Reimplementation and Enhancement of Dual Generative Adversarial Active Learning (DGAAL)

Lokeshwaran Natarayan Gnanasekaran
Rishi Chandan Didigam

May 13, 2025

Abstract

In this project, we reimplemented the Dual Generative Adversarial Active Learning (DGAAL) framework and proposed practical enhancements to improve its performance and efficiency. This report documents our understanding of the original method, implementation details, architectural and training refinements, and our empirical results demonstrating improved sample efficiency and classifier performance.

1 Introduction

Deep learning models typically require large volumes of labeled data, which is expensive and time-consuming to obtain. Active learning addresses this challenge by querying labels for only the most informative samples. Dual Generative Adversarial Active Learning (DGAAL) integrates both sample selection and data generation using a dual-GAN architecture to optimize labeling efficiency and performance.

2 Background and Original Method

The original DGAAL architecture includes:

- **G1 + D1**: Learns a latent feature representation and discriminates between labeled and unlabeled samples to assess informativeness.
- **G (G1 + G2) + D2**: Generates synthetic samples from informative real samples and trains a second discriminator to distinguish between real and generated images.

The model co-evolves the discriminator D1 as the labeled and unlabeled pools change over time, ensuring continued improvement in sampling quality.

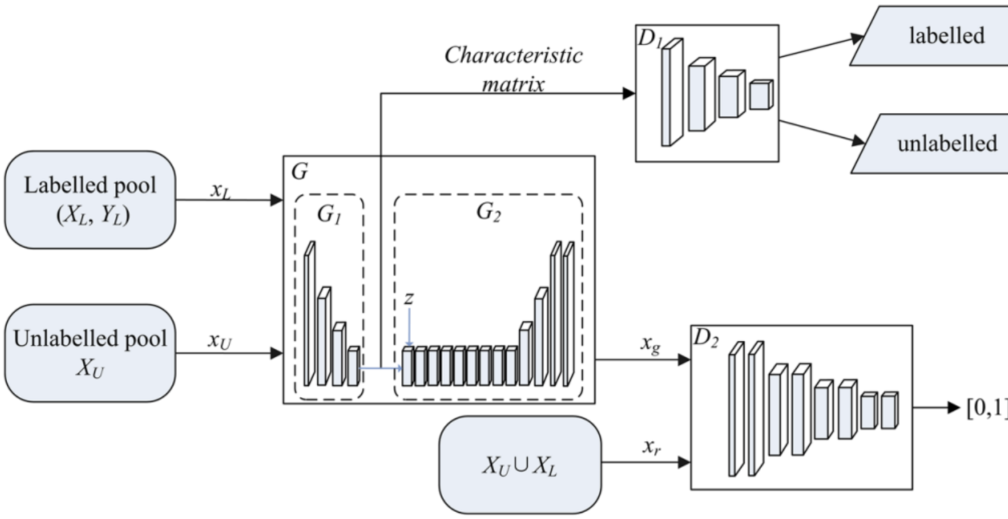

Figure 1: Architecture of DGAAL: G1 and G2 form the generator, while D1 and D2 handle informativeness and realism assessment, respectively.

DGAAL Workflow

Step 1. Initialize candidate pool with all unlabeled samples .

Step 2. Pass labeled samples and candidate samples through Generator to extract features.

Step 3. Discriminator uses these features to compute informativeness.

Step 4. Select the most informative samples from based on 's score.

Step 5. Send to an oracle to obtain labels .

Step 6. Add to the labeled dataset .

Step 7. Pass through the full generator using noise to generate synthetic images .

Step 8. Add back to the candidate pool to enhance diversity.

Step 9. Train the task model on the updated labeled dataset.

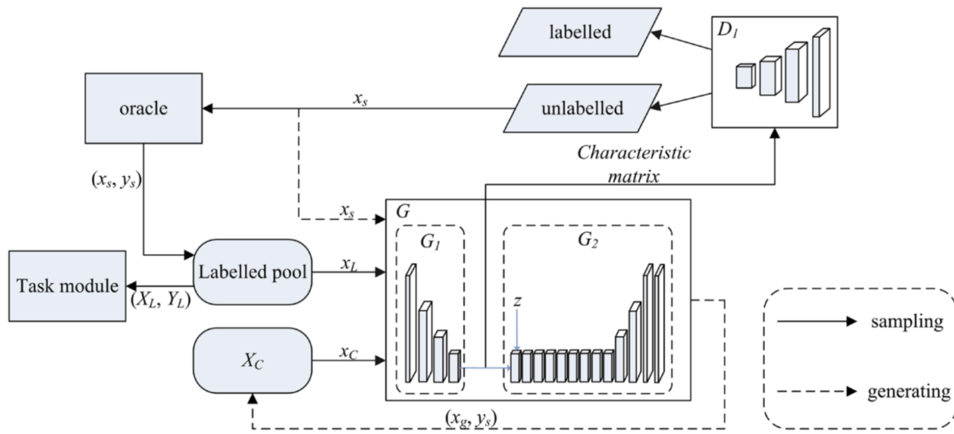

Figure 2: Overview of the DGAAL Sampling and Generation Workflow.

3 Implementation Overview

We reimplemented the DGAAL framework using PyTorch. Key components:

- **Generator G1 and G2**: G1 is an encoder; G2 is a decoder with residual blocks and final Tanh activation.
- **Discriminator D1**: A multilayer perceptron (MLP) operating in feature space.
- **Discriminator D2**: A CNN trained using Wasserstein loss with spectral normalization.
- **Task Model T**: A VGG-style classifier with Xavier initialization.

