Investigating the Potential of Auxiliary-Classifier GANs for Image Classification in Low Data Regimes

Amil Dravid\textsuperscript{1}, Florian Schiffers\textsuperscript{1}, Yunan Wu\textsuperscript{1}, Oliver Cossairt\textsuperscript{1}, Aggelos Katsaggelos\textsuperscript{1}

\textsuperscript{1}Northwestern University
Problem Definition and Contribution

- Convolutional neural networks (CNNs) rely on extremely large datasets to perform well on new data.
- We examine the potential for Auxiliary-Classifier GANs (AC-GANs) as a 'one-stop-shop' architecture for image classification and generation, particularly in low data regimes.
- We propose modifications to the typical AC-GAN framework: latent space sampling scheme and Wasserstein loss with gradient penalty.

Figure 1: Simple techniques can improve generated image quality even with limited data.
Generative Adversarial Networks (GANs)\(^1\)

- Generator (G) tries to create samples to "fool" the discriminator (D).
- Discriminator takes turns looking at real (x) and fake images (G(z)).

Figure 2: GAN Training Scheme.
Background: Auxiliary-Classifier GAN (AC-GAN)

- Auxiliary-Classifier GAN (AC-GAN) builds on the Conditional GAN (C-GAN) in order to improve image synthesis\(^2\).\(^3\).

- The discriminator outputs both the source of the input image (real or fake) and a second label corresponding to the input’s class.

Figure 3: AC-GAN Training Scheme.
Methods: Loss Function

- AC-GAN two-part objective:

\[
L_S = \mathbb{E} [\log P(S = \text{real}|X_{\text{real}})] + \mathbb{E} [\log P(S = \text{fake}|X_{\text{fake}})] \quad (1)
\]

\[
L_C = \mathbb{E} [\log P(C = c|X_{\text{real}})] + \mathbb{E} [\log P(C = c|X_{\text{fake}})], \quad (2)
\]

- Discriminator maximizes $L_S + L_C$. Generator maximizes $L_C - L_S$.

- $L_S$ becomes gradient-penalty Wasserstein loss\(^4\) to stabilize simultaneous image synthesis and classification.

\[
L_S = \mathbb{E} [D(x)] - \mathbb{E} [D(G(z))] + \lambda \Phi \quad (3)
\]
The Latent Space

Figure 4: Illustrating the Latent Space.
Methods: Latent Sampling Scheme

- **Truncation trick**: sample latent vector $z \sim p_z$ closer to the mode of the distribution, resulting in images with greater realism, but low diversity.
- We propose **feeding truncated samples into the classifier**.

Figure 5: Sampling the latent vector from a truncated distribution results in higher fidelity images, but lower diversity.
WAC-GAN-GPT
(Wasserstein AC-GAN with Gradient Penalty and Truncation).

Figure 6: WAC-GAN-GPT Training Scheme

\[
x, G(z) \quad x, G(z_{\text{trunc}}) \quad G(z) \quad G(z)
\]

\[
D_s(\cdot) \quad D_c(\cdot) \quad D_s(\cdot) \quad D_c(\cdot)
\]

Real/Fake? \quad Class? \quad Real/Fake? \quad Class?

Train D \quad \text{Train G}
Results and Discussion

- Ablation studies on varying training set sizes on **Fashion MNIST** to compare test accuracy.

<table>
<thead>
<tr>
<th>Train Size</th>
<th>Baseline CNN</th>
<th>AC-GAN</th>
<th>WAC-GAN-GP</th>
<th>AC-GAN with Truncation</th>
<th>WAC-GAN-GPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>77.5% ± 1.5</td>
<td>77.6% ± 1.7</td>
<td>77.9% ± 1.5</td>
<td>78.8% ± 1.5</td>
<td>79.8% ± 1.5</td>
</tr>
<tr>
<td>2500</td>
<td>83.5% ± 1.0</td>
<td>81.2% ± 2.1</td>
<td>84.4% ± 1.5</td>
<td>84.8% ± 1.1</td>
<td>86.0% ± 1.2</td>
</tr>
<tr>
<td>10000</td>
<td>86.4% ± 1.5</td>
<td>87.3% ± 1.3</td>
<td>87.6% ± 0.9</td>
<td>87.8% ± 0.7</td>
<td>88.4% ± 1.1</td>
</tr>
<tr>
<td>20000</td>
<td>87.5% ± 1.3</td>
<td>88.6% ± 1.6</td>
<td>88.1% ± 1.2</td>
<td>89.1% ± 0.5</td>
<td>89.8% ± 0.9</td>
</tr>
<tr>
<td>40000</td>
<td>90.3% ± 0.8</td>
<td>90.9% ± 0.8</td>
<td>91.0% ± 0.4</td>
<td>90.7% ± 0.8</td>
<td>91.3% ± 0.7</td>
</tr>
</tbody>
</table>

Table 1: Performance on Fashion MNIST test set based on varying training set sizes.

