Disentangling Mean Embeddings for Better Diagnostics of Image Generators

Sebastian G. Gruber
German Cancer Consortium (DKