Kernel Mean Matching for Content ADdressability of GANs

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

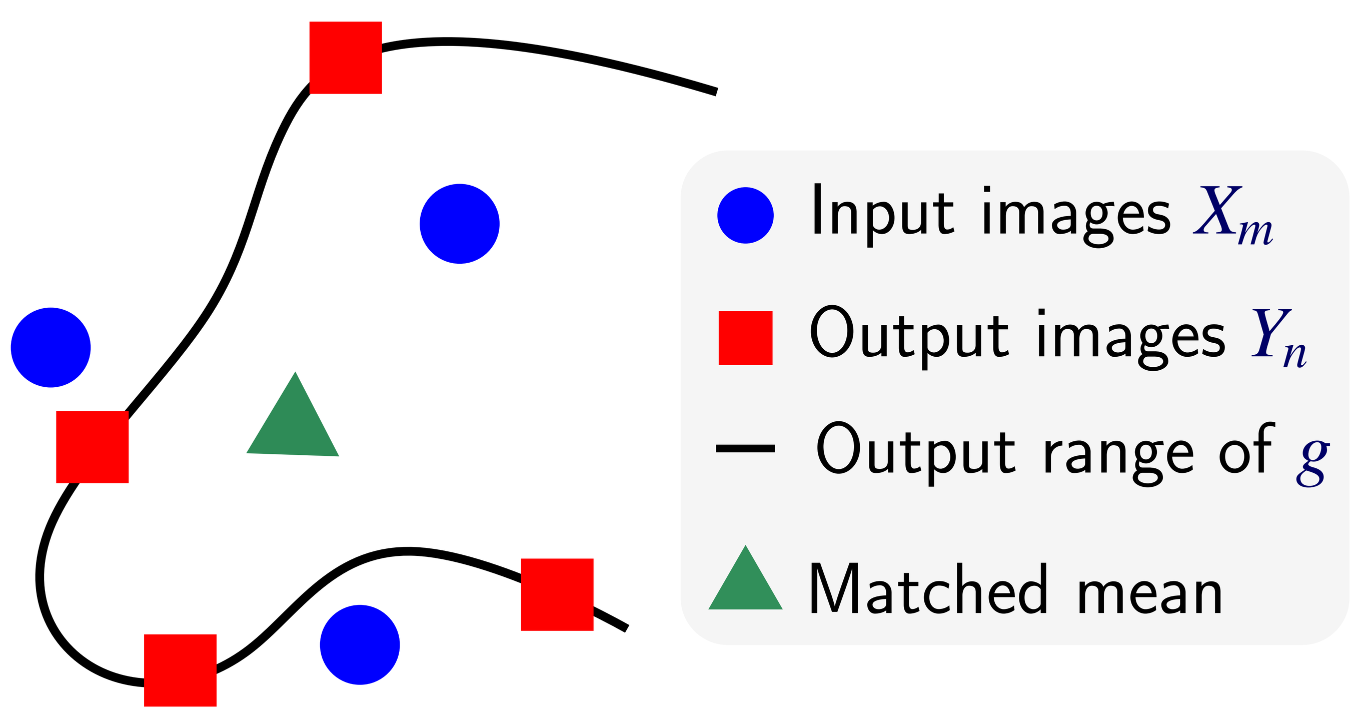
Summary

- **Given**: Pre-trained GAN g, input images $X_m := \{\mathbf{x}_i\}_{i=1}^m$
- **Goal**: Generate images $Y_n := \{\mathbf{y}_j\}_{j=1}^n$ similar to X_m.
- **Propose CADGAN**: a kernel mean matching procedure that adds "content-addressability" to g at run-time.
- **Advantages**:

1. No need to retrain g.
2. Flexible choice of the similarity criterion.
3. Fine-grained control with input weights $\{w_i\}_{i=1}^m$.

