MO-PaDGAN: Generating Diverse Designs with Multivariate Performance Enhancement

Wei Chen [1] **Faez Ahmed** [2]

Abstract

Deep generative models have proven useful for automatic design synthesis and design space exploration. However, they face three challenges when applied to engineering design: 1) generated designs lack diversity, 2) it is difficult to explicitly improve all the performance measures of generated designs, and 3) existing models generally do not generate high-performance novel designs, outside the domain of the training data. To address these challenges, we propose MO-PaDGAN, which contains a new Determinantal Point Processes based loss function for probabilistic modeling of diversity and performances. Through a real-world airfoil design example, we demonstrate that MO-PaDGAN expands the existing boundary of the design space towards high-performance regions and generates new designs with high diversity and performances exceeding training data.

1. Introduction

A designer wants good design solutions which are creative and meets multiple performance requirements. By manually and iteratively exploring design ideas using experience and design heuristics, the designers take the risks of 1) wasting time on unfavorable or even invalid design candidates and 2) not exploring as deeply as they might want to. While recent advances in machine learning assisted automatic design synthesis and design space exploration are promising, the current methods are still far from this ideal picture. To model a design space, researchers have used deep generative models like variational autoencoders (VAEs) (Kingma & Welling, 2013) and generative adversarial networks (GANs) (Goodfellow et al., 2014), as they can learn the distribution of existing designs. The hope is that by learning an underlying *latent space*, which can represent most designs, one can automatically synthesize many new designs from the low-dimensional *latent vectors*, which makes design exploration more efficient due to the reduced dimensionality (Chen et al., 2017; Chen & Fuge, 2019; Chen et al., 2019). However, unlike image generation tasks where these generative models are commonly applied, engineering design problems typically have multiple performance (or quality) measures. Each performance measure quantifies how well a design achieves its intended goals and is defined based on the specific problem. For example, beam design problems often have the compliance value (Bendsoe & Sigmund, 2004) or both the compliance and natural-frequency (Ahmed et al., 2016) as the performance measures. For aerodynamic wing design, researchers have defined performance using measures like the lift-to-drag ratio (Chen et al., 2019) or the inverse of the drag coefficient (Shu et al., 2020).

Current state-of-the-art generative models have no mechanism of explicitly promoting design generation with improved performance or diversity. In this work, we focus on addressing the problem of simultaneously maximizing diversity and (possibly multivariate) performance of generated designs. Specifically, we develop a new loss function, based on Determinantal Point Processes (DPPs) (Kulesza & Taskar, 2012), for generative models to encourage both high-performance and diverse design generation. Using this loss function, we develop a new variant of GAN, named MO-PaDGAN (Multi-Objective Performance Augmented Diverse Generative Adversarial Network). We show that it can generate new samples with a better coverage of the design space and improvement in all performance measures compared to a baseline GAN. More importantly, we found that MO-PaDGAN can expand the existing boundary of the design space towards high-performance regions outside the training data, which indicates its ability of generating novel high-performance designs.

One closely related work is the GDPP method (Elfeki et al., 2019), where the authors devised an objective term that matches the diversity of generated data with training data. The diversity is modeled by the DPP kernel. MO-PaDGAN differs from this method in two aspects. First, MO-PaDGAN aims to maximize the diversity of generated samples rather than matching it with training data. Thus, MO-PaDGAN can generate diverse samples even when the original training data is biased in favor of a few modes, while GDPP will mimic the bias in generated samples. Second, GDPP does not consider the performance of generated samples, whereas we incorporate (possibly multivariate) performance measurements into the DPP kernel and encourage gener-

