
Grad-CAM Supervision for Semantically Aligned GAN Image Generation

Anushka Bhawe

Carnegie Mellon University
Pittsburgh, PA 15213
abhawe@andrew.cmu.edu

Anning Li

Carnegie Mellon University
Pittsburgh, PA 15213
anningl@andrew.cmu.edu

Urmi Dedhia

Carnegie Mellon University
Pittsburgh, PA 15213
udedhia@andrew.cmu.edu

Andrew Zhang

Carnegie Mellon University
Pittsburgh, PA 15213
az4@andrew.cmu.edu

Abstract

Generative Adversarial Networks (GANs) can produce highly realistic images, but ensuring that these images capture meaningful, class-specific features remains a challenge. Standard adversarial training may generate visually convincing outputs without guaranteeing that they activate the same discriminative regions as real images. To address this, we integrate Grad-CAM into GAN training for CIFAR-10 classification problems. We first train a CNN classifier on real images and use Grad-CAM to generate heatmaps highlighting important class features. These heatmaps are then compared to those from GAN-generated images, and a Grad-CAM loss is introduced to minimize their differences alongside the standard adversarial loss. We evaluate our method using the Generator Loss, Discriminator Loss, and the final Classification Accuracy. By aligning synthetic images with real feature activations, our approach improves both the realism and interpretability of GAN outputs, making them more useful for downstream classification tasks.

1 Overview and Context

1.1 Motivation

Deep learning models achieve high performance when trained on large, diverse datasets, but data collection and annotation are expensive and time-consuming. GANs offer a solution by generating synthetic images to augment datasets, yet a key challenge remains:

"Do these generated images capture the essential, class-specific features needed for reliable classification?"

Traditional evaluation metrics, such as Frechet Inception Distance (FID) and Inception Score (IS), focus on image realism but lack interpretability—they do not verify whether a classifier relies on the same meaningful features in real and synthetic images.

This gap in interpretability limits the usefulness of GAN-generated data for critical applications like medical imaging, autonomous driving, and low-resource learning environments, where feature fidelity is essential. By incorporating Grad-CAM to compare activation heatmaps of real and synthetic images, we introduce a method that not only enhances the visual quality of generated images but also ensures they align with the discriminative features that classifiers rely on. This approach benefits researchers

and practitioners seeking more reliable, transparent, and effective synthetic data generation for deep learning applications.

1.2 Objective

This project aims to:

1. Integrate Grad-CAM-based supervision into the generator’s learning process. Develop a loss function using Grad-CAM heatmaps to minimize differences between real and synthetic images, guiding the generator towards producing images with class-consistent and semantically meaningful features.
2. Qualitatively investigate how the generator learns class-specific features relative to real data, using heatmap visualizations. This provides interpretability into GAN decision-making, highlighting areas where the synthetic images align with or deviate from real image activations.
3. Quantitatively evaluate the method’s effectiveness by measuring classification accuracy improvements across multiple classes. We aim to demonstrate that emphasizing Grad-CAM loss not only improves structural alignment, quantified by SSIM, but also significantly enhances downstream classification performance.

2 Related Work and Background

2.1 Generative Adversarial Networks

Generative Adversarial Networks (GANs) [6] were first introduced in 2014, proposing a minimax game between a generator (G) and a discriminator (D). The generator attempts to produce realistic images that fool the discriminator, while the discriminator aims to correctly distinguish real images from those generated. This adversarial training framework has since inspired numerous extensions and improvements.

One significant leap was the Deep Convolutional GAN (DCGAN) [21] proposed by Radford, which introduced convolutional architectures into GAN frameworks, enabling more stable training and improved image quality. Building on these foundational ideas, Salimans [23] presented a set of practical training enhancements—such as feature matching and mini-batch discrimination—to address common issues like mode collapse. These early works laid the groundwork for applying GANs to diverse image generation tasks, including benchmarks like CIFAR-10.

Recent research has explored using GANs for data augmentation, particularly in generating synthetic datasets to enhance model performance. DA-GAN [17] and ADA-GAN [28] optimize GAN-based augmentation strategies to improve data efficiency in low-resource settings. Similarly, GANs have been applied in medical imaging [5] and autonomous driving datasets [29], generating high-quality synthetic samples to mitigate data scarcity issues. Architectural advancements such as StyleGAN [13], Self-Attention GAN (SAGAN) [31], and BigGAN [2] have significantly improved image fidelity and diversity.

However, despite their success, challenges in training stability, mode diversity, and interpretability persist, motivating further research into enhancing their reliability and usability across diverse applications. A primary concern is mode collapse, shown as Fig. 1 where the generator produces a limited variety of outputs, failing to capture the full diversity of the real data distribution. Several works have explicitly targeted this challenge. Unrolled GANs [18] propose “unrolling” the optimization steps of the discriminator within the generator’s objective to stabilize training and reduce mode collapse. Mode Regularized GANs [4] add regularization terms to the generator’s objective, encouraging it to capture more modes in the data distribution. Further, Lucic [15] performed a large-scale study comparing different GAN variants, revealing a wide variance in performance and underscoring that many GANs do not fully capture the underlying data manifold. Their findings reinforce the notion that different training strategies, architectures, and hyperparameters can lead to vastly different outcomes in terms of mode diversity and image quality.

