DI4DS Efficient Open-Vocabulary Models for Low-Power Computer Vision - LPCV Competition

Zainab Alhaddad, Hsiang-Yu Huang, Winni Tai

May 1, 2025

Abstract

This project aims to improve open-vocabulary grounding segmentation using the LPCVC 2025 dataset and model framework. We will start by reproducing the sample code provided and then enhance it using Focal-T models trained on COCO and evaluated on RefCOCOg. Our goal is to refine segmentation performance and reduce power usage by experimenting with different model improvements, such as fine-tuning strategies and data augmentation.

1 Introduction

Grounding segmentation is a crucial task in computer vision that connects language with image regions. It has many applications, such as image captioning, human-computer interaction, and autonomous systems. While existing models achieve strong results, improving open-vocabulary segmentation remains a challenge due to the variety of natural language inputs and diverse image content. This project will focus on building upon the provided baseline model from LPCVC 2025 and enhancing it with better training techniques and model refinements.

2 Related Work

Developing energy-efficient deep learning models for object detection is a growing focus, particularly for deployment on low-power devices. Research has explored methods to balance accuracy and computational efficiency, while advancements in open-vocabulary segmentation aim to bridge the gap between visual and linguistic representations.

Zou, Xueyan, et al. (2022) introduced **X-Decoder**, a unified model designed to seamlessly handle both pixel-level segmentation and language token prediction. X-decoder processes two types of query: (i) generic non-semantic queries and (ii) semantic queries derived from text inputs, which facilitates smooth integration across tasks, allowing them to enhance each other without relying on pseudo-labeling. By leveraging a rich visual-semantic representation space, X-Decoder generalizes effectively across diverse segmentation tasks, including open-vocabulary panoptic segmentation, referring segmentation, and dense prediction [1].

Yang, Jianwei, et al. (2022) introduced **Focal Modulation**, a novel mechanism designed to replace self-attention in Vision Transformers for dense prediction tasks. Unlike traditional self-attention that computes all pairwise interactions, Focal Modulation captures both short- and long-range dependencies through a multi-level pooling and modulation process, significantly reducing computational complexity [2].

Lin, Tsung-Yi, et al. (2014) introduce the **Microsoft COCO (Common Objects in Context)** dataset, aiming to advance object recognition within the broader scope of scene understanding [3].

Zhu, Jiachen, et al. (2025) propose **Dynamic Tanh (DyT)** as a simple, effective alternative to normalization layers in Transformers. DyT is an element-wise operation defined as $\mathrm{DyT(x)} = \tanh(\alpha \mathrm{x})$, where α is a learnable parameter [4].

Shazeer, Noam (2020) introduced **SwiGLU**, a gated activation variant that improves Transformer models by enhancing the expressiveness and efficiency of the feed-forward network layers [5].

Choromanski, Krzysztof, et al. (2021) introduced **Performer**, a novel Transformer variant that replaces the traditional softmax attention with an efficient **linear attention** mechanism based on positive random feature approximations [6].

Liu, Zihang, et al. (2021) proposed the **Gated Attention Unit (GAU)**, a lightweight and effective attention mechanism designed to improve the efficiency and expressiveness of Transformer models [7].

3 Methodology

The original baseline model is built using the Xdecoder framework. As its core, the model employs FocalNet as the vision backbone, which is responsible for extracting multi-scale image features. For the language component, the model uses a transformer-based language encoder (CLIPTokenizer) to process text inputs and generate text embeddings.