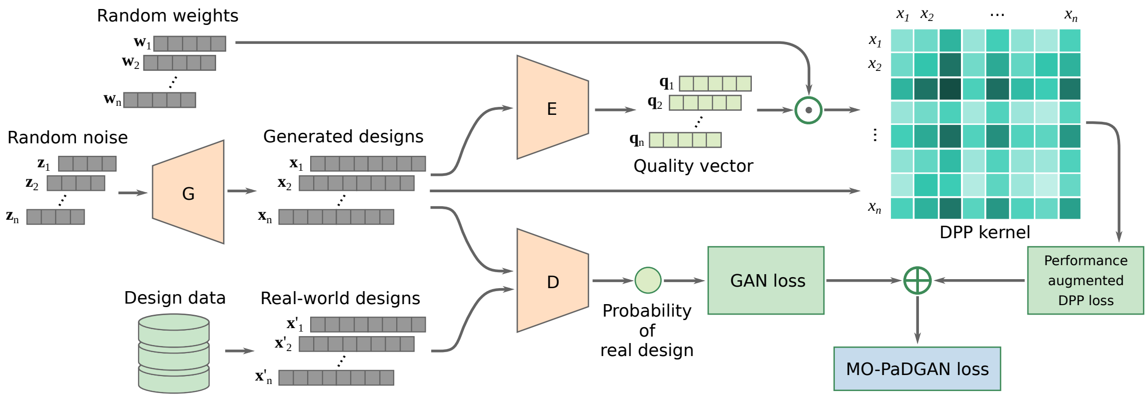

Figure 1. Architecture of MO-PaDGAN. The operator $\odot$ does the performance aggregation.

ation of high-performance samples. This is important in engineering design settings as we want the generated designs to not only look realistic, but also be useful. The contributions and novelty of this work are as follows:

1. We propose a novel design generation method that simultaneously encourage generation of diverse and high-performance designs.
2. We propose a way to incorporate multivariate performance measurements into the DPP kernel-based loss function of GAN, so that the generated samples have higher average and peak performance than training data in all dimensions.
3. We find that MO-PaDGAN can expand the design space boundary towards high-performance regions that it had not seen from existing data.

2. Background

Below we provide background on GANs and DPP kernels.

2.1. Generative Adversarial Nets

Generative Adversarial Networks (Goodfellow et al., 2014) model a game between a generative model (*generator*) and a discriminative model (*discriminator*). The generator G maps an arbitrary noise distribution to the data distribution (*i.e.*, the distribution of designs in our scenario), thus can generate new data; while the discriminator D tries to perform classification, *i.e.*, to distinguish between real and generated data. Both G and D are usually built with deep neural networks. As D improves its classification ability, G also improves its ability to generate data that fools D. Thus, a GAN has the following objective function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim P_{data}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim P_{\mathbf{z}}}[\log(1-D(G(\mathbf{z})))], \tag{1}$$

where $\mathbf{x}$ is sampled from the data distribution P_{data}, $\mathbf{z}$ is sampled from the noise distribution $P_{\mathbf{z}}$, and $G(\mathbf{z})$ is the generator distribution. A trained generator thus can map from a predefined noise distribution to the distribution of designs. The noise input $\mathbf{z}$ is considered as the latent representation of the data, which can be used for design synthesis and exploration.

2.2. Decomposition of a DPP kernel

DPP kernels can be decomposed into quality and diversity parts (Kulesza & Taskar, 2012). The $(i,j)^{th}$ entry of a positive semi-definite DPP kernel L can be expressed as:

$$L_{ij} = q_i\ \phi(i)^T\ \phi(j)\ q_j. \tag{2}$$

We can think of $q_i \in R^+$ as a scalar value measuring the quality of an item i, and $\phi(i)^T\ \phi(j)$ as a signed measure of similarity between items i and j. The decomposition enforces L to be positive semidefinite. Suppose we select a subset S of samples, then this decomposition allows us to write the probability of this subset S as the square of the volume spanned by $q_i\phi_i$ for $i \in S$ using the equation below:

$$\mathbb{P}_L(S) \ \propto\ \prod_{i\in S}(q_i{}^2)\det(K_S), \tag{3}$$

where K_S is the similarity matrix of S. As item i's quality q_i increases, so do the probabilities of sets containing item i. As two items i and j become more similar, $\phi_i{}^T\phi_j$ increases and the probabilities of sets containing both i and j decrease. The key intuition of MO-PaDGAN is that if we can integrate the probability of set selection from Eq. (3) to the loss function of any generative model, then while training it will be encouraged to generate high probability subsets, which will be both diverse and high-performance.

3. Methodology

MO-PaDGAN adds a *performance augmented DPP loss* to a standard GAN architecture which measures the diversity and performance of a batch of generated designs during training. The overall model architecture of MO-PaDGAN is shown in Fig. 1. We describe the DPP kernel part next.

3.1. Creating a DPP kernel

We create the kernel L for a sample of points generated by MO-PaDGAN from known inter-sample similarity values and performance vector.

