine.

- **Independent Decoder Supervision:** The network's two output branches—the room type prediction stream and the room boundary prediction stream—each generate their own error signals based on how well their outputs match ground truth annotations. For each task, the model calculates a pixel-wise comparison between predicted class probabilities and actual labels using a standard categorical divergence measure.
  - Probability Distribution Conversion: Each raw output map is passed through a SoftMax function to yield a probability distribution over the relevant classes at each pixel. This converts logits into interpretable confidence scores.

$$\text{SoftMax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{3}$$

$$N = num\ classes,\ \ i = single\ class$$

- Per-Pixel Error Measurement: Using the target segmentation masks (formatted as one-hot maps), the model computes the discrepancy between predicted distributions and true labels via categorical cross-entropy at every pixel location. These errors are then aggregated spatially.

$$L = -\sum_{k=1}^{K} y_k \log(\mathrm{SoftMax}(z)_k) \tag{4}$$

$$K : \textit{num classes},\ k : \textit{single class}$$
$$y_k : \textit{ground truth}$$

  where y_k is the ground truth label (1 if the pixel belongs to class k, 0 otherwise).

- Mean Loss per Task: The spatially aggregated losses are averaged over all pixels to yield a scalar loss value per decoder. This results in one loss for room type prediction ($L_{\text{room-type}}$) and another for boundary delineation ($L_{\text{room-boundary}}$).

- **Balancing Task Contributions:** Instead of assigning fixed importance to each task, the model dynamically scales each decoder's contribution to the total loss based on the volume of relevant labels in the batch. This strategy ensures that classes with more representation do not disproportionately influence the optimization.

The total loss is calculated as:

$$L_{\text{total}} = \alpha L_{\text{room-type}} + (1-\alpha)L_{\text{room-boundary}} \tag{5}$$

where α is a proportional weight constant equal to the ratio of labeled pixel count of one type over the combined pixel count:

$$\alpha = \frac{P_{\text{room-type}}}{P_{\text{room-type}} + P_{\text{room-boundary}}} \tag{6}$$

where P represents the count of labeled pixels for each task in the batch.

We implemented an **Early Stopping** training procedure which periodically (e.g., after each epoch), validates the model's performance and if the validation mIoU fails to improve by at least a minimum threshold for a specified number of consecutive validation cycles (patience period), training is terminated early. The model state corresponding to the best validation mIoU achieved is retained as the final trained model is evaluated on a separate validation dataset using the evaluation procedure described below. The primary metric monitored is the mean Intersection over Union (mIoU).

4.4 Hyper-Parameter Selection

Hyperparameter tuning is crucial for optimal model performance, generalization, and convergence during training. These parameters, set before training and not learned from data, are vital for preventing overfitting or underfitting, efficient use of computational resources, and improved model robustness. Given the vast hyperparameter space, we utilized an exhaustive grid search to determine the optimal values for the following hyperparameters.

- Learning Rate: The model was trained using the Adam optimizer with a fixed learning rate of 1×10^{-4}, which provided a stable gradient descent trajectory and prevented oscillations or vanishing updates during training.

- Epochs: We trained each model configuration for a maximum of 400 epochs, an intentionally high upper limit to ensure convergence across all variations. Our early stopping strategy proved effective, as most models converged near 250 epochs, at which point training automatically ceased, thus conserving both time and computational resources.

- Attention Heads: For configurations incorporating multi-head self-attention, each attention module was configured with 1, 2, 4, and 8 heads. This setting allowed the model to simultaneously learn multiple independent attention distributions, enriching its ability to capture spatial dependencies in different semantic contexts.

- Global Tokens: For the Global Context (GC) module, the number of learnable global tokens was varied across a range of values: 1, 2, 4, 8, 16, and 32 tokens. The number of tokens controls the capacity of the GC module to aggregate global context. Fewer tokens might lead to information bottleneck, while more tokens might increase computational cost and risk overfitting.

4.5 Evaluation Metrics

The following metrics were used to evaluate model performance.

- **Intersection over Union (IoU):** The primary metric used to evaluate class-specific segmentation quality is the Intersection over Union. For each semantic class i, IoU is calculated as:

$$\mathrm{IoU}_i = \frac{|P_i \cap G_i|}{|P_i \cup G_i|} = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i} \tag{7}$$

  where P_i is the set of pixels predicted as class i, G_i is the set of ground truth pixels labeled as class i, TP is True Positives, FP is False Positives, and FN is False Negatives. The numerator measures correctly predicted pixels (the intersection), while the denominator accounts for all pixels that are either predicted or labeled as class i (the union).

- **Mean Intersection over Union (mIoU):** To obtain a unified metric that reflects performance across all semantic classes, the average IoU is computed:

$$\mathrm{mIoU} = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathrm{IoU}_i \tag{8}$$

  where N_c is the total number of classes. This metric is sensitive to class imbalance and reflects both under- and over-segmentation tendencies.

- **Per-Class Pixel Accuracy:** This metric evaluates classification accuracy per class, defined as the ratio of correctly predicted pixels to the total number of ground truth pixels for each class:

$$\mathrm{Accuracy}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i} \tag{9}$$

  where TP_i represents true positives and FN_i denotes false negatives for class i. This allows for granular analysis of which room types or boundary types are more frequently misclassified.

- **Overall Pixel Accuracy:** To summarize general pixel-wise correctness across the entire dataset, the overall pixel accuracy is computed as:

$$\text{Overall Accuracy} = \frac{\sum_{i=1}^{N_c} \mathrm{TP}_i}{\sum_{i=1}^{N_c} (\mathrm{TP}_i + \mathrm{FN}_i)} \tag{10}$$

  This metric indicates the fraction of all pixels in the image that are correctly classified, irrespective of class.

All performance metrics are computed on the test split of the R3D dataset, using the best-performing epoch checkpoint per training run.

5 Results

This section presents the quantitative results of our proposed model configuration compared to the baseline [1] model. All results are reported on the R3D test set, following the evaluation procedure outlined in Section 4, with the optimal hyperparameter selection determined via grid search.

5.1 Quantitative Results

Table 3 summarizes the overall performance metrics. Our proposed model configuration demonstrates notable improvements in both mean Intersection over Union (mIoU) and Overall Pixel Accuracy compared to the baseline.

Table 3: Overall Performance Metrics

Model	**mIoU (%)**	**Overall Accuracy (%)**
Baseline	53.5	85.0
Proposed	**62.1**	**89.4**

Table 4 provides a detailed comparison of per-class Intersection over Union (IoU) scores (%). The proposed model shows improvement across all classes, with particularly substantial gains in several room categories.

Table 5 shows the per-class pixel accuracy. Similar trends of improvement are observed, complementing the IoU results.

Analysis of our results shows the overall mIoU increased by 8.6% (from 53.5% to 62.1%), and

Table 4: Per-Class IoU Comparison (mIoU is calculated as the mean of the IoU values for all listed classes)

Class	**Baseline (%)**	**Proposed (%)**
Closet	31.8	**42.0**
Bathroom	49.7	**56.5**
Living room, Kitchen, & Dining room	53.8	**68.8**
Bedroom	46.5	**62.5**
Hall	38.3	**48.7**
Balcony	20.1	**34.0**
Door & Window	56.3	**60.0**
Wall	90.5	**91.6**
mIoU	53.5	**62.1**

Table 5: Per-Class Pixel Accuracy Comparison

Class	**Baseline (%)**	**Proposed (%)**
Closet	45.8	**52.2**
Bathroom	68.1	**80.0**
Living room, Kitchen, & Dining room	72.2	**84.1**
Bedroom	61.6	**73.3**
Hall	54.3	**65.8**
Balcony	23.4	**44.5**
Door & Window	69.0	**69.8**
Wall	94.8	**95.9**
Overall Accuracy	85.0	**89.4**

the overall pixel accuracy improved by 4.4% (from 85.0% to 89.4%). The per-class IoU scores reveals improvements across all categories. The most significant gains were observed in room-type classes, including Bedroom (+16.0%), Livingroom/Kitchen/Dining room (+15.0%), Balcony (+13.9%), Hall (+10.4%), and Closet (+10.2%). Bathroom also saw a modest improvement (+6.8%). While structural elements (Wall, Door & Window) started from a higher baseline IoU, they also showed minor improvements (+1.1% and +3.7% respectively).

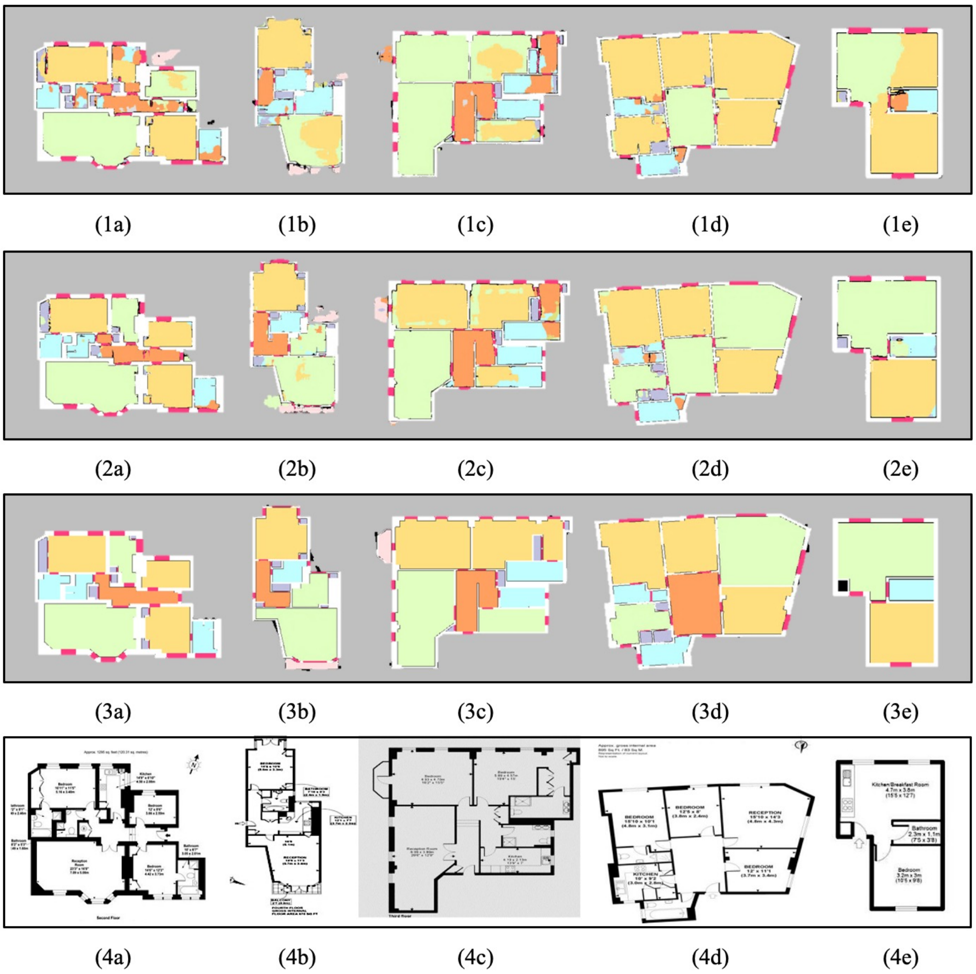
The per-class pixel accuracy results mirror the per-class IoU results. These results suggest that the architectural enhancement effectively improved the model's ability to understand and segment diverse architectural elements, especially complex room layouts. The following qualitative examples will further illustrate these improvements visually.

5.2 Qualitative Results

To visually assess and compare the performance between our proposed model and the baseline, we present segmentation results alongside the original input images and ground truth maps for four representative examples from the R3D test set in Figure 10.

A detailed comparison across the corresponding images within Figure 10 provides valuable insights into the specific improvements offered by our method across various floor plan layouts:

- column a:
  - *Comparison:* The baseline output (1a) exhibits considerable noise and misclassifications, particularly within the large top-left yellow room and the central orange hallway. In contrast, the proposed model (2a) provides much cleaner and more accurate segmentation for these areas, aligning better with the ground truth (3a). Smaller light-blue rooms are also better defined in (2a).
  - *Insight:* Demonstrates the proposed model's improved region consistency and handling of adjacent rooms.
- column b:
  - *Comparison:* The baseline (1b) struggles with the large bottom light-green room and misses the bottom balcony area (compare with 3b). The proposed model (2b) captures the overall structure more effectively, including the balcony, although some internal misclassifications persist within the large green room.
  - *Insight:* Highlights improvement in capturing smaller features (balcony) and overall structure, though internal consistency in large rooms remains a challenge.
- column c:
  - *Comparison:* Significant misclassifications are present in the baseline output (1c), especially in the large top-right yellow room.



(1a) (1b) (1c) (1d) (1e)

(2a) (2b) (2c) (2d) (2e)

(3a) (3b) (3c) (3d) (3e)

(4a) (4b) (4c) (4d) (4e)

Fig. 10: Qualitative comparison of segmentation results. Rows show Baseline predictions (1), Proposed model predictions (2), Ground Truth (3), and Original Inputs (4) for five examples from the test set (columns a-e). (Colors represent different room/boundary classes according to the R3D dataset legend, details can be found in Figure 9).

The proposed model (2c) yields a substantially cleaner result with more accurate room definitions and less internal noise, closely resembling the ground truth (3c).

  - *Insight:* Reinforces the proposed model's advantage in region homogeneity and accurate segmentation of different room types.

- column d:
  - *Comparison:* The baseline (1d) largely fails to identify the central light-green room correctly and shows fragmentation in the bottom-left areas. The proposed model (2d) successfully identifies the central room and provides cleaner segmentation of the smaller adjacent rooms/closets, significantly improving upon the baseline and matching the ground truth (3d) more closely
  - *Insight:* Clearly shows the proposed model's superior ability to differentiate adjacent rooms and handle areas with multiple small features.
- column e:

- *Comparison:* This simpler layout consists primarily of a large yellow room, a smaller light-green room, and a light-blue room. The baseline (1e) introduces some noise near the boundary between the yellow and light-green rooms. The proposed model (2e) provides a very clean segmentation, accurately delineating the three main rooms with minimal noise, closely matching the ground truth (3e).
- *Insight:* Even on simpler layouts, the proposed model demonstrates cleaner boundaries and better region definition compared to the baseline.

These qualitative findings support the quantitative results and demonstrate the practical benefits of the architectural enhancements (GC, EMHSA, FFA) in producing more accurate and reliable floor plan segmentations.

6 Discussion

The experimental results presented in Section 5, encompassing both quantitative metrics and the detailed qualitative analysis shown in Figure 10, provide compelling evidence for the effectiveness of the proposed enhanced architecture for floor plan semantic segmentation. The integration of a Global Context (GC) module, Efficient Multi-Head Self-Attention (EMHSA), and Feature Fusion Attention (FFA) modules leads to significant improvements over the baseline model [1]. The substantial quantitative gain, marked by an 8.6% increase in mIoU, is backed by the visual examples.

The GC module likely played a large role in improving the model's understanding of long-range spatial dependencies inherent in floor plan layouts. This is visually supported by the enhanced consistency observed within large rooms and the better handling of overall structures in the proposed model's outputs (Figure 10). For instance, the ability to correctly segment large contiguous rooms without the internal fragmentation seen in the baseline (1a vs 2a and 1c vs 2c) and the successful differentiation of distinct regions like the central green room in example 2d, where the baseline failed (1d), point towards the effectiveness of capturing global context early in the processing pipeline.