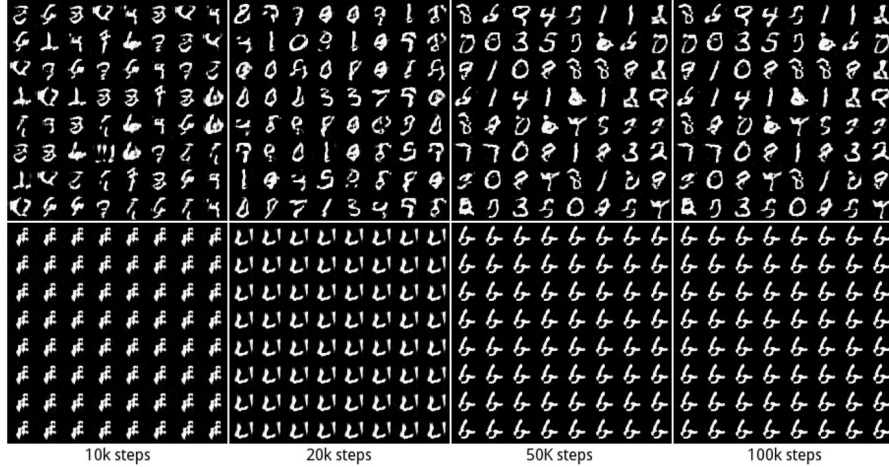


Figure 1: Real-life data distributions are multimodal. For example, in MNIST, there are 10 major modes from digit ‘0’ to digit ‘9’. The samples below are generated by two different GANs. The top row produces all 10 mode while the second row creates a single mode only (the digit “6”). This problem is called mode collapse when only a few modes of data are generated.

2.2 Integrating Interpretability: Grad-CAM

As machine learning models grow more sophisticated, understanding their decision-making processes has become increasingly important. Grad-CAM[25] is a widely used interpretability technique that visualizes which regions of an image contribute most to a convolutional neural network’s (CNN) classification decision. By leveraging gradient information from the final convolutional layers, Grad-CAM produces class-discriminative heatmaps, highlighting the most influential regions in an image. Grad-CAM++[3] extends this method by refining the weighting of pixel contributions, leading to higher-resolution and more detailed attention maps.

While interpretability techniques like Grad-CAM were initially developed to analyze real-image classifiers, they have also been applied to generative models. In the context of GANs, Grad-CAM can be used to compare the feature activations of real and synthetic images. If a generated image is intended to represent a specific class (e.g., “airplane” in CIFAR-10), its Grad-CAM heatmap can be compared to that of a real airplane image to assess whether the generator is capturing the correct class-specific features in the right spatial regions.

Beyond Grad-CAM, several other interpretability methods have been explored for deep learning models. Layer-wise Relevance Propagation (LRP)[1] assigns importance scores to each pixel by backpropagating relevance from the model’s output layer, offering fine-grained explanations of model predictions. SHAP (SHapley Additive exPlanations)[16] quantifies feature importance by attributing predictive influence using concepts from cooperative game theory, making it useful for both CNNs and other black-box models. Attention Maps in Transformer-based architectures[26] provide an alternative interpretability mechanism by visualizing self-attention distributions, which can also be leveraged in generative models like GANs or diffusion models.

By integrating these interpretability techniques into GAN training, researchers can better understand how synthetic images encode class-specific features, diagnose potential failures in generation, and refine training objectives to improve realism and consistency.

3 Methodology

3.1 Motivation

Generative Adversarial Networks (GANs) demonstrate significant potential for synthetic data augmentation. However, traditional GAN training focuses primarily on image realism without explicitly enforcing semantic alignment. Our method addresses this gap by integrating Grad-CAM visual

attention supervision into the GAN training pipeline. We hypothesize that aligning synthetic images with real images at the semantic feature level, guided by Grad-CAM heatmaps from pretrained classifiers, will produce more semantically consistent and class-discriminative synthetic data.

3.2 Model Description

3.2.1 Overview

Our methodology consists of three distinct stages aimed at enhancing the classification accuracy by generating semantically aligned synthetic images.

Step 1: Classifier Pipeline (Figure 2) shows how we establish a baseline classification accuracy by training a ResNet18 model on the CIFAR-10 dataset. Each image is classified, and Guided Grad-CAM generates heatmaps to visualize areas critical to the classification decisions. These heatmaps are grouped into class-specific buckets to represent the classifier’s attention patterns clearly.

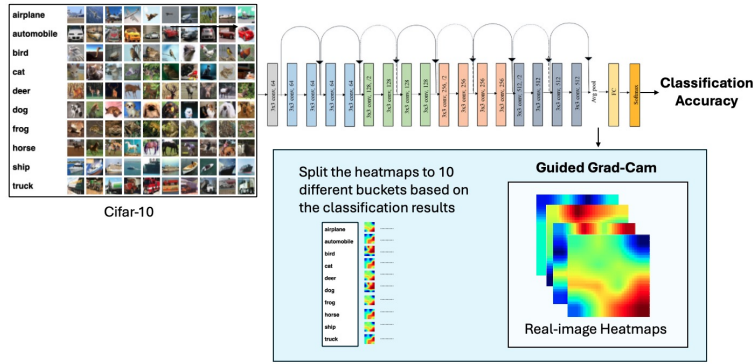


Figure 2: Model Step 1 - Classifier Pipeline

Step 2: GAN Pipeline (Figure 3) incorporates Grad-CAM supervision directly into GAN training. The generator network is optimized using a combined loss function comprising the standard binary cross entropy and a novel Grad-CAM emphasis loss. For each synthetic image generated, its heatmap is computed and compared with five randomly selected real-image heatmaps from the same predicted class. Similarity scores (SSIM and Pearson correlation) form the Grad-CAM emphasis loss component guiding the generator to create images with features aligned to classifier focus areas.

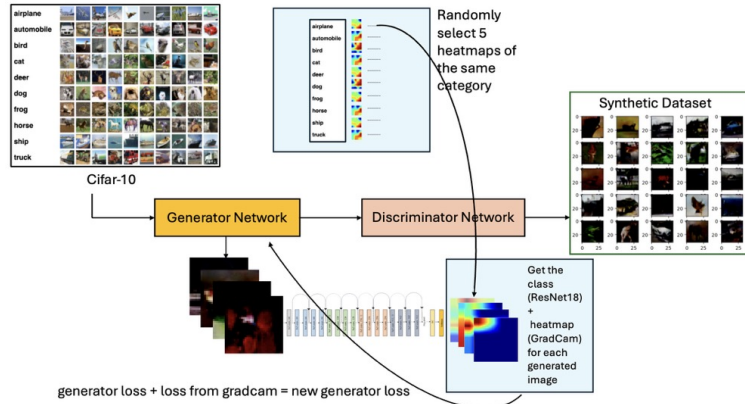


Figure 3: Model Step 2 - GAN Pipeline

Step 3: Final Classifier Pipeline (Figure 4) evaluates the impact of generated synthetic data on classification performance. The synthetic images generated by the GAN are combined with the original CIFAR-10 training dataset. This augmented dataset is then used to retrain the classification

model, assessing improvements in accuracy due to the addition of Grad-CAM supervised synthetic images.