Proposal: CADGAN

CADGAN: *Generate* **images** *from g so as to match the* **mean feature** *of the* **input images** *represented in a reproducing kernel Hilbert space (RKHS).*



How: Minimize the distance (MMD) between the input and output means in RKHS $\mathcal{H}$ (kernel mean matching):

$$\arg\min_{\{\mathbf{y}_j\}_{j=1}^n \text{ in range}(g)} \left\| \sum_{i=1}^m w_i \phi(\mathbf{x}_i) - \frac{1}{n}\sum_{j=1}^n \phi(\mathbf{y}_j) \right\|_{\mathcal{H}}^2. \quad (1)$$

- $\{\mathbf{y}_j\}_{j=1}^n$ are constrained to be in the output range of g.
- ϕ: an implicit nonlinear function (induced by a kernel).
- w_i: weight of the input $\mathbf{x}_i$. $\sum_{i=1}^m w_i = 1$ and $w_i \in [0, 1]$.

Kernel Mean Matching with a Generator

- Let $K(\mathbf{a}, \mathbf{b}) = \langle \phi(\mathbf{a}), \phi(\mathbf{b}) \rangle_{\mathcal{H}}$ be a kernel ($\approx$ similarity) between two images $\mathbf{a}, \mathbf{b}$.
- Parametrize $\mathbf{y}_j = g(\mathbf{z}_j)$ where $\mathbf{z}_j$ is a latent vector.

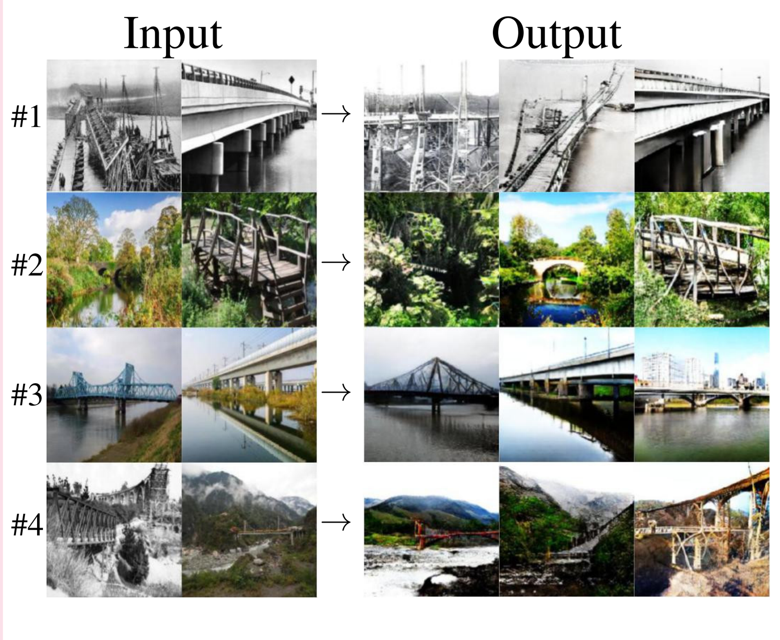
Then, (1) can be rewritten as

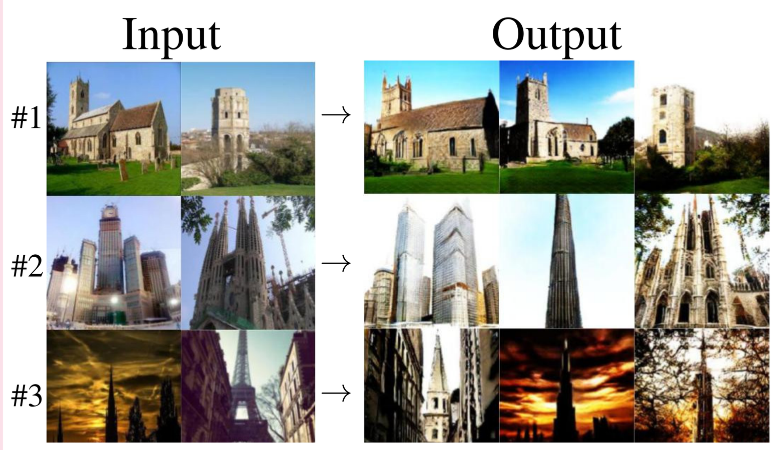
$$\sum_{i,j=1}^m w_i w_j K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{n^2}\sum_{i,j=1}^n K(g(\mathbf{z}_i), g(\mathbf{z}_j)) - \frac{2}{n}\sum_{i=1}^m w_i \sum_{j=1}^n K(\mathbf{x}_i, g(\mathbf{z}_j)). \quad (2)$$

