Energy based gan: https://arxiv.org/pdf/1609.03126.pdf
It changes the discriminator to an autoencoder. The input of discriminator is image and output is also image. The loss of discriminator comes from the distance between two images. The structure of began:

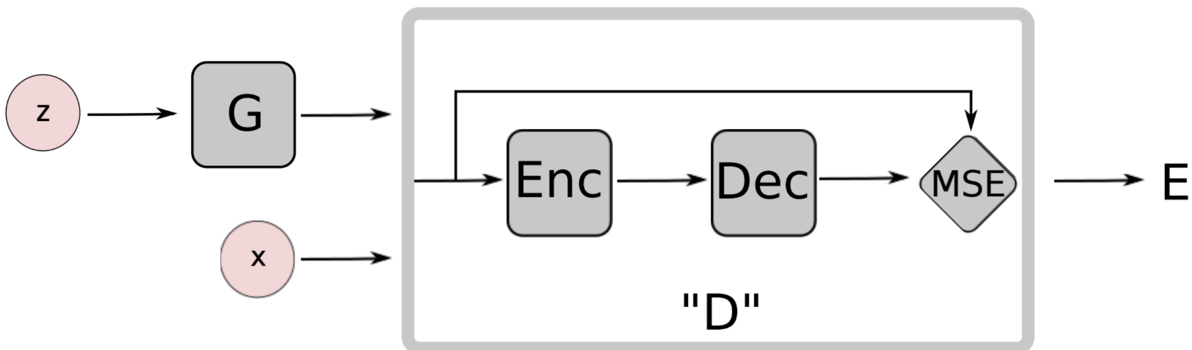

Figure 1: EBGAN architecture with an auto-encoder discriminator.

Wasserstein GAN: https://arxiv.org/pdf/1701.07875.pdf

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.
Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

The f function in the algorithm is the discriminator. Basically, WGAN changes the metric of judging the loss. It uses L1 divergence instead of Jensen-Shannon divergence (binary entropy error) to estimate the error. The paper proves the Wasserstein distance is weaker than Jensen-Shannon divergence so that it is more stable during the training but harder to converge.

Boundary Equilibrium Gan: https://arxiv.org/abs/1703.10717
The idea comes from above 2 papers. The most different part is its training goal.

The BEGAN objective is:

$$\begin{cases} \mathcal{L}_D = \mathcal{L}(x) - k_t.\mathcal{L}(G(z_D)) & \text{for } \theta_D \\ \mathcal{L}_G = \mathcal{L}(G(z_G)) & \text{for } \theta_G \\ k_{t+1} = k_t + \lambda_k(\gamma\mathcal{L}(x) - \mathcal{L}(G(z_G))) & \text{for each training step } t \end{cases}$$

We use Proportional Control Theory to maintain the equilibrium $\mathbb{E}\left[\mathcal{L}(G(z))\right] = \gamma\mathbb{E}\left[\mathcal{L}(x)\right]$. This is implemented using a variable $k_t \in [0, 1]$ to control how much emphasis is put on $\mathcal{L}(G(z_D))$ during gradient descent. We initialize $k_0 = 0$. λ_k is the proportional gain for k; in machine learning terms, it is the learning rate for k. We used 0.001 in our experiments. In essence, this can be thought of as a form of closed-loop feedback control in which k_t is adjusted at each step to maintain equation 5.

The loss function is the L1 distance between input image and output image of discriminator (actually an autoencoder). I think the theoretical part of the model is not convincible although the results are pretty good.