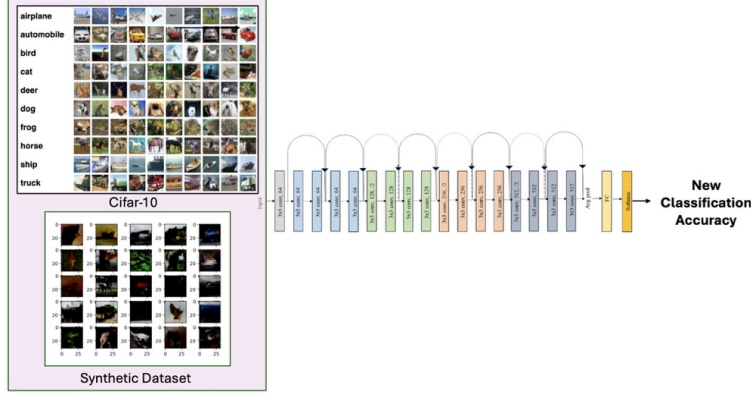


Figure 4: Model Step 3 - Final Classifier Pipeline

3.2.2 Classifier Setup

To generate meaningful visual attention maps, we employ distinct CNN architectures for classifier backbones:

1. **MobileNetV2** (Pretrained on CIFAR-10 via TorchHub)
2. **ResNet18** (Trained from scratch on CIFAR-10)

The Grad-CAM heatmaps are computed from the last convolutional layer of each classifier using the `pytorch-grad-cam` library, providing spatial attention maps highlighting class-discriminative features.

3.2.3 GAN Model Architecture

We adopt a Deep Convolutional GAN (DCGAN) architecture due to its stable training properties and high image-quality generation capabilities on small datasets.

Generator The generator takes a random noise vector $z \sim \mathcal{N}(0, I)$ of dimension $100 \times 1 \times 1$ and progressively upsamples it to generate a 32×32 RGB image:

Table 1: Generator architecture.

Layer	Configuration
Input	Noise vector $z \in \mathbb{R}^{100 \times 1 \times 1}$
ConvTranspose2d	$100 \rightarrow 512$, kernel size 4, stride 1
BatchNorm2d + ReLU	Standard normalization and non-linearity
ConvTranspose2d	$512 \rightarrow 256$, kernel size 4, stride 2
BatchNorm2d + ReLU	Standard normalization and non-linearity
ConvTranspose2d	$256 \rightarrow 128$, kernel size 4, stride 2
BatchNorm2d + ReLU	Standard normalization and non-linearity
ConvTranspose2d	$128 \rightarrow 3$, kernel size 4, stride 2
Tanh Activation	Output normalized image $3 \times 32 \times 32$

Discriminator The discriminator classifies real vs. synthetic images via convolutional downsampling:

Table 2: Discriminator architecture.

Layer	Configuration
Input	RGB image $3 \times 32 \times 32$
Conv2d	$3 \rightarrow 64$, kernel size 4, stride 2
LeakyReLU	Non-linearity (negative slope = 0.2)
Conv2d	$64 \rightarrow 128$, kernel size 4, stride 2
BatchNorm2d + LeakyReLU	Normalization and non-linearity
Conv2d	$128 \rightarrow 256$, kernel size 4, stride 2
BatchNorm2d + LeakyReLU	Normalization and non-linearity
Conv2d	$256 \rightarrow 1$, kernel size 4, stride 1
Output	Scalar logit value

3.2.4 Precomputed Real Grad-CAM Heatmaps

Grad-CAM heatmaps for a representative subset of CIFAR-10 training images are precomputed and cached per class, significantly reducing computation during GAN training. For each fake image, heatmaps from the predicted class are retrieved from this cache to guide training.

3.3 Dataset

The CIFAR-10 dataset[14] is a widely used benchmark dataset for image classification tasks. It consists of 60,000 color images divided into 10 different classes, with 50,000 images used for training and 10,000 for testing. Each image has a resolution of 32×32 pixels and belongs to one of the following categories:

Table 3: CIFAR-10 class labels.

Airplane	Automobile	Bird	Cat	Deer
Dog	Frog	Horse	Ship	Truck



Figure 5: Sample images from CIFAR-10 dataset

The data set is balanced, which means that each class contains an equal number of images. The images are stored in three RGB channels and the data set is available in both Python and binary formats for ease of use. Data is normalized to the range $[-1, 1]$, using the standard transform:

$$\text{transforms.Normalize}((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))$$

3.4 Evaluation Metric

Classification accuracy is a direct measure of the classifier’s performance on both real and synthetic datasets. It quantifies the percentage of correctly classified samples out of the total number of samples:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}} \times 100\% \quad (1)$$

We compare the classification accuracy of models trained solely on real CIFAR-10 data against those trained with the addition of Grad-CAM-guided GAN-generated synthetic images. Improved

accuracy on synthetic images indicates that the GAN generates semantically meaningful and class-discriminative samples.

3.5 Loss Function

Our proposed Grad-CAM guided GAN employs a combination of three loss functions to enhance both the visual realism and the semantic relevance of the generated images. The total generator loss is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{GAN}} + \lambda \mathcal{L}_{\text{GC}} \quad (2)$$

where λ is a hyperparameter controlling the contribution of Grad-CAM component.