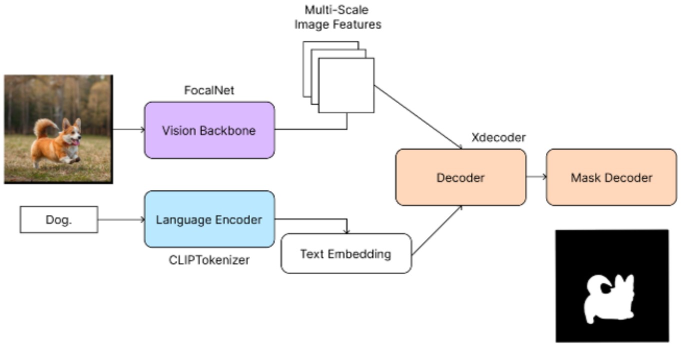

Figure 1: Baseline Model Structure

We plan to begin by replicating the baseline open-vocabulary grounding segmentation model provided in the LPCVC 2025 starter code. This model utilizes the X-Decoder framework with FocalNet as the vision backbone and CLIP as the language encoder. Once the baseline is running successfully, we will implement and evaluate a series of architectural improvements to enhance segmentation performance while optimizing for low-power efficiency.

Our planned enhancements include:

1. **Replace Normalization Layers with DyT**
2. **Integrate SwiGLU in FFN**
3. **Combine DyT and SwiGLU**
4. **Introduce DyT to Attention Layers**
5. **Replace Self-Attention with Linear Attention + Gated Attention Units (GAU)**
6. **Insert DyT Inside GAU**

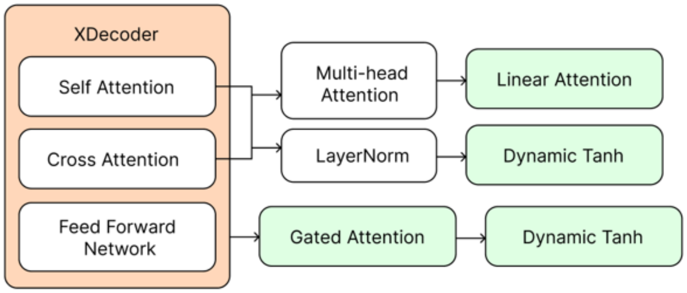

Figure 2: X-Decoder Optimization method

Models Used

X-Decoder: (our main model)

- Generalized decoding model that seamlessly predicts both pixel-level segmentations and language tokens.
- Unified architecture supports a wide range of tasks.
- Pretrained on a combination of segmentation data and millions of image-text pairs.

Focal (backbone):

- Vision Transformer variant designed to efficiently capture both local and global visual dependencies.
- Introduces focal self-attention mechanism.

CLIP (text encoder):

- Process text input using Transformer-based text encoder
- Input the tokenized vector of the input text to the model

4 Datasets

We will use the COCO dataset for training and RefCOCOg for evaluation. COCO provides a large set of annotated images, which is ideal for training vision-language models.

Key Features of the COCO Dataset:

- **Extensive Image Collection:** Over 330,000 images.
- **Diverse Object Categories:** 80 object categories.
- **Rich Annotations:** Bounding boxes, segmentation masks, keypoints, and captions.
- **Contextual Scene Representation:** Emphasizes objects in their natural contexts.

5 Evaluation

We will compare our improved model against the baseline results provided in the LPCVC 2025 sample code using the following metrics:

1. **mIoU (Mean Intersection over Union):**

$$mIoU = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{A_f a_i}{A_j a_j} \right|$$

2. **cIoU (Complete Intersection over Union):**

$$cIoU = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + av$$

3. **Precision@Threshold:**

$$Precision = \frac{True\,Positives}{True\,Positives + False\,Positives}$$

4. **Inference Speed (FPS - Frames Per Second):**

$$FPS = \frac{N}{T}$$

5. **Qualcomm AI Hub latency time (second / image-text):** Compare the model performance through latency time.

6 Timeline

- **Week 1-2:** Reproduce the baseline model
- **Week 3-4:** Identify key areas for improvement

- **Week 5-6:** Implement data augmentation
- **Week 7-8:** Optimize hyperparameters
- **Week 9-10:** Perform final evaluations
- **Week 11:** Prepare final report

7 Evaluation Results

In our final evaluations, we tested multiple model enhancements on top of the LPCVC 2025 baseline to assess their impact across key metrics: **GPU Power Usage**, **Dice loss for mask prediction**, **Inference Time on Edge Device**, **cIoU**, **mIoU**, and **Precision** at different IoU thresholds (0.5 to 0.9), shown in Table 1 and Table 2, and Figure 3 to Figure 4.

7.1 Baseline Performance

The original model achieved a cIoU of **20.294** and mIoU of **17.366**, with an inference time of **256.1 ms** and GPU power usage of **744,577.6 units**. This served as the starting point for comparing all enhancements.

7.2 Model Enhancement Analysis

- **DyT Only:** Integrating Dynamic Tanh (DyT) into the Feed-Forward Networks reduced GPU power by **12.2%** and improved efficiency (inference time dropped to **188.7 ms**). However, it slightly decreased cIoU and mIoU compared to the baseline, showing that while DyT improved energy efficiency, it did not significantly enhance segmentation accuracy on its own.

- **SwiGLU Only:** Replacing standard MLPs with SwiGLU activations led to a **10.6% reduction** in GPU power, though with slight decreases in mIoU and precision values. However, **mIoU** and **precision** values slightly dropped compared to baseline, indicating again an efficiency gain but not a major accuracy gain.

- **SwiGLU+DyT:** Combining SwiGLU and DyT increased GPU power usage by **15.3%**yet showed slight improvements in cIoU (**20.736**). compared to only using DyT or swiGLU. However, the computational cost remains high in the last half of the training section, shown in Figure 4, making it less ideal for a low-power setting.

- **Adding DyT in Attention Layer:** Adding DyT directly into attention blocks achieved the **highest cIoU (25.191)** and **mIoU (23.346)**, among all tested methods. Precision at all thresholds (0.5–0.9) also significantly improved. However, GPU power usage increased by **36%** over the baseline, and inference speed became slower. This made the method very accurate, but at the cost of higher energy consumption — not ideal for our low-power goal.

- **Linear Attention + Gated Attention:** Replacing standard attention with Linear Attention and replacing FFN layer to Gated Attention Unit (GAU) achieved a good balance. Although there was a high peak during the training section, shown in Figure 4, the overall GPU Power still dropped by **8.3%**. Inference time stayed efficient (**190.5 ms**). cIoU improved to **21.92** and mIoU to **17.837**, precision also slightly improved over the baseline at most IoU thresholds. This approach provided both higher accuracy and lower power usage without any significant tradeoffs.

- **DyT in Gated Attention:** Introducing DyT inside the Gated Attention module also boosted performance strongly, reaching a cIoU of **24.235** and mIoU of **22.742**, with much better precision at all levels. GPU Power decreased **7.5% reduction**, which is an improvement to our goals. This method matched the best accuracy and with much better energy efficiency than our baseline models.

Method	GPU Power	Inference Time	cIoU	mIoU
Baseline	744577.6	256.1 ms	20.294	17.366
DyT	653833.8 (-12.2%)	188.7 ms	19.816	16.218
SwiGLU	665617.6 (-10.6%)	189.2 ms	20.426	16.262
SwiGLU+DyT	859082.7 (+15.3%)	190.4 ms	20.736	16.608
DyT to Attention	1012993.5 (+36%)	190.9 ms	25.191	23.346
Linear + GAU	682878.5 (-8.3%)	190.5 ms	21.92	17.837
DyT in GAU	688777.8 (-7.5%)	191.0 ms	24.235	22.742

Table 1: Evaluation Table for GPU usage and accuracy

Method	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9
Baseline	14.536	9.755	6.452	2.759	0.622
DyT	13.214	9.211	5.596	2.293	0.428
SwiGLU	12.709	8.434	5.208	2.604	0.349
SwiGLU+DyT	13.486	8.9	5.13	2.41	0.466
DyT to Attention	21.492	15.7	9.988	5.829	1.71
Linear + GAU	14.691	10.027	5.985	2.759	0.699
DyT in GAU	21.337	16.09	11.27	6.568	1.477