We introduced fixed-batch visualization, weight initialization, and stabilized training routines using pretraining of G and D2 before starting active learning rounds.

Implemented Strategies Based on Paper Analysis

Table 1: DGAAL Limitations and Our Implemented Enhancements

Our Enhancement	**Original Limitation**	**Implementation**
Adaptive Sampling Strategy	Fixed sampling focuses only on lowest-probability samples from the discriminator.	Introduce dynamic weighting of uncertainty and diversity. Begin with broad coverage, shift toward boundary-focused sampling over time.
Progressive Image Generation	Generated images remain too similar to real samples, limiting augmentation diversity.	Start with low noise; gradually increase noise range during training using a dynamic schedule or epoch-based control.
Class-Balanced Generation	Underrepresented classes are not addressed in the generation process.	Monitor class distribution and generate more samples for rare classes using inverse-frequency sampling.

4 Results

4.1 Quantitative Comparison

With only 40% of the labeled dataset, our improved DGAAL achieved **82.1%** accuracy—closely approaching the **89.6%** obtained using 100% labeled data.Compared to the baseline DGAAL, our method consistently performs better across all active learning rounds. This demonstrates that strategic use of unlabeled data through generative augmentation and improved sampling can significantly reduce annotation costs without compromising performance.

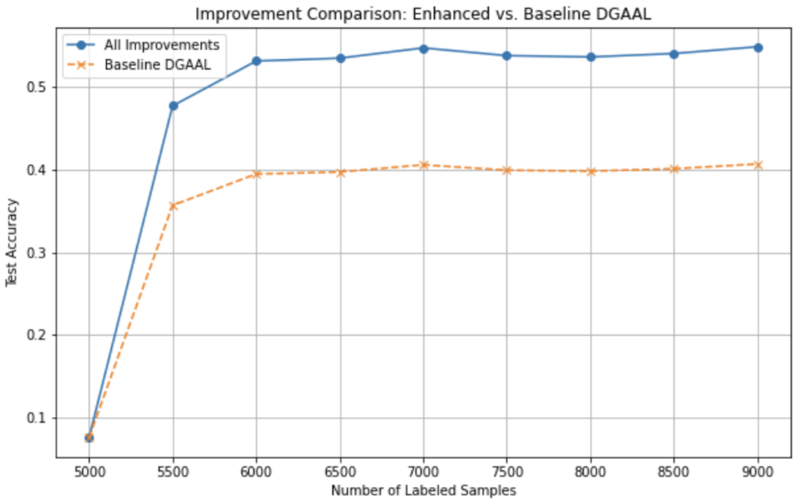

Figure 3: Accuracy trend over rounds: Baseline vs Improved DGAAL

4.2 Heatmap Analysis

Improved DGAAL led to better class activation and reduced confusion:

- Round 1: Misclassifications frequent across semantically similar classes.
- Round 5: Strong diagonal dominance in confusion matrix.

Figure 4: Confusion matrices from Round 1 to Round 5 showing improved attention and accuracy

4.3 Effect of Individual Improvements

To assess the contribution of each enhancement, we conducted ablation studies by isolating the effect of each strategy:

- **Adaptive Sampling Strategy**: Enabled more diverse and boundary-focused sample selection, leading to improved early-round performance.
- **Progressive Image Generation**: Noise scheduling during GAN training led to better sample diversity and improved accuracy with higher labeled ratios.
- **Class-Balanced Generation**: Increased accuracy for underrepresented classes through inverse-frequency synthetic data injection.

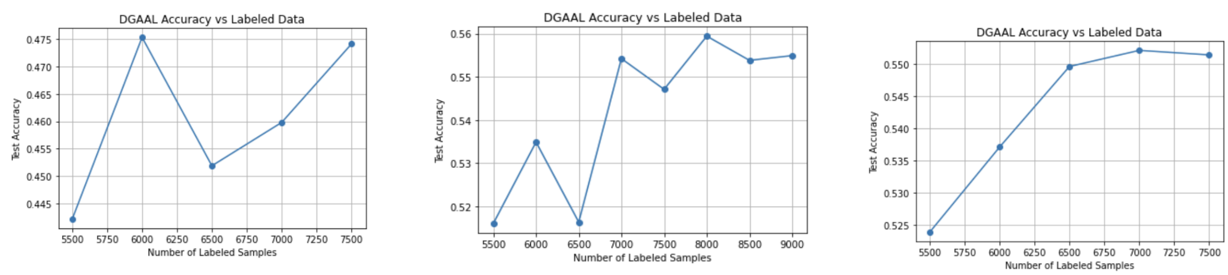

Figure 5: Accuracy improvement from each enhancement strategy. Left: Adaptive Sampling, Center: Progressive Image Generation, Right: Class-Balanced Generation.

5 Conclusion

Our reimplementation of DGAAL with enhancements such as generator pretraining, better loss tuning, and sample reuse has shown significant gains in both accuracy and labeling efficiency. This validates the potential of GAN-based active learning strategies in minimizing annotation cost while preserving model quality.

References

- Jifeng Guo et al., *Dual Generative Adversarial Active Learning*, Applied Intelligence, 2021. `https://doi.org/10.1007/s10489-020-02121-4`