3.5.1 Adversarial Loss ($\mathcal{L}_{\text{GAN}}$)

We adopt the standard adversarial loss from Generative Adversarial Networks (GANs) [7]. The discriminator D attempts to distinguish between real images x and generated images $G(z)$, while the generator G tries to fool the discriminator into classifying its outputs as real. We use the Binary Cross Entropy with Logits Loss (BCEWithLogitsLoss), which combines a sigmoid layer and the binary cross entropy loss in one function. The loss for the discriminator and generator is defined as:

$$\mathcal{L}_D = -\mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] - \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))] \quad (3)$$

$$\mathcal{L}_G = -\mathbb{E}_{z \sim p_z} [\log D(G(z))] \quad (4)$$

This adversarial loss encourages the generator to produce images that the discriminator classifies as real.

3.5.2 Grad-CAM Region Emphasis Loss ($\mathcal{L}_{\text{GC}}$)

We propose a custom Grad-CAM region emphasis loss to ensure that the generated images activate similar discriminative regions as real images when passed through a classifier. Grad-CAM [25] produces class-specific heatmaps that highlight important regions for decision-making. The region emphasis loss will be the average of SSIM (Structural Similarity Index Measure) score and Pearson's Correlation. We use these to quantify the similarity between the Grad-CAM heatmaps of real and fake images. SSIM captures spatial structural similarities, ensuring that both heatmaps highlight similar regions in an image, making it useful for understanding feature alignment. Pearson Correlation, on the other hand, measures linear similarity between the pixel intensities of the two heatmaps, indicating whether the activations follow a consistent pattern. Together, these metrics help evaluate whether the GAN-generated images activate similar regions in the classifier as real images, ensuring meaningful feature learning.

SSIM Loss[30] The SSIM loss evaluates the similarity between the generated image $G(z)$ and the real image x by measuring luminance, contrast, and structure similarities. The SSIM-based loss is defined as:

$$\mathcal{L}_{\text{SSIM}} = 1 - \text{SSIM}(x, G(z)), \quad (5)$$

where $\text{SSIM}(\cdot, \cdot)$ computes the structural similarity between two images.

Pearson Loss[20] The Pearson loss measures the linear correlation between the pixel values of the real and generated images. It is computed as:

$$\mathcal{L}_{\text{Pearson}} = 1 - \frac{\sum (x - \bar{x})(G(z) - \overline{G(z)})}{\sqrt{\sum (x - \bar{x})^2} \sqrt{\sum (G(z) - \overline{G(z)})^2}} \quad (6)$$

This loss encourages a high correlation between corresponding pixels in real and generated images.

Final Grad-CAM Region Loss We take the average of both losses to obtain the final content consistency loss:

$$\mathcal{L}_{\text{GC}} = \frac{1}{2} (\mathcal{L}_{\text{SSIM}} + \mathcal{L}_{\text{Pearson}}) \quad (7)$$

This loss ensures that the generated images not only capture the structural integrity of the real images but also maintain strong statistical relationships with them.

3.6 Detailed Training Pipeline

Each training batch follows these explicit steps:

1. Generate synthetic images $G(z)$ from sampled random noise z .
2. Pass generated images through a fixed classifier:
 - Predict class $\hat{y}$.
 - Compute Grad-CAM heatmaps (H_{fake}) for the predicted class.
3. Randomly select 5 real Grad-CAM heatmaps ($H_{real}^{(1..5)}$) from the cache corresponding to the predicted class $\hat{y}$.
4. Compute Grad-CAM region alignment loss (L_{GC}) using SSIM and Pearson correlation between H_{fake} and each H_{real} :

$$\mathcal{L}_{GC} = 1 - \frac{1}{2} [\text{SSIM}(H_{fake}, H_{real}) + \text{Pearson}(H_{fake}, H_{real})]$$

averaged over the 5 sampled heatmaps.

5. Backpropagate combined losses through the generator:

$$\mathcal{L}_{total} = \mathcal{L}_{GAN} + \lambda_{GC} \cdot \mathcal{L}_{GC}$$

6. Train discriminator using standard binary cross-entropy loss:

$$\mathcal{L}_D = -\mathbb{E}_{x \sim p_{data}} [\log D(x)] - \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))]$$

4 Baseline and Extensions

4.1 Baseline for GANs with CIFAR-10 dataset

We establish baselines for two popular GAN architectures: DCGAN (Deep Convolutional GAN[22]) and CGAN (Conditional GAN[19]) on the CIFAR-10 dataset.

DCGANs are composed entirely of convolutional layers, deliberately excluding max-pooling and fully connected layers. They rely on strided convolutions for downsampling and transposed convolutions for upsampling. This design helps capture spatial hierarchies in the data and improves training stability. On the other hand, CGANs extend the basic GAN framework by incorporating label information into both the generator and the discriminator. By conditioning the generation process on class labels, CGANs enable more controlled image synthesis and often achieve better class-specific generation performance compared to standard GANs.

Table 4 summarizes the baseline performance of DCGAN and CGAN on the CIFAR-10 dataset. We report the generator loss (L_G) and discriminator loss (L_D) at different training epochs (50, 100, and 150 epochs). These results serve as benchmarks for evaluating the impact of our proposed Grad-CAM-based supervision in subsequent experiments.

Table 4: Baseline performance of GAN models on CIFAR-10.

Model (GAN)	Epochs	Generator Loss (L_G)	Discriminator Loss (L_D)
DCGAN	50	1.970	0.619
DCGAN	100	1.664	0.444
DCGAN	150	1.858	0.549
CGAN	50	2.109	0.421
CGAN	100	2.369	0.391
CGAN	150	2.809	0.318



Figure 6: Generated images at different training epochs.

4.2 Baseline for Classification

We use the standard CIFAR-10 classification accuracy as our baseline metric, evaluating multiple convolutional neural network (CNN) architectures including VGG16, ResNet18, ResNet20, ResNet50, MobileNet, and DenseNet121. The baseline training is conducted on an RTX 3080 Ti GPU with 50 epochs, a batch size of 128 for training and 100 for testing, and a stochastic gradient descent (SGD) optimizer configured with an initial learning rate of 0.1, momentum of 0.9, and weight decay of 5×10^{-4} . A cosine annealing learning rate scheduler with $T_{max} = 200$ is applied to adjust the learning rate over time. We measure the total training time and the final classification accuracy on the test set as key evaluation metrics. To assess the impact of our proposed method, we extend the baseline by introducing additional training images generated by a Grad-CAM-guided GAN. These augmented images are designed to emphasize the most salient regions identified by Grad-CAM, thereby guiding the GAN to produce more informative and discriminative samples. We compare the performance of each model with and without the GAN-augmented data, demonstrating improvements in test accuracy and convergence behavior due to the enhanced data diversity and guidance provided by Grad-CAM.

