Parallel Optimal Transport GAN
Supplementary Material

Gil Avraham* Yan Zuo*
Tom Drummond
ARC Centre of Excellence for Robotic Vision, Monash University, Australia
{gil.avraham, yan.zuo, tom.drummond}@monash.edu

1. Converting latent vectors into a soft decision forest

Soft internal decision function The soft decision function held by each internal decision node in the soft decision forest is defined as:

$$d_n(\boldsymbol{z}, \boldsymbol{\Theta}) = \sigma(z_n - t_n) \tag{1}$$

$\sigma(z) = (1 + e^{-z})^{-1}$ denotes a sigmoid function and Θ represents the parameters of the decision forest. z_n is the activation value of the latent vector and t_n is a threshold value for the decision node d_n which z_n is compared against. The blending function $\mu_\ell(\boldsymbol{z}, \Theta)$ dictates the portions allocated to the values q_ℓ held by each leaf ℓ towards a tree's final output:

$$\mu_\ell(\boldsymbol{z}|\Theta) = \prod_{n \in \mathcal{N}} d_n(z, \Theta)^{1_{\ell \swarrow n}} \bar{d}_n(\boldsymbol{z}, \Theta)^{1_{\ell \searrow n}} \tag{2}$$

$\bar{d}_n(\boldsymbol{z}, \Theta)$ is the complement of $d_n(\boldsymbol{z}, \Theta)$ (*i.e.* $1 - d_n(\boldsymbol{z}, \Theta)$). The indicator function is denoted by 1_C with a condition:

$$1_C = \begin{cases} 1, & \text{if } C = 1 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The conditions $\ell \swarrow n$ and $\ell \searrow n$ are defined as:

$$\ell \swarrow n = \begin{cases} 1, & \text{if } z_n \leq t_n \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$\ell \searrow n = \begin{cases} 1, & \text{if } z_n > t_n \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The soft decision tree outputs a weighted sum prediction given by:

$$\mathcal{Q}(\boldsymbol{z}, \boldsymbol{\Theta}) = \sum_\ell \mu_\ell(\boldsymbol{z}|\Theta) q_\ell \tag{6}$$

*Authors contributed equally

Soft residual decision forest For combining the ensemble of soft decision trees, we employ the residual method in [5], and multiplicatively combine distributions to generate the final output. Hence the transformed latent vector, z'_g , is learned as the product of each individual decision tree's given output *i.e*:

$$z'_g = \prod_{t=1}^{\mathcal{T}} \mathcal{Q}^t(\mathcal{D}^t(\boldsymbol{z}, \Theta^t)) \tag{7}$$

Where $\mathcal{D}^t$ represents the internal decision node functions in the decision tree t.

2. Ablation Study on the choice of $\mathcal{F}$

Various configurations were evaluated as preliminary tests to verify the effectiveness of optimal transport on a low dimension representation. Here, we show performance in Inception Score and FID Score across various configurations:
1) We tested naïve L_2 regularisation by omitting Algorithm 1 and using the L_2 cost between randomly sampled z_r, z_g (WGAN-GP+L_2).
2) Using $c(a, b) = \|a - b\|^2$ as the cost function (POT-GAN (L_2))
3) Using $c(a, b) = \|a - \mathcal{F}_\theta(b)\|^2$ as the cost function, where $\mathcal{F}_\theta$ is a 3-layer Multi-Layer Perceptron (MLP) parameterised by θ (POT-GAN (MLP)).

Inception Score (CIFAR-10)			
WGAN-GP +L_2	POT-GAN (L_2)	POT-GAN (MLP)	POT-GAN (LTF)
5.92±0.08	6.62±0.05	6.72±0.05	**6.87±0.04**
FID Score (CIFAR-10)			
65.3	34.1	33.5	**32.5**

3. Critic Loss Curves

Fig. 1 shows the critic loss curves for POT-GAN and WGAN-GP over 100k training iterations on the CIFAR-10 dataset. These loss curves correlate well with the improved convergence rates of our proposed method.

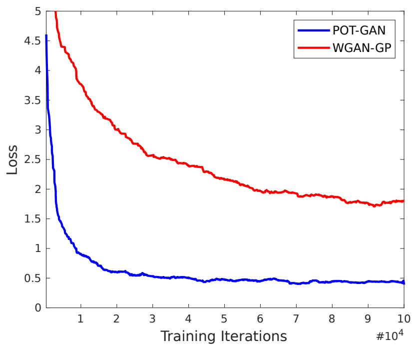


Figure 1: Critic loss curves over a 100k training iterations for POT-GAN and WGAN-GP on CIFAR-10

4. Architectures

GAN architecture Our GAN architecture is similar to generator and discriminator networks in [3]. The generator network is composed of 2-strided 5×5 deconvolution layers with batch normalisation and ReLU activation. The critic network consists of 2-strided 5×5 convolution layers with Leaky ReLU activation. Layer normalisation [1] is used as a drop-in replacement for batch normalisation in the critic network following the recommendation of [2]. Upsampling and downsampling is achieved via these strided deconvolution and convolution layers respectively.