The similarity terms $\phi(i)^T\phi(j)$ can be derived using any similarity kernel, which we represent using $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(i)^T\phi(j)$ and $\|\phi(i)\| = \|\phi(j)\| = 1$. Here $\mathbf{x}_i$ is a vector representation of a design. Note that in a DPP model, the quality of an item is a scalar value representing design performance, like compliance, displacement, drag-coefficient, *etc.* The performance can be estimated using an external model (like a physics-based simulator). For multivariate performance, we use a *performance aggregator*, which outputs a scalar quality value (q_i) by taking a weighted sum of multiple dimensions of performance (p) defined as $q(\mathbf{x}) = \sum_{j=1}^{K} w_j p_j(\mathbf{x})$. For each design, the weights $w_1, ..., w_K$ are positive numbers sampled uniformly at random and sum to 1. Maximizing the weighted sum of objectives gradually pushes the non-dominated Pareto set of generated samples in the performance space to have higher values. While more complex performance aggregators like the Chebyshev distance from an ideal performance vector can also be used in our method, we used the commonly used weighted sum to have fewer assumptions about the performance space.

3.2. Performance Augmented DPP Loss

Our performance augmented DPP loss models diversity and performance simultaneously and gives a lower loss to sets of designs which are both high-performance and diverse. Specifically, we construct a kernel matrix L_B for a generated batch B based on Eq. (2). For each entry of L_B, we have

$$L_B(i,j) = k(\mathbf{x}_i, \mathbf{x}_j)\left(q(\mathbf{x}_i)q(\mathbf{x}_j)\right)^{\gamma_0}, \tag{4}$$

where $\mathbf{x}_i, \mathbf{x}_j \in B$, $q(\mathbf{x})$ is the quality function at $\mathbf{x}$, and $k(\mathbf{x}_i, \mathbf{x}_j)$ is the similarity kernel between $\mathbf{x}_i$ and $\mathbf{x}_j$. For a given kernel, DPP decomposition does not allow us to change the trade-off between quality and diversity. To allow this, we adjust the dynamic range of the quality scores by using an exponent (γ_0) as a parameter to change the distribution of quality. A larger γ_0 increases the relative importance of quality as compared to diversity, which provides the flexibility to a user of MO-PaDGAN in deciding emphasis on quality vs diversity.

The performance augmented DPP loss is expressed as

$$\mathcal{L}_{\text{PaD}}(G) = -\frac{1}{|B|}\log\det(L_B) = -\frac{1}{|B|}\sum_{i=1}^{|B|}\log\lambda_i, \tag{5}$$

where λ_i is the i-th eigenvalue of L_B. We add this loss to the vanilla GAN's objective in Eq. (1) and form a new objective:

$$\min_G \max_D V(D,G) + \gamma_1 \mathcal{L}_{\text{PaD}}(G), \tag{6}$$

where γ_1 controls the weight of $\mathcal{L}_{\text{PaD}}$(G). For the backpropogation step, to update the weight θ_G^i in the generator in terms of $\mathcal{L}_{\text{PaD}}(G)$, we descend its gradient based on the chain rule:

$$\frac{\partial \mathcal{L}_{\text{PaD}}(G)}{\partial \theta_G^i} = \sum_{j=1}^{|B|}\left(\frac{\partial \mathcal{L}_{\text{PaD}}(G)}{\partial q(\mathbf{x}_j)}\frac{dq(\mathbf{x}_j)}{d\mathbf{x}_j} + \frac{\partial \mathcal{L}_{\text{PaD}}(G)}{\partial \mathbf{x}_j}\right)\frac{\partial \mathbf{x}_j}{\partial \theta_G^i}, \tag{7}$$

where $\mathbf{x}_j = G(\mathbf{z}_j)$. Equation (7) indicates a need for $dq(\mathbf{x})/d\mathbf{x}$, which is the gradient of the quality function. In practice, this gradient is accessible when the quality is evaluated through a performance estimator that is differentiable, like adjoint-based solver methods. If the gradient of a performance estimator is not available, one can either use numerical differentiation or approximate the quality function using a differentiable surrogate model (*e.g.*, a neural network-based surrogate model, as used in our experiments).

4. Experimental Results

In this section, we demonstrate the merit of modeling performance and diversity simultaneously by applying MO-PaDGAN on a real-world airfoil shape generation problem and comparing it against a vanilla GAN.