4.3 GradCAM Baseline and Extensions

To evaluate the effectiveness of integrating GradCAM into GAN training, we used a pretrained CNN classifier (ResNet18) on the CIFAR-10 dataset and used GradCAM to generate heatmaps highlighting important class features for real images. Then we used a pretrained DCGAN to generate synthetic CIFAR-10 images. We generated 10 pairs of real and synthetic images for different CIFAR-10 classes, ensuring the synthetic image was classified as the same class as the real image. For each pair, we computed GradCAM heatmaps and calculated the Grad-CAM Region Emphasis Loss ($\mathcal{L}_{GC}$)

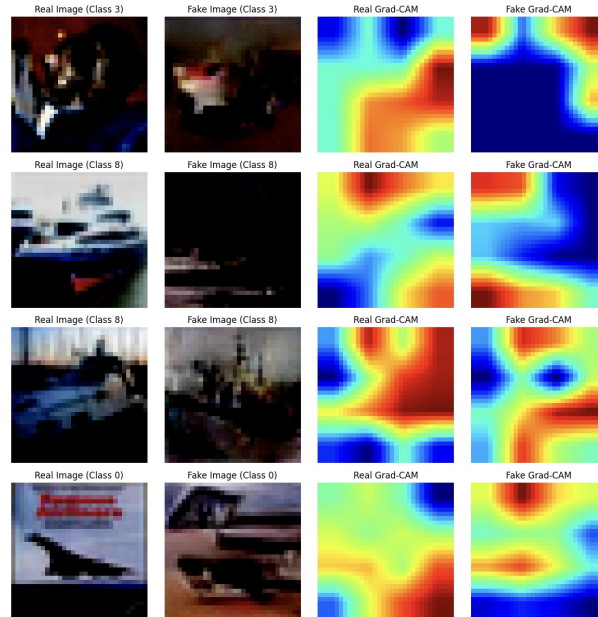


Figure 7: Initial sample gradcam result

The results of our GradCAM experimentation reveal interesting insights into the alignment of feature activations between real and synthetic images, shown as in Fig.7 and Table.5:

Table 5: Structural similarity (SSIM) and Pearson correlation values across different image pairs.

Pair	SSIM	Pearson
1	-0.0424	-0.2850
2	0.2369	-0.0365
3	0.3426	0.4089
4	0.1242	-0.3347
5	-0.0152	-0.0582
6	-0.1441	-0.0753
7	0.5440	0.8797
8	0.0838	-0.0135
9	-0.0072	0.2066
10	0.3124	0.4764

These relatively low similarity scores indicate that there is still a significant gap between the features activated by real and synthetic images. However, some key observations can be made:

- **Variability across pairs:** The similarity scores varied widely between different image pairs, with some showing strong alignment (e.g. Pair 7 with SSIM 0.5440 and Pearson 0.8797) while others showed very poor alignment or even negative correlation.
- **Class-dependent performance:** Certain classes appeared to have better alignment than others, suggesting the GAN may be more effective at capturing discriminative features for some categories.
- **Structural vs. intensity alignment:** The generally higher SSIM scores compared to Pearson correlations suggest that the synthetic images may be activating roughly the right regions, but with different intensity patterns.
- **Potential for improvement:** The fact that some pairs showed strong alignment indicates that the GradCAM-guided approach has promise, but may require further refinement to consistently produce well-aligned synthetic images.

Table 6: Performance comparison without GradCAM-guided GAN augmented images.

Network	GPU	Epochs	Training Time (s)	Accuracy (%)
VGG16[27]	RTX 3080 Ti	50	558.07	80.030
ResNet18[8]	RTX 3080 Ti	50	703.42	85.180
ResNet50[9]	RTX 3080 Ti	50	2283.56	84.170
PreActResNet50[10]	RTX 3080 Ti	50	2243.01	83.660
MobileNet[11]	RTX 3080 Ti	50	561.23	77.410
MobileNetV2[24]	RTX 3080 Ti	50	951.98	81.330
DenseNet121[12]	RTX 3080 Ti	50	2670.44	86.570

5 Results and Analysis

Our experiments evaluate the impact of the hyperparameter λ_{GC} controlling the importance of Grad-CAM alignment loss, across different classifiers:

- **Experiments 1–3 (MobileNetV2):** $\lambda_{GC} \in \{1.0, 0.7, 0.5\}$
- **Experiment 4 (ResNet18):** $\lambda_{GC} = 1.0$

5.1 Training and Validation Loss

5.1.1 MobileNetV2 GradCAM Loss Analysis

For the MobileNetV2-based GAN experiments, Figure 8 highlights two key observations:

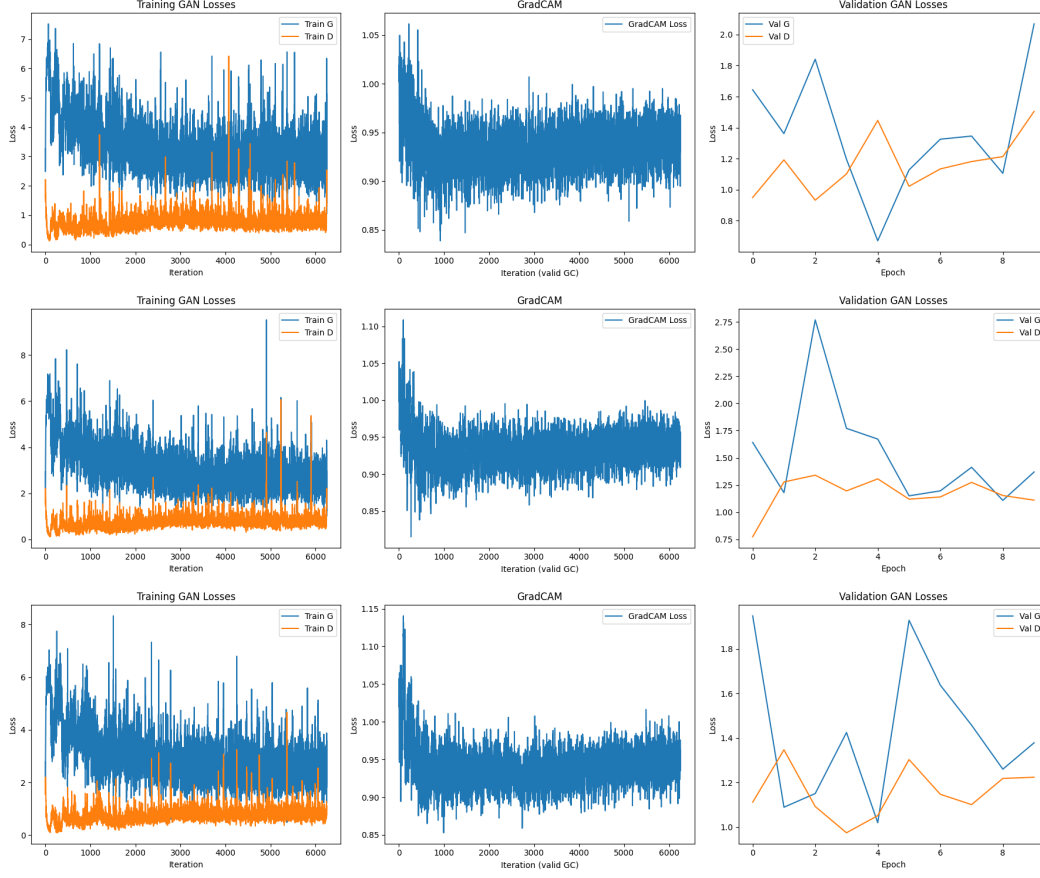


Figure 8: Training and Validation Losses for MobileNetV2 with GradCAM λ values of 1.0 (top), 0.7 (middle), and 0.5 (bottom).

1. The GradCAM loss decreases significantly and most efficiently within the first approximately 1500 iterations (around 2 epochs), across all three λ settings. Beyond this point, the loss plateaus, suggesting that the primary benefit of GradCAM supervision occurs early during training.
2. There is no notable difference in the trends of training and validation losses for both the generator and discriminator across the three choices of λ (1.0, 0.7, and 0.5). This indicates that altering the strength of the GradCAM supervision within these tested ranges does not significantly affect the general training dynamics.

5.1.2 ResNet18 GradCAM Loss Analysis

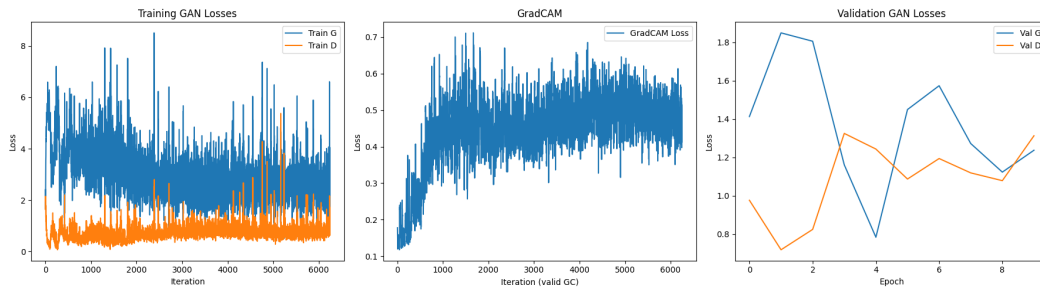


Figure 9: Training and Validation Losses for ResNet18 with GradCAM λ value of 1.0.

In contrast, the ResNet18-based GAN experiment (Figure 9) exhibits different characteristics:

1. The GradCAM loss notably increases over the first 6000 iterations. This behavior suggests that, unlike the MobileNetV2 scenario, GradCAM supervision with ResNet18 does not contribute positively within the initial training phase. It also indicates the potential need for further training epochs or model tuning for convergence.
2. Compared to the MobileNetV2 results, ResNet18’s GradCAM supervision shows less stable dynamics and a reversed beneficial effect in early iterations. This contrast emphasizes model-specific interactions with GradCAM loss and highlights the need for tailored hyperparameter tuning and possibly extended training periods for optimal performance.

5.2 Heatmap Analysis

We performed an analysis of the heatmap alignment performance across varying GradCAM emphasis strengths (λ) for MobileNetV2-based GAN training after 10 full epochs (around 6000 iterations). Specifically, we measured two critical metrics, Structural Similarity Index Measure (SSIM) and Pearson correlation, to quantify the similarity between generated and real image heatmaps. This analysis helps in understanding the effectiveness of emphasizing GradCAM loss in guiding the GAN towards producing images aligned with classifier attention regions.