1) Baseline CNN
2) AC-GAN
3) Wasserstein AC-GAN with Gradient Penalty (WAC-GAN-GP)
4) AC-GAN with Truncation
5) Our WAC-GAN-GPT
Results and Discussion

- **T-SNE** on CNN embeddings for **real samples**, **AC-GAN samples**, and **WAC-GAN-GPT samples** based on **Fashion MNIST**.

- Average distance to center of class cluster: 7.83, 5.16, and 3.94 for the CNN, AC-GAN, and WAC-GAN-GPT, respectively.

- Standard deviations: 4.71, 2.17, and 1.76.

Figure 8: T-SNE visualizations on Fashion MNIST samples.
Results and Discussion

- Are low diversity but more representative images helpful?
- Find optimal truncation factor for each training set size experiment.
- Bilevel optimization: find optimal truncation factor $\tau$ while optimizing GAN:

$$\arg\min_{\tau} \text{CE}(f(G(z_{\tau}, c)), c)$$  \hspace{1cm} (4)

$$z \sim \mathcal{N}(0, I)$$  \hspace{1cm} (5)

$$z_{\tau} = \text{sgn}(z) \cdot \min(|z|, \tau)$$  \hspace{1cm} (6)

<table>
<thead>
<tr>
<th>Training Size</th>
<th>500</th>
<th>2500</th>
<th>10000</th>
<th>20000</th>
<th>40000</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau$</td>
<td>0.89 ± 0.14</td>
<td>1.05 ± 0.12</td>
<td>1.20 ± 0.13</td>
<td>1.49 ± 0.08</td>
<td>1.63 ± 0.05</td>
</tr>
</tbody>
</table>

Table 2: Optimal Truncation Factors for various training set sizes.
Results and Discussion

- Compare CIFAR10 test accuracy, and then CIFAR10.1v6$^6$ to compare domain generalizability.

<table>
<thead>
<tr>
<th></th>
<th>AlexNet</th>
<th>AC-GAN</th>
<th>WAC-GAN-GPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR 10</td>
<td>70.5% ± 0.5</td>
<td>70.1% ± 0.8</td>
<td><strong>72.9% ± 0.7</strong></td>
</tr>
<tr>
<td>CIFAR 10.1v6</td>
<td>53.5% ± 1.0</td>
<td>56.4% ± 1.1</td>
<td><strong>59.3% ± 0.6</strong></td>
</tr>
</tbody>
</table>

Table 3: CIFAR test performance and generalizability.

- COVID-19 Detection on 128x128 chest X-rays.

<table>
<thead>
<tr>
<th></th>
<th>CNN</th>
<th>AC-GAN</th>
<th>WAC-GAN-GPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>COVID-19</td>
<td><strong>94.0% ± 1.5</strong></td>
<td><strong>95.5% ± 0.5</strong></td>
<td><strong>97.6% ± 0.9</strong></td>
</tr>
</tbody>
</table>

Table 4: COVID-19 test performance.
Summary and Conclusion

- AC-GANs can achieve competitive performance with standard CNNs.
  - Particular performance gains in lower data regimes.
- Modifications: Wasserstein-GP + truncation.
- Future work: more diverse datasets, higher resolution images.
- More advanced techniques: adaptive discriminator augmentation or progressive growing.\(^7,8\)

![Image](image.png)

Figure 9: Naively applying data augmentation transformations leaks through to generated images.
References


Contact: amildravid2023@u.northwestern.edu

IEEE Xplore Link: https://ieeexplore.ieee.org/document/9747286

arXiV Link: https://arxiv.org/abs/2201.09120

Code: https://github.com/avdravid/AC-GANS-FOR-IMAGE-CLASSIFICATION