An airfoil is the cross-sectional shape of a wing or a propeller/rotor/turbine blade. In this example, we use the UIUC airfoil database[1] as our data source. It provides the geometries of nearly 1,600 real-world airfoil designs. Each design is represented by discrete 2D coordinates along their upper and lower surfaces. We preprocessed and augmented the dataset based on Chen et al. (2019) to generate a dataset of 38,802 airfoils. We use two performance measures for designing the airfoils — the lift coefficient (C_L) and the lift-to-drag ratio (C_L/C_D). These two are common objectives in aerodynamic design optimization problems and have been used in different multi-objective optimization studies (Park & Lee, 2010). We use XFOIL (Drela, 1989) for computational fluid dynamics (CFD) simulations and compute C_L and C_D values[2]. We scaled the performance scores between 0 and 1. To provide the gradient of the quality function for Eq. (7), we trained a neural network-based surrogate model on all 38,802 airfoils to approximate both C_L and C_D.

To demonstrate the effectiveness of MO-PaDGAN, we compare it with a vanilla GAN. We use a RBF kernel with

[1] http://m-selig.ae.illinois.edu/ads/coord_database.html

[2] We set $C_L = C_L/C_D = 0$ for unsuccessful simulations.

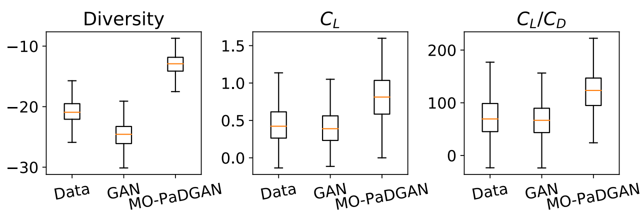

Figure 2. Diversity and performance statistics of randomly sampled airfoils.

a bandwidth of 1 when constructing L_B in Eq. (4), *i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-0.5\|\mathbf{x}_i - \mathbf{x}_j\|^2)$. We set $\gamma_0 = 5$ and $\gamma_1 = 0.2$ for MO-PaDGAN. We used a residual neural network (ResNet) (He et al., 2016) as the surrogate model and a BézierGAN (Chen et al., 2019; Chen & Fuge, 2018) to generate airfoils. For simplicity, we refer to the BézierGAN as a vanilla GAN and the BézierGAN with loss $\mathcal{L}_{PaD}$ as a MO-PaDGAN in the rest of the paper.

We measure the diversity of generated designs using the log determinant of the similarity matrix:

$$\text{Diversity} = \log\det(L_{S_i}), \tag{8}$$

where $S_i \subseteq Y$ is a random subset of Y (the set of generated samples or training data), and L_{S_i} is the similarity matrix of S_i with entries $L_{S_i}(j,k) = k(\mathbf{x}_j, \mathbf{x}_k)$ for each $\mathbf{x}_j, \mathbf{x}_k \in S_i$. We evaluate the diversity for 1000 times. Each time we randomly sample 100 designs from Y (which contains 1000 airfoils). We show the statistics of computed diversity in Fig. 2, together with two performance measures (C_L and C_L/C_D) of Y. It shows that MO-PaDGAN can generate samples with higher diversity and performances than training data and samples from the vanilla GAN.

To compare the distribution of real and generated airfoils in the design space, we map randomly sampled airfoils into a two-dimensional space through t-SNE, as shown in Figure 3. The results indicate that comparing with a vanilla GAN, MO-PaDGAN generates airfoils that are further away from training data, driven by the DPP loss.

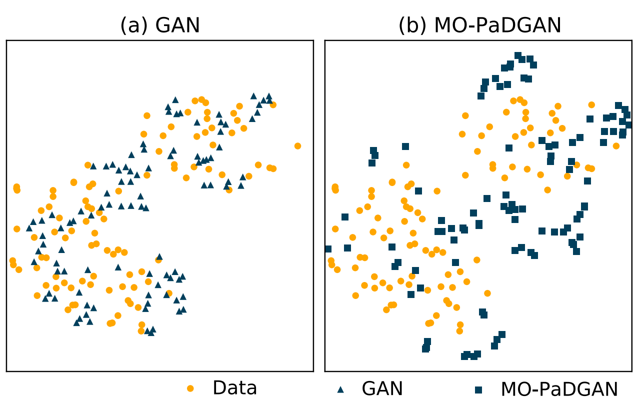

Figure 3. Randomly sampled airfoils embedded into a 2D space via t-SNE. MO-PaDGAN expands the boundary of training data.