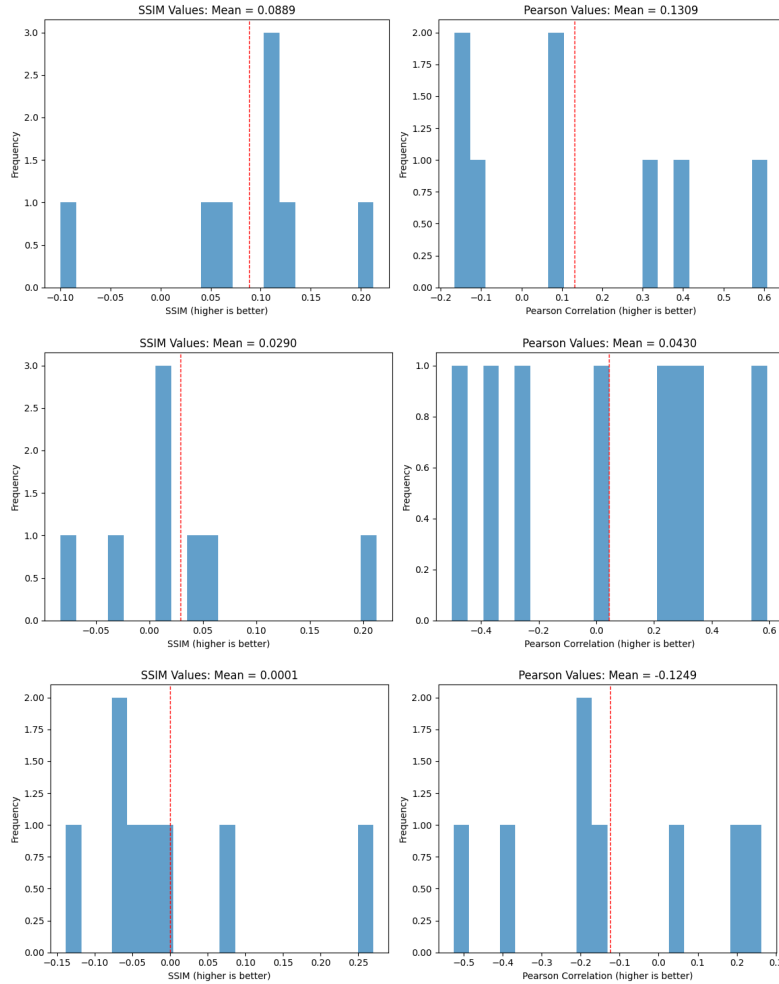


Figure 10: SSIM and Pearson Correlation distributions for MobileNetV2 with GradCAM λ values of 1.0 (top), 0.7 (middle), and 0.5 (bottom).

The table 7 summarizes the key results. Our analysis highlights several important observations:

1. Both SSIM and Pearson correlation values increase noticeably as the GradCAM λ value rises from 0.5 to 1.0. At $\lambda = 0.5$, the mean SSIM and Pearson correlation are near zero or negative (mean SSIM = 0.0001, mean Pearson = -0.1249), indicating minimal alignment between generated and real heatmaps.
2. At $\lambda = 0.7$, an improvement is observed, with mean SSIM increasing to 0.0290 and mean Pearson correlation becoming positive at 0.0430. This change signifies that emphasizing GradCAM loss more strongly begins to produce images that more closely align with real image attention regions.
3. At the highest tested $\lambda = 1.0$, we observe the greatest improvement, with mean SSIM rising further to 0.0889 and mean Pearson correlation reaching 0.1309. This clearly indicates significant alignment improvement, underscoring the effectiveness of higher GradCAM loss weighting.

Table 7: Mean SSIM and Pearson correlation values for different GradCAM λ settings.

GradCAM λ	Mean SSIM	Mean Pearson Correlation
0.5	0.0001	-0.1249
0.7	0.0290	0.0430
1.0	0.0889	0.1309

These quantitative results strongly suggest that greater emphasis on GradCAM loss during training improves the semantic alignment between generated and real images, enhancing the interpretability and effectiveness of GAN-generated data for classification tasks.

5.3 Classification Result

We evaluated classification accuracy across all 10 CIFAR-10 classes on a 10% subset, comparing performance before and after augmenting with Grad-CAM-supervised GAN images, as shown in Table 8. The results reveal mixed effects: several classes (Bird, Cat, and Airplane) show improvements, with Bird increasing most significantly by 20.2 percentage points, suggesting that Grad-CAM supervision helped the GAN generate more discriminative features for fine-grained classes. However, other categories such as Automobile, Truck, and Frog suffer notable drops (for example Truck drops 11.8 *pp*), indicating that the generated images may have introduced noise or poorly aligned features for those classes. These outcomes suggest that while Grad-CAM guidance can enhance class-specific performance, its effectiveness varies depending on how well the generator learns and preserves class-distinctive visual patterns.

Table 8: Per-class classification accuracy on a 10% subset of CIFAR-10 before and after augmenting with *Grad-CAM-supervised* GAN images.

Sample Class	Accuracy without Grad-CAM (%)	Accuracy with Grad-CAM (%)
Airplane	70.50	71.50
Automobile	68.00	56.80
Bird	68.50	88.70
Cat	77.60	83.70
Deer	90.90	86.60
Dog	52.40	53.30
Frog	88.20	77.10
Horse	66.20	66.20
Ship	93.00	92.50
Truck	67.90	56.10

5.4 Discussion

5.4.1 Key Findings

The experimental evidence supports our central claim that *feature-level* supervision with Grad-CAM can steer a GAN toward producing semantically richer images: higher λ_{GC} values improved both

SSIM and Pearson alignment scores (Section 10), and modest yet tangible classification gains corroborate the utility of these images for downstream tasks. This is encouraging, because it moves GAN-based data augmentation beyond blind realism toward *task-aware synthesis*.

5.4.2 Risks and Open Questions

First, we have not explored $\lambda_{GC} > 1.0$; excessively weighting the Grad-CAM term may over-constrain the generator, hurt diversity, or cause mode collapse. Second, our results are derived from a 10 % slice of CIFAR-10 at 32×32 resolution; sensitivity to higher resolutions, different domains, or larger class vocabularies remains unknown. Third, per-class outcomes (Table 8) reveal that mis-aligned synthetic samples can *degrade* accuracy, highlighting the need for automatic quality filters or adaptive λ_{GC} schedules. Finally, because the current Grad-CAM matching is not shift-invariant, small spatial translations between real and generated features can be penalised unfairly, reducing heat-map scores even when semantics match.

In summary, Grad-CAM supervision shows promise as a plug-and-play mechanism for generating more *classifier-relevant* images, but its effectiveness is highly sensitive to the weighting hyperparameter, class distribution, and spatial alignment assumptions. Future work should therefore explore adaptive λ tuning, translation-invariant similarity measures, and class-specific quality control pipelines to mitigate these risks and fully unlock the benefits of semantic GAN guidance.

