Training Self-supervised Class-conditional GANs with Classifier Gradient Penalty and Dynamic Prior

Jeongik Cho

Overview

Classifier Gradient Penalty GAN (CGPGAN) is a self-supervised GAN framework that enables clustering and class-conditional generation **without requiring any labels, similarity metrics, or prior knowledge of category distributions**.

The model consists of a generator G, a discriminator D, and a classifier Q. The generator synthesizes samples from a continuous latent vector z and a categorical latent vector c_f, producing $G(z, c_f)$. The classifier Q is trained to recover c_f from generated data, and its prediction on real data is used to estimate the categorical prior.

To prevent the classifier decision boundary from converging to narrow and unstable low-density regions, CGPGAN introduces a **classifier gradient penalty**, which encourages smoother decision boundaries that span broader regions of the data space.

We also propose a **codebook extension**, in which the categorical latent vector selects a learned embedding from a trainable codebook, instead of using a one-hot vector, to improve discrete representation and the interpretability of generated data.

Loss Functions

Classifier Loss. The classifier is trained using a combination of cross-entropy loss and a classifier gradient penalty:

$$L_{cls} = \mathbb{E}_{z,c_f}\left[-c_f \cdot \log Q(G(z, c_f))\right] \tag{1}$$

$$L_{cgp} = \mathbb{E}_{z,c_f}\left\|\nabla_{G(z,c_f)}\left((1 - Q(G(z, c_f)) \cdot c_f)^2\right)\right\|_2^2 \tag{2}$$

$$L_q = \lambda_{cls} L_{cls} + \lambda_{cgp} L_{cgp} \tag{3}$$

The gradient penalty term L_{cgp} encourages the decision boundary to align with broader low-density regions, improving clustering robustness. The hyperparameter λ_{cgp} controls the coarseness of cluster separation.

Adversarial Loss. The generator and discriminator are trained via class-conditional adversarial loss. Since ground-truth labels are unavailable, the classifier provides pseudo-labels:

$$\tilde{c}_r = \text{argmax\_onehot}(Q(x)) \tag{4}$$

The adversarial objectives are:

$$L^d_{adv} = \mathbb{E}_{x,z,c_f}\left[A_d\left(D(x) \cdot \tilde{c}_r,\ D(G(z, c_f)) \cdot c_f\right)\right] \tag{5}$$

$$L^g_{adv} = \mathbb{E}_{z,c_f}\left[A_g\left(D(G(z, c_f)) \cdot c_f\right)\right] \tag{6}$$

$$L_d = L^d_{adv}, \quad L_g = L^g_{adv} \tag{7}$$

Here, A_d and A_g denote the adversarial loss functions for the discriminator and generator, respectively.

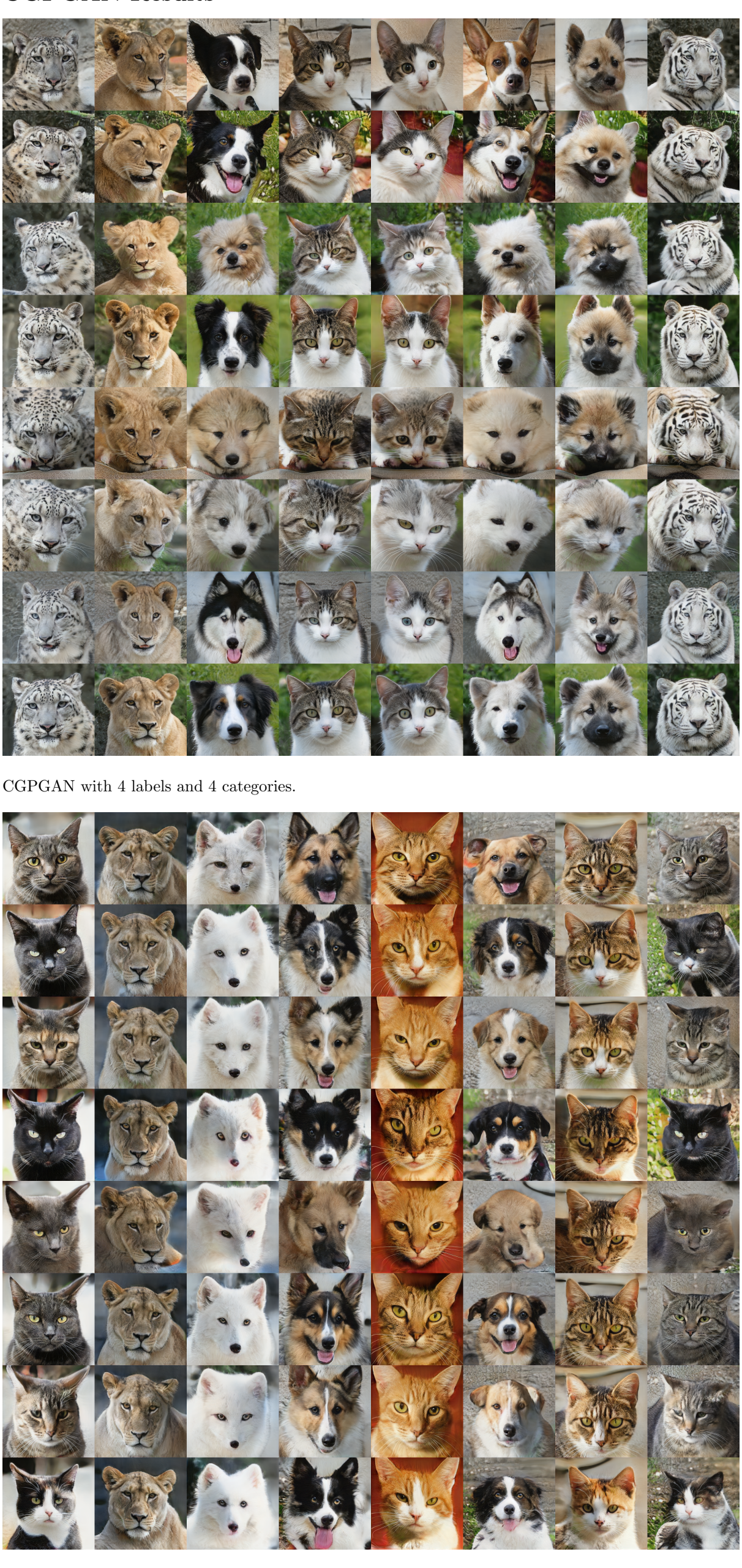
Prior Update. The categorical prior $P(C)$ is dynamically estimated as the expectation over the classifier's output:

$$P(C) \approx \mathbb{E}_x[Q(x)]$$

To avoid premature mode collapse, this update is disabled during the early stage of training. Initially, $Q(x)$ is normalized to encourage a uniform distribution.

Codebook Architecture. We extend CGPGAN with a codebook mechanism to improve discrete representation and the interpretability of generated data. Instead of directly feeding a one-hot vector into the generator, a trainable embedding vector is selected from a codebook according to the categorical index.

CGPGAN Results



CGPGAN with 4 labels and 4 categories.

CGPGAN with 16 labels and 16 categories.

The figures above show images generated by CGPGAN. Each column shares the same categorical latent vector, and each row shares the same continuous latent vector. Without using any similarity metric, the model successfully disentangles discrete features (such as animal species) and continuous features (such as pose). As the number of labels and categories increases, each cluster contains fewer real samples, leading to reduced intra-cluster diversity in generated outputs.

Full codes for the experiments are available at `https://github.com/jeongik-jo/CGPGAN_codebook`