Table 2: Evaluation Table for Precision

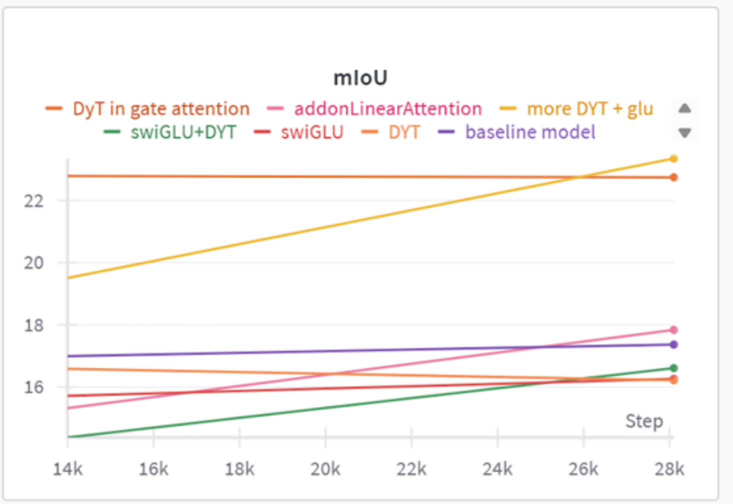

Figure 3: mIoU comparison

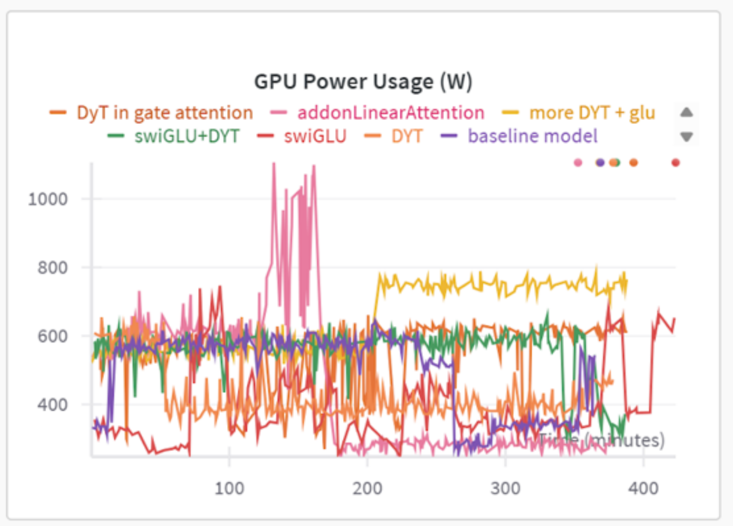

Figure 4: Real-time power consumption of GPU in watts

8 Conclusion

Since **balancing accuracy and low-power efficiency** is the goal, our final model, which combines replacing LayerNorm with DyT, replacing Multi-head Attention to linear attention and replacing FFN layer to Gated Attention while also replacing the layerNorm in Gated Attention to DyT, is the best choice. It gives a huge boost in segmentation accuracy (**24.235** cIoU, **22.742** mIoU) while lowering GPU power usage compared to the baseline (**7.5%** lower), with fast inference times under 1 second. Thus, our overall model achieved the LPCV project's goal the best, offering a strong improvement in accuracy while maintaining energy efficiency and speed.

References

[1] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. *arXiv preprint arXiv:2212.11270*, 2022.

[2] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal modulation networks. *arXiv preprint arXiv:2203.11926*, 2022.

[3] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. *arXiv preprint arXiv:1405.0312*, 2014.

[4] Jiachen Zhu et al. Transformers without normalization. *arXiv preprint arXiv:2503.10622*, 2025.

[5] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[6] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2021.

[7] Zihang Liu, Xuezhi Wang, Mostafa Dehghani, Hanxiao Liu, Yi Tay, et al. Gated attention units: Efficient transformers for language and vision. *arXiv preprint arXiv:2202.10447*, 2022.