6 Limitations and Future Directions

6.1 Limitations

Our current Grad-CAM-based supervision technique exhibits several limitations. First, the approach is not shift-invariant. This means that our GAN model is constrained to generate synthetic images predominantly with features centered within the image. Consequently, important class-specific features that appear off-center in real-world data are not effectively captured, limiting the practical utility of generated images for general classification tasks. Secondly, our methodology only leverages a subset of the CIFAR-10 dataset for Grad-CAM supervision, restricting the effectiveness of the approach. A comprehensive integration across the entire dataset could further enhance the feature alignment and semantic consistency between synthetic and real images. Lastly, our GAN model has not been trained to convergence due to computational constraints. This incomplete training potentially restricts the full benefits of Grad-CAM supervision from being realized, suggesting that a longer and more resource-intensive training regimen could significantly boost performance.

6.2 Future Directions

Several promising avenues exist for future research. First, addressing shift invariance issues is critical. Implementing translation-invariant metrics for heatmap comparison could ensure that spatially shifted but semantically similar features do not result in penalization, improving overall semantic alignment. Second, developing a robust dataset integration framework is essential. A more balanced mixing strategy between real and synthetic images would likely provide a fairer evaluation and greater benefits from synthetic data augmentation. Third, adaptive hyperparameter tuning could replace manual hyperparameter selection for Grad-CAM loss weighting. Dynamic adjustments of the hyperparameter based on training progression and feature alignment quality could significantly enhance the effectiveness of supervision. Fourth, exploring advanced GAN architectures designed to maintain or enhance generation quality while fully integrating Grad-CAM guidance may yield even more realistic and class-specific synthetic images. Finally, scaling the proposed methodology to higher-resolution and more complex datasets beyond CIFAR-10 will be crucial to evaluate its broader applicability and robustness in more challenging real-world settings.

7 Conclusion

The project has demonstrated that integrating Grad-CAM-based supervision into GAN training substantially enhances the semantic alignment of synthetic images with real image features, as indicated by improved Structural Similarity Index (SSIM) and Pearson correlation metrics. The positive correlation observed between the strength of Grad-CAM supervision and performance

metrics underscores the viability of this approach as a powerful data augmentation tool for improving classifier accuracy. Despite the identified limitations, such as the lack of shift-invariance, incomplete dataset utilization, and computational constraints preventing full training convergence, the approach shows clear potential. Addressing these challenges through shift-invariant methods, comprehensive dataset integration, adaptive hyperparameter tuning, more sophisticated GAN architectures, and scaling to complex, higher-resolution datasets constitute essential steps forward. By pursuing these directions, future research can harness the full potential of Grad-CAM-guided GAN-generated synthetic data, significantly benefiting various practical classification applications.

8 Administrative Details

Github Repository: <https://github.com/Anning-Li/IDL-Project>

Anushka Bhawe: Wrote code for the ResNet18, ResNet50, and PreActResNet models and trained on GPU to get a complete training time and accuracy analysis. Conducted research on related work, identifying key prior studies, comparing methodologies, and summarizing relevant findings that informed the direction of our research. Additionally, she helped structure the literature review to provide a comprehensive background.

Anning Li: Led the development of the methodology section by outlining the framework and running the model. Drew the high-level visualization graphs for the model. Implemented and trained both the classification and GAN models, performed baseline evaluations on the CIFAR-10 dataset, and prepared the visualizations and tables demonstrating the models' performance across different epochs. Managed the project repository on GitHub, and coordinated the integration of written content into the Overleaf project.

Urmi Dedhia: Implemented the GAN training pipeline and developed the Grad-CAM heatmap generation module for real and synthetic images. Integrated Grad-CAM region emphasis loss using SSIM and Pearson correlation into the generator's objective, and wrote the Methodology section detailing the model pipeline. Drafted the introduction and abstract, ensuring clarity in presenting the problem statement, research motivation, and key contributions. Synthesized core ideas concisely while making sure the introduction provided a logical flow into the methodology and results sections. Performed experimentations on GradCAM heatmap comparisons for MobileNetV2 and DCGAN generated images for CIFAR-10 data and evaluated the same. Also wrote the related sections for the same in the report. Additionally, ran experiments on a DCGAN for CIFAR-10 to assess its performance as a part of the baseline.

Andrew Zhang: Conducted experiments on MobileNetV2 by varying GradCAM loss weight (λ) values, systematically evaluating the impact on model interpretability metrics such as SSIM and Pearson correlation. Analyzed and consolidated these experimental results into Section 5 of the report, providing a detailed discussion on how different λ settings affected heatmap quality and model performance. Conducted research on methodology and future directions by analyzing existing approaches, evaluating their strengths and weaknesses, and helping refine our proposed framework.

References

- [1] Alexander Binder, Grégoire Montavon, Sebastian Bach, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers, 2016.
- [2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2019.
- [3] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, March 2018.
- [4] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks, 2017.

- [5] Paul Friedrich, Yannik Frisch, and Philippe C. Cattin. Deep generative models for 3d medical image synthesis, 2024.
- [6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
- [7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks, 2016.
- [11] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.
- [12] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.
- [13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019.
- [14] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009.
- [15] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study, 2018.
- [16] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.
- [17] Shuang Ma, Jianlong Fu, Chang Wen Chen, and Tao Mei. Da-gan: Instance-level image translation by deep attention generative adversarial networks (with supplementary materials), 2018.
- [18] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks, 2017.
- [19] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.
- [20] K. Pearson and Galton Laboratory for National Eugenics. *"Note on Regression and Inheritance in the Case of Two Parents"*. Proceedings of the Royal Society. Royal Society, 1895.
- [21] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.
- [22] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.
- [23] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016.
- [24] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019.
- [25] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, October 2019.
- [26] Tomohiro Shiraishi, Daiki Miwa, Teruyuki Katsuoka, Vo Nguyen Le Duy, Kouichi Taji, and Ichiro Takeuchi. Statistical test for attention map in vision transformer, 2024.

- [27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.
- [28] Ilya Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. Adagan: Boosting generative models, 2017.
- [29] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization, 2018.
- [30] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [31] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks, 2019.