Proposed **CADGAN**: $\arg\min_{\{\mathbf{z}_j\}_{j=1}^n} (2)$

- Optimize the latent vectors $\{\mathbf{z}_j\}_{j=1}^n$ with Adam.
- Output images: $\{g(\mathbf{z})_j\}_{j=1}^n$.
- Use kernel $K(\mathbf{a}, \mathbf{b}) := k(E(\mathbf{a}), E(\mathbf{b}))$ where E is an image feature extractor of choice e.g., VGG Face, Places365-ResNet.
- Use IMQ kernel $k(\mathbf{s}, \mathbf{t}) = (c^2 + \|\mathbf{s} - \mathbf{t}\|_2^2)^{-1/2}$ for some $c > 0$.

Experiment: LSUN-{Bridge, Bedroom, Tower}

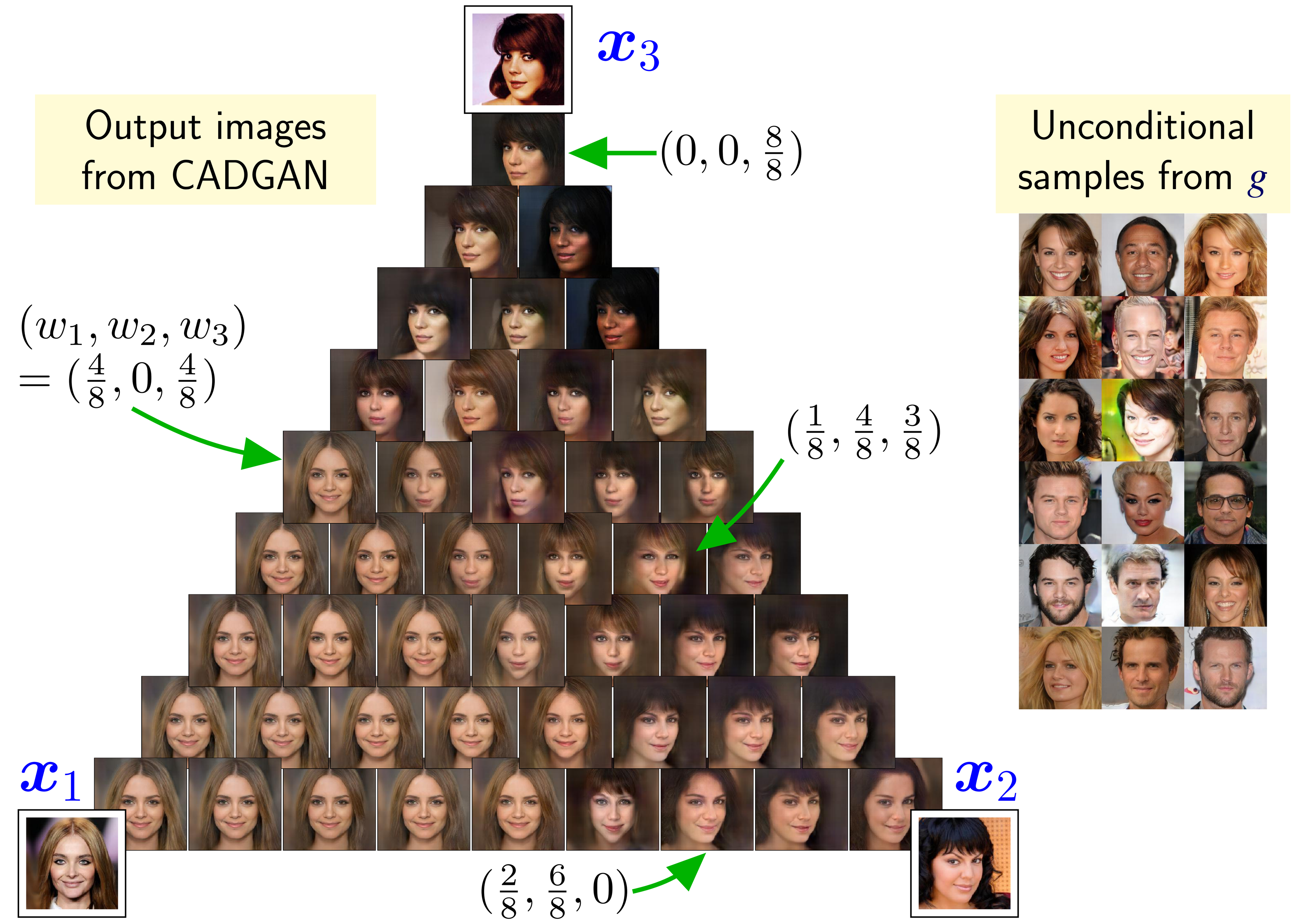




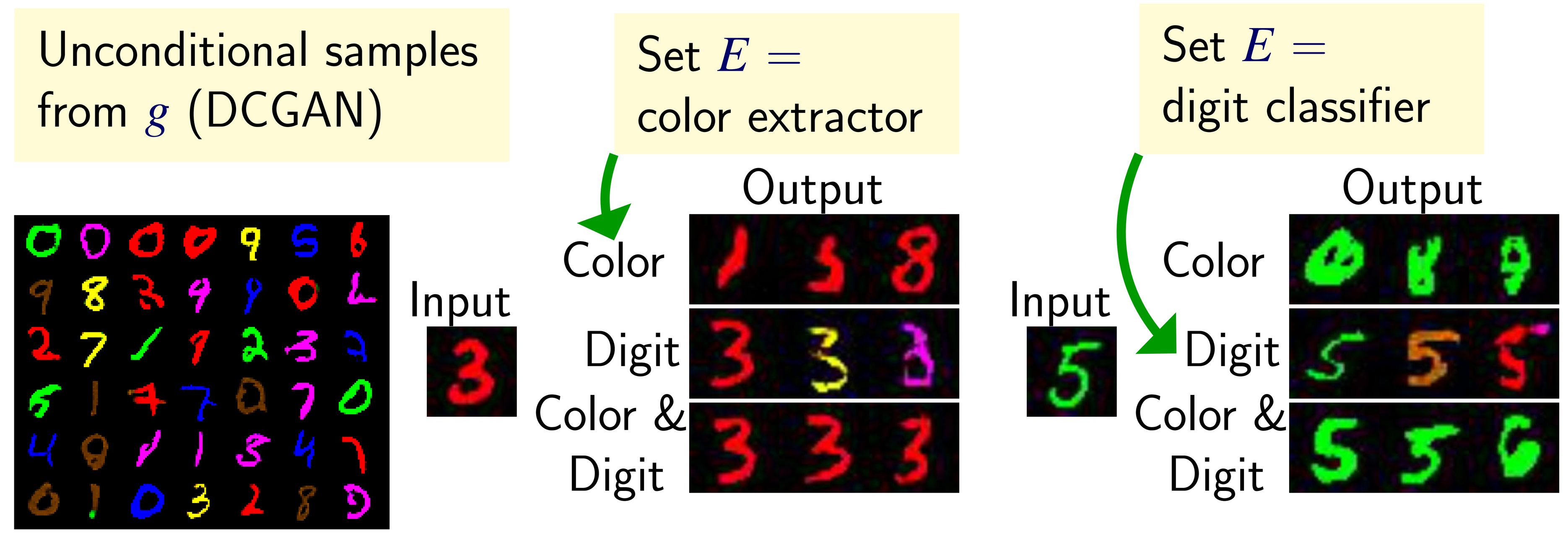
- 3 GANs from Mescheder et al., 2018 trained on LSUN-bridge, LSUN-bedroom, LSUN-tower.
- Extractor E = Places365-ResNet.

Experiment: CelebA-HQ

- g = GAN from Mescheder et al., 2018 trained on CelebA-HQ.
- For each (w_1, w_2, w_3), generate $n = 1$ image from $m = 3$ input images.



Experiment: Flexible Choice of Similarity Criterion



Aspects of the input image(s) that will be captured can be controlled by changing the extractor E.

* Wittawat Jitkrittum and Patsorn Sangkloy contributed equally.
Contact: wittawat@tuebingen.mpg.de, patsorn.sangkloy@gmail.com

Code: https://github.com/wittawatj/cadgan