Furthermore, the Feature Fusion Attention (FFA) modules, utilizing EMHSA for computational feasibility within the decoder, appear successful in their goal of refining room predictions by leveraging boundary cues. The qualitative results show cleaner boundaries and better separation between adjacent rooms in the proposed model's outputs across multiple examples (e.g., comparing Row 1 vs Row 2 in Figure 10). The improved segmentation of complex shapes like hallways (visible in 2a, 2c) and the better definition of smaller adjacent features like closets (seen in 2d) or balconies (identified in 2b, unlike the baseline 1b) further support the hypothesis that attention-based fusion enhances local prediction accuracy and detail. The general reduction in noise near room edges compared to the baseline is also supportive of this refinement.

Despite these improvements, certain challenges persist. Achieving perfect segmentation for inherently ambiguous or very small classes remains difficult, as reflected in their lower IoU scores compared to larger rooms. The qualitative example (2b) demonstrates that even the proposed model can exhibit some internal noise within large, relatively uniform areas, and minor boundary inaccuracies can still occur. This suggests potential limitations in the feature representations or the fusion process that could be areas for future refinement. Moreover, while the combined system shows strong performance, precisely quantifying the individual contributions of the GC module versus the FFA/EMHSA components would require dedicated ablation studies.

7 Future Work

Based on the outcomes and limitations identified in this study, several promising directions for future research can be pursued:

- **Refining Attention and Fusion Mechanisms:**
  - Explore alternative implementations or configurations of the **GC module**, such as different cross-attention variants, dynamically learning the number of global tokens, or integrating GC information at multiple scales within the decoder.

  - Investigate different **Efficient MHSA** strategies (e.g., linear attention variants, different spatial reduction methods) or apply attention blocks more selectively within the decoders.

- **Exploring Advanced Network Backbones:** Replace the dated VGG encoder with a more powerful and modern CNN backbone (e.g., ResNet variants, ResNeXt, EfficientNet) or explore hybrid/pure transformer-based encoders (e.g., Swin Transformer [19], SegFormer [16], PVT [20]) while potentially retaining the enhanced dual-decoder structure.

- **Dataset Expansion and Robustness:** Evaluate the proposed architecture on larger, more diverse floor plan datasets (if available) or synthetic datasets to better assess its generalization capabilities across different architectural styles, complexities, and annotation conventions. Investigate domain adaptation or generalization techniques to improve robustness to unseen drawing styles.

- **Loss Function and Training Strategy Enhancements:** Explore more sophisticated loss functions designed for segmentation, such as Focal Loss or Dice Loss, especially for handling class imbalance. Investigate adaptive loss weighting schemes to dynamically balance the RTD and RBD task contributions during training. Explore boundary-aware loss functions that explicitly penalize errors near object edges.

- **Comprehensive Ablation Studies:** Conduct rigorous ablation studies to systematically isolate and quantify the individual contribution of each proposed component (GC module, the use of Efficient MHSA vs. standard MHSA or no attention, the FFA fusion strategy itself vs. simple concatenation) to the overall performance gain.

8 Conclusion

This paper presented an enhanced dual-branch encoder-decoder network architecture for semantic segmentation of architectural floor plans, extending the foundational work of [1]. We integrated several key enhancements: a Global Context (GC) module at core bottlenecks for long-range dependency modeling, computationally Efficient Multi-Head Self-Attention (EMHSA) utilizing spatial reduction, and Feature Fusion Attention (FFA) modules to combine the output feature maps of the RBD and RTD at each matching layer in a contextually aware manner via EMHSA.

Our experimental evaluation on the R3D dataset demonstrated that the full proposed architecture outperformed the baseline model. The enhanced network achieved a mean Intersection over Union (mIoU) of **62.1%**, a substantial increase from the baseline's **53.5%**. This improvement was consistent across all evaluated classes, indicating the robustness and effectiveness of the integrated enhancements. We attribute this success to the synergistic effect of the components within the RBACFM: the GC module improved the model's understanding of the overall floor plan layout, while the FFA modules with EMHSA allowed for more effective integration of boundary information into room predictions through attention-based feature fusion. This work underscores the value of combining mechanisms for both global context awareness and fine-grained, context-aware feature refinement in complex segmentation tasks.