VAE architecture Our VAE architecture is based off the architecture in [3]. The decoder network is made of 2-strided 5×5 deconvolution layers with batch normalisation and ReLU activation. The encoder network consists of 2-strided 5×5 convolution layers with batch normalisation and ReLU activation. Upsampling and downsampling is achieved via these strided deconvolution and convolution layers respectively.

Soft decision forest architecture Our soft decision forest is composed of 8 soft decision trees, each of 6-depth. Each decision tree contains $2^6 - 1 = 63$ internal decision nodes and $2^6 = 64$ leaf nodes. Hence, in total the soft decision forest contains $8 \times 63 = 504$ decision nodes and $8 \times 64 = 512$ leaf nodes. The latent vector z_g is first upsampled from 128 dimensions to 504 dimensions to match the required number of decision nodes using a fully connected linear layer. Each decision node is assigned one threshold value, and each leaf node is assigned a 128 vector to match the output transformed vector z'_g.

Generator $G(z)$

	Kernel Size	Batch Norm	Activation	Resample	Output Shape
z	-	No	-	-	128
Linear + Reshape	-	Yes	ReLU	-	$512 \times 4 \times 4$
Deconv	5×5	Yes	ReLU	$\uparrow$	$512 \times 8 \times 8$
Deconv	5×5	Yes	ReLU	$\uparrow$	$256 \times 16 \times 16$
Deconv	5×5	Yes	ReLU	$\uparrow$	$128 \times 32 \times 32$
Deconv + Tanh	5×5	Yes	-	$\uparrow$	$3 \times 64 \times 64$

Table 1: Generator network architecture. $\uparrow$ represents upsampling via strided deconvolution

Critic $D(x)$

	Kernel Size	Batch Norm	Activation	Resample	Output Shape
Conv	5×5	No	Leaky ReLU	$\downarrow$	$64 \times 32 \times 32$
Conv	5×5	No	Leaky ReLU	$\downarrow$	$128 \times 16 \times 16$
Conv	5×5	No	Leaky ReLU	$\downarrow$	$256 \times 8 \times 8$
Conv	5×5	No	Leaky ReLU	$\downarrow$	$512 \times 4 \times 4$
Reshape + Linear	-	No	-	-	1

Table 2: Critic network architecture. $\downarrow$ represents downsampling via strided convolution

Decoder $Dec(z)$

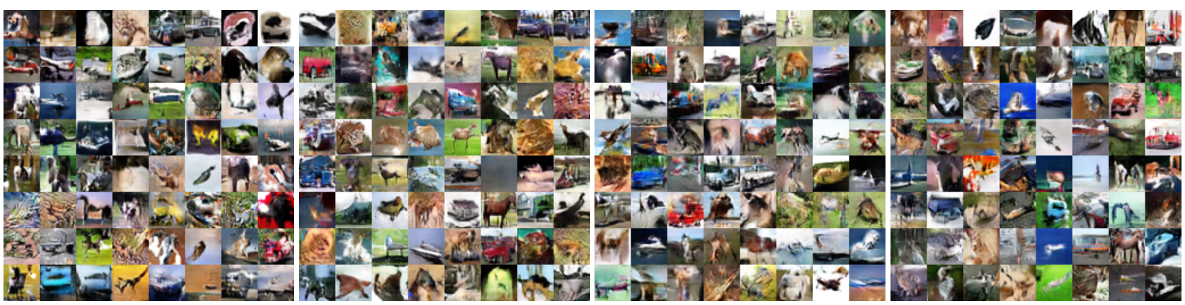
	Kernel Size	Batch Norm	Activation	Resample	Output Shape
z	-	-	128		
Linear + Reshape	-	No	ReLU	-	$512 \times 4 \times 4$
Deconv	5×5	Yes	ReLU	$\uparrow$	$256 \times 8 \times 8$
Deconv	5×5	Yes	ReLU	$\uparrow$	$128 \times 16 \times 16$
Deconv	5×5	Yes	ReLU	$\uparrow$	$64 \times 32 \times 32$
Deconv + Tanh	5×5	Yes	-	$\uparrow$	$3 \times 64 \times 64$

Table 3: Decoder network architecture. $\uparrow$ represents upsampling via strided deconvolution

Encoder $Enc(x)$

	Kernel Size	Batch Norm	Activation	Resample	Output Shape
Conv	5×5	Yes	ReLU	$\downarrow$	$64 \times 32 \times 32$
Conv	5×5	Yes	ReLU	$\downarrow$	$128 \times 16 \times 16$
Conv	5×5	Yes	ReLU	$\downarrow$	$256 \times 8 \times 8$
Conv	5×5	Yes	ReLU	$\downarrow$	$512 \times 4 \times 4$
Reshape + Linear	-	No	-	-	128

Table 4: Encoder network architecture. $\downarrow$ represents downsampling via strided convolution



(a) DCGAN [3] (b) VEEGAN [4] (c) WGAN-GP [2] (d) POT-GAN (Ours)

Figure 2: Additional qualitative results on the CIFAR-10 dataset.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 2

[2] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017. 2, 4

[3] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2, 4

[4] A. Srivastava, L. Valkoz, C. Russell, M. U. Gutmann, and C. Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3310–3320, 2017. 4

[5] Y. Zuo and T. Drummond. Fast residual forests: Rapid ensemble learning for semantic segmentation. In *Conference on Robot Learning*, pages 27–36, 2017. 1