Generate Semantically Similar Images with Kernel Mean Matching

Wittawat Jitkrittum[1*] Patsorn Sangkloy[2*] Muhammad Waleed Gondal [1] Amit Raj[2]
James Hays[2] Bernhard Schölkopf[1]

[1]Max Planck Institute for Intelligent Systems [2]Georgia Institute of Technology

We propose a novel procedure which adds "content-addressability" to any given unconditional implicit model e.g., a generative adversarial network (GAN). The procedure allows users to control the generative process by specifying a set (arbitrary size) of desired examples based on which similar samples are generated from the model. The proposed approach, based on kernel mean matching, is applicable to any generative models which transform latent vectors to samples, and does not require retraining of the model. Experiments on various high-dimensional image generation problems (CelebA-HQ, LSUN bedroom, bridge, tower) show that our approach is able to generate images which are consistent with the input set, while retaining the image quality of the original model. To our knowledge, this is the first work that attempts to construct, *at test time*, a content-addressable generative model from a trained marginal model.

1. Background

We first briefly review maximum mean discrepancy [2], and the kernel mean matching problem [1]. Our proposed method (Section 2) will be based on the kernel mean matching.

Maximum Mean Discrepancy (MMD) Given two distributions P, Q (on images), and a positive definite kernel $K(\boldsymbol{x}, \boldsymbol{y})$ where $\boldsymbol{x}$ and $\boldsymbol{y}$ are two images, MMD [2] defines a distance between P and Q, and can be written as $\mathrm{MMD}^2(P, Q) =$

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'} K(\boldsymbol{x}, \boldsymbol{x}') + \mathbb{E}_{\boldsymbol{y},\boldsymbol{y}'} K(\boldsymbol{y}, \boldsymbol{y}') - 2\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} K(\boldsymbol{x}, \boldsymbol{y}),$$

where $\boldsymbol{x}, \boldsymbol{x}' \overset{i.i.d.}{\sim} P$ and $\boldsymbol{y}, \boldsymbol{y}' \overset{i.i.d.}{\sim} Q$. MMD has been successfully applied in many problems such as two-sample testing and training generative adversarial networks. It can be shown that MMD is equivalent to the distance between two points in a reproducing kernel Hilbert space (induced by the kernel K) that represent the two distributions P, Q; these two points are known as mean embeddings (or **mean features**) of P and Q.

Given two independent samples $X_m := \{\boldsymbol{x}_i\}_{i=1}^m \overset{i.i.d.}{\sim} P$ and $Y_n := \{\boldsymbol{y}_i\}_{i=1}^n \overset{i.i.d.}{\sim} Q$, a plug-in estimator of MMD^2 is given by

$$\frac{1}{m^2}\sum_{i,j=1}^{m} K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} K(\boldsymbol{y}_i, \boldsymbol{y}_j) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} K(\boldsymbol{x}_i, \boldsymbol{y}_j).$$

A more general form of the estimator is given by

$$\sum_{i,j=1}^{m} w_i w_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} K(\boldsymbol{y}_i, \boldsymbol{y}_j) - \frac{2}{n}\sum_{i=1}^{m} w_i \sum_{j=1}^{n} K(\boldsymbol{x}_i, \boldsymbol{y}_j).$$
$$=: \widehat{\mathrm{MMD}}^2(X_m, Y_n, \boldsymbol{w}), \tag{1}$$

where we have introduced weights $\boldsymbol{w} := (w_1, \ldots, w_m)$ on the m points in X_m such that $w_i \in [0, 1]$ for all $i = 1, \ldots, m$, and $\sum_{i=1}^m w_i = 1$. The weighted form in (1) will be useful in our task for controlling the amount of contribution from each of the input images X_m to the generated images Y_n.

Kernel Mean Matching Given an input set of images X_m and weights $\boldsymbol{w}$, kernel mean matching [1] aims to find a set of points $Y_n := \{\boldsymbol{y}_i\}_{i=1}^n$ so as to minimize the MMD. Mathematically, $Y_n^* = \arg\min_{\{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_n\}} \widehat{\mathrm{MMD}}^2(X_m, Y_n, \boldsymbol{w})$. By interpreting K as a similarity function on images, this formulation yields diverse output images Y_n which are similar to the input samples (in the sense that the two underlying distributions are close).

2. Content-Addressable Image Generation

In this section, we detail our proposed procedure that enables any implicit generative models to perform content-based image generation. Let $\boldsymbol{z}$ be a latent random vector (code) of an implicit generative model g such that $\boldsymbol{y} = g(\boldsymbol{z})$ is a sample drawn from the model, where $\boldsymbol{z} \sim p_z$ and p_z is a fixed prior distribution defined on a domain $\mathcal{Z}$. Given a trained model $g\colon \boldsymbol{z} \mapsto \boldsymbol{x}$, a kernel K (discussed in Section

*Equal contribution. Corresponding authors: Wittawat Jitkrittum (wittawat@tuebingen.mpg.de) and Patsorn Sangkloy (patsorn.sangkloy@gmail.com.

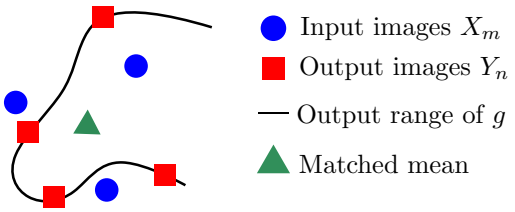


Figure 1: Given input images (blue circles), our approach generates images (red squares) from the model g so as to match the mean feature (green triangle) of the input images represented in a reproducing kernel Hilbert space. The input images do not need to be in the range of g.

1), and a set of input points $X_m = \{\boldsymbol{x}_i\}_{i=1}^m$ (content), we propose to generate new samples Y_n, conditioned on X_m, by solving the following optimization problem:

$$\min_{Z_n := \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}} \widehat{\mathrm{MMD}}^2(X_m, \{g(\boldsymbol{z}_i)\}_{i=1}^n, \boldsymbol{w}) \text{ s.t. } \forall i, \boldsymbol{z}_i \in \mathcal{Z}. \tag{2}$$

An illustration of our approach is presented in Figure 1.

Our approach relies on a positive definite kernel K to specify similarity between two images. It characterizes features of the input images that determine the output images. We propose using a kernel K which takes the form: $K(\boldsymbol{x}, \boldsymbol{y}) := k(E(\boldsymbol{x}), E(\boldsymbol{y}))$, where E is a pre-trained image feature extractor, and k is a simple, nonlinear kernel (e.g., an IMQ kernel) on top of the extracted features. In experiments, we use the inverse multi-quadric (IMQ) kernel $k(\boldsymbol{a}, \boldsymbol{b}) = (c^2 + \|\boldsymbol{a} - \boldsymbol{b}\|_2^2)^{-1/2}$ for some $c > 0$. We empirically observe that this choice yields realistic output images relevant to the input. To solve (2), we use Adam [5] which relies on the gradient $\nabla_{Z_n} \widehat{\mathrm{MMD}}^2(X_m, \{g(\boldsymbol{z}_i)\}_{i=1}^n, \boldsymbol{w})$ to update Z_n and find a local minimum.

3. Experiments

In this section, we show that our approach is able to perform content-based image generation on many image datasets and GAN models. Code to reproduce all the results will be made available.

3.1. Content-Based Generation of Complex Scenes

We consider three categories of the LSUN dataset [8]: bedroom, bridge, tower, and use pretrained GAN models from [6] which were trained separately on training samples from each category. The models are based on DCGAN architecture with additional residual connections [3]. For content-based generation, we use the IMQ kernel with parameter $c = 100$ and set the extractor E to be the output of the layer before the last fully connected layer of a pretrained Places365-ResNet classification model [9]. This network was trained to classify 365 unique scenes (training set comprising ten million images), and is expected to be able to capture high-level visual features of complex scenes.

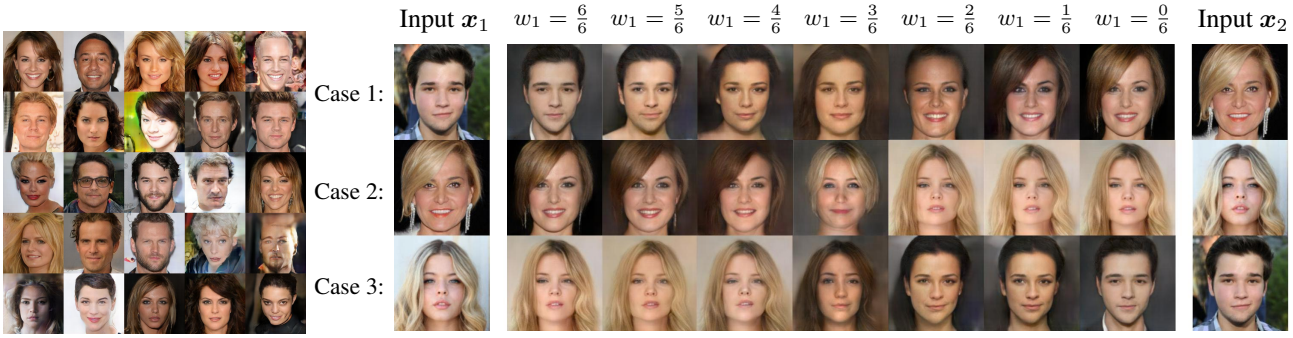
Our results in Figure 3 show that in each test case, the three generated images are highly consistent with the two input images (from the LSUN's test set). For instance, in bridge#1 (test case #1 of the LSUN-bridge category in Figure 3), not only is the tone black-and-white but the bridge structure is also well captured. In other cases such as tower#1, our procedure appears to generate similar buildings as present in the input images, but with a different viewing angle. This feat demonstrates that the proposed procedure can generate images that are *semantically similar* to the input.