While challenges remain, particularly regarding the precise contribution of individual components without specific ablation studies, the proposed architecture represents a clear advancement in automated floor plan analysis. Future research focusing on ablation studies, further refinement of these attention modules, and exploration of more advanced backbones holds continued promise for the field.

Acknowledgments. I would like to express my sincere gratitude to my supervisor, Dr. Mrinal Mandal, and his PhD student, Zhongguo Xu, for their guidance, encouragement, and continuous support throughout the duration of this work. Their expertise and insights were invaluable in shaping the direction and depth of this research.

Declarations

Funding: Not applicable

Conflict of interest/Competing interests: Not applicable

Ethics approval and consent to participate: Not applicable (This study did not involve human participants, human data, or human tissue. It utilized a publicly available dataset.)

Consent for publication: Not applicable

Data availability: The R3D dataset used in this study is publicly available as cited in [1].

Materials availability: Not applicable

Code availability: Not applicable

Author contribution: Uzair Sipra conceived the study, designed and implemented the model, conducted the experiments, analyzed the results, and wrote the manuscript under the supervision of Dr. Mrinal Mandal.

References

[1] Zeng, Z., Li, X., Yu, Y.K., Fu, C.-W.: Deep floor plan recognition using a multi-task network with room-boundary-guided attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 8698–8706 (2019). https://doi.org/10.1109/ICCV.2019.00879

[2] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 234–241 (2015). https://doi.org/10.1007/978-3-319-24574-4_28

[3] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR) (2021). https://arxiv.org/abs/2010.11929

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, , Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 5998–6008 (2017). https://papers.nips.cc/paper/7181-attention-is-all-you-need

[5] Cheng, B., Misra, I., Schwing, A., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1290–1299 (2022). https://arxiv.org/abs/2112.01527

[6] Hatamizadeh, A., Yin, H., Heinrich, G., Kautz, J., Molchanov, P.: Global context vision transformers. In: Proceedings of the 40th International Conference on Machine Learning (ICML), pp. 12633–12646 (2023). https://proceedings.mlr.press/v202/hatamizadeh23a.html

[7] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015) https://doi.org/10.1038/nature14539

[8] Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), pp. 1971–1980 (2019). https://openaccess.thecvf.com/content_ICCVW_2019/papers/NeurArch/Cao_GCNet_Non-Local_Networks_Meet_Squeeze-Excitation_Networks_and_Beyond_ICCVW_2019_paper.pdf

[9] Buhl, N.: Introduction to Semantic Segmentation. Accessed: 2025-05-03. https://encord.com/blog/guide-to-semantic-segmentation/

[10] Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 315–323 (2011). https://proceedings.mlr.press/v15/glorot11a.html

[11] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks.

In: Advances in Neural Information Processing Systems (NeurIPS), pp. 1097–1105 (2012). https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[12] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997) https://doi.org/10.1162/neco.1997.9.8.1735

[13] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pp. 4171–4186 (2019). https://aclanthology.org/N19-1423/

[14] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. Technical report, OpenAI (2018). https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[15] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440 (2015). https://doi.org/10.1109/CVPR.2015.7298965

[16] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. In: Advances in Neural Information Processing Systems (NeurIPS) (2021). https://arxiv.org/abs/2105.15203

[17] Xu, X., Li, Y., Ding, X.: Combined resnet attention multi-head net (cramnet): A novel approach to fault diagnosis of rolling bearings using acoustic radiation signals and advanced deep learning techniques. Applied Sciences **14**(4), 8431 (2024) https://doi.org/10.3390/app14048431

[18] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[19] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 10012–10022 (2021). https://doi.org/10.1109/ICCV48922.2021.00988

[20] Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 568–578 (2021). https://doi.org/10.1109/ICCV48922.2021.00062 . https://arxiv.org/abs/2102.12122