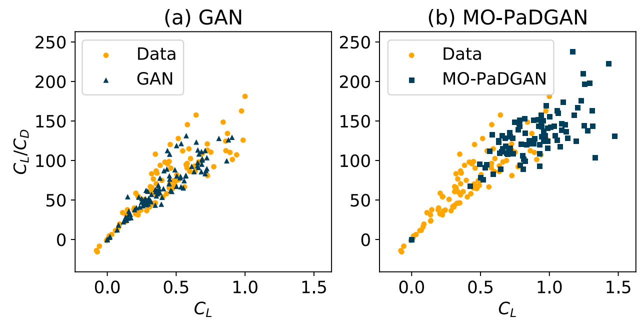

Figure 4. Performance space visualization for airfoils shown in Fig. 3 shows MO-PaDGAN improves both performance objectives.

Rank	Data		GAN		MO-PaDGAN	
	C_L	C_L/C_D	C_L	C_L/C_D	C_L	C_L/C_D
1	1.00	181.14	0.91	131.42	1.47	237.68
2	0.99	162.69	0.88	129.33	1.43	222.40
3	0.97	157.56	0.87	127.99	1.33	209.56
4	0.94	148.59	0.81	126.17	1.32	197.55
5	0.91	145.38	0.72	123.32	1.31	196.59

Figure 5. Top five performances and shapes among airfoils shown in Fig. 4. We see MO-PaDGAN samples have significantly higher performance than GAN.

Figure 4 visualizes the joint distribution of C_L and C_L/C_D for randomly sampled airfoils. It shows that MO-PaDGAN generates airfoils with performances exceed randomly sampled airfoils from training data and the vanilla GAN (*i.e.*, the non-dominated Pareto set of generated samples is pushed further in the performance space to have higher values). Figures 3 and 4 indicate that MO-PaDGAN can expand the existing boundary of the design space towards high-performance regions outside the training data. This directed expansion is allowed since we provide the quality gradients (*i.e.*, $dq(\mathbf{x})/d\mathbf{x}$) information to MO-PaDGAN. Figure 5 further demonstrates that the top airfoils generated by MO-PaDGAN have much higher performances than those from data and the vanilla GAN (*i.e.*, the performances of the top five airfoils generated by MO-PaDGAN dominates those from training data and the vanilla GAN).

5. Conclusion

We proposed MO-PaDGAN with a new loss function based on Determinantal Point Processes. This model is useful when we want to explore different high-performance design alternatives or discover novel solutions. For example, when performing design optimization, one may accelerate the search for global optimal solutions by sampling start points from the proposed model. It can also be a tool in the early conceptual design stage to aid the creative process. The proposed framework also generalizes to other generative models like VAEs and can be used for various synthesis problems like 3D shape generation and molecule discovery.

References

Ahmed, F., Deb, K., and Bhattacharya, B. Structural topology optimization using multi-objective genetic algorithm with constructive solid geometry representation. *Applied Soft Computing*, 39:240–250, 2016.

Bendsoe, M. P. and Sigmund, O. *Topology Optimization: Theory, Methods and Applications*. Springer, February 2004. ISBN 9783540429920.

Chen, W. and Fuge, M. Béziergan: Automatic generation of smooth curves from interpretable low-dimensional parameters. *arXiv preprint arXiv:1808.08871*, 2018.

Chen, W. and Fuge, M. Synthesizing designs with interpart dependencies using hierarchical generative adversarial networks. *Journal of Mechanical Design*, 141(11), 2019.

Chen, W., Fuge, M., and Chazan, J. Design manifolds capture the intrinsic complexity and dimension of design spaces. *Journal of Mechanical Design*, 139(5), 2017.

Chen, W., Chiu, K., and Fuge, M. Aerodynamic design optimization and shape exploration using generative adversarial networks. In *AIAA SciTech Forum*, San Diego, USA, Jan 2019. AIAA.

Drela, M. Xfoil: An analysis and design system for low reynolds number airfoils. In *Low Reynolds number aerodynamics*, pp. 1–12. Springer, 1989.

Elfeki, M., Couprie, C., Riviere, M., and Elhoseiny, M. Gdpp: Learning diverse generations using determinantal point processes. In *International Conference on Machine Learning*, pp. 1774–1783, 2019.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kulesza, A. and Taskar, B. Determinantal point processes for machine learning. *arXiv preprint arXiv:1207.6083*, 2012.

Park, K. and Lee, J. Optimal design of two-dimensional wings in ground effect using multi-objective genetic algorithm. *Ocean Engineering*, 37(10):902–912, 2010.

Shu, D., Cunningham, J., Stump, G., Miller, S. W., Yukish, M. A., Simpson, T. W., and Tucker, C. S. 3d design using generative adversarial networks and physics-based validation. *Journal of Mechanical Design*, 142(7), 2020.