3.2. Compression by Matching the Mean

An interesting use case of our formulation arises when $m > n$ (more input images than output images). In this case, the output mean features have fewer degrees of freedom than do the input mean features. As a result, for the two mean features to match, each output image is forced to combine visual features from multiple input images. For this reason, we refer to this task as the *compression task*. The simplest instance of this task is when $m = 2$ and $n = 1$. With $m = 2$ input images, there are two input weights: w_1 and w_2 such that $w_2 = 1 - w_1$. The weight $w_1 \in [0, 1]$ specifies the importance of the first input image $\boldsymbol{x}_1$ relative to the second.

To illustrate the compression, we use a GAN model from [6] pretrained on the CelebA-HQ problem [4]. Sample images from the model are shown in Figure 2a. We set the extractor E to be the output of layer Relu3-3 of the VGG-Face network [7]. The images generated from our procedure are shown in Figure 2b, where we consider three independent test cases, each defining a pair of input images $(\boldsymbol{x}_1, \boldsymbol{x}_2)$. We observe that when the weight w_1 is strictly between 0 and 1, the output images contain some visual features of the two input faces, and are consistent with both inputs.

It is worth noting that varying w_1 is not equivalent to linear interpolation between the latent vector that generates $\boldsymbol{x}_1$ and the latent vector that generates $\boldsymbol{x}_2$. In fact, there may not exist a latent vector $\boldsymbol{z}$ such that $g(\boldsymbol{z}) = \boldsymbol{x}$ for a given image $\boldsymbol{x}$. In our procedure, for each w_1, the obtained latent vector $\boldsymbol{z}_{w_1}$ is such that $g(\boldsymbol{z}_{w_1})$ is an output image whose feature vector well approximates the mean features defined by the input images X_m. Simply interpolating between two latent vectors may not give output images with this property.



(a) Unconditional samples

(b) Generated images given two inputs $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$

Figure 2: **(a)**: Unconditional samples from the GAN model studied in [6] (trained on the CelebA-HQ dataset). **(b)**: Generated images from the proposed procedure given two inputs $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ from three independent test cases. The weight w_1 specifies the emphasis on the input $\boldsymbol{x}_1$.

References

[1] Yutian Chen, Max Welling, and Alexander J. Smola. Super-samples from kernel herding. In *UAI*, 2010.

[2] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2017.

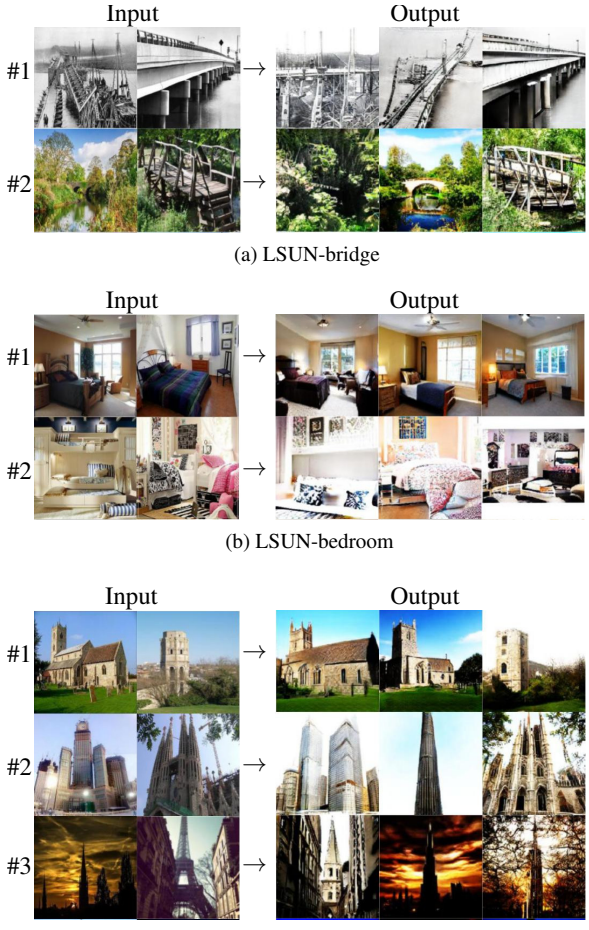
[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[6] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Which training methods for gans do actually converge? In *ICML*, 2018.

[7] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[8] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[9] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.



(a) LSUN-bridge

(b) LSUN-bedroom

(c) LSUN-tower

Figure 3: Generated output images from our approach. In each of the three LSUN categories, there are 2-3 test cases (denoted by #1, . . ., #3), each containing two input images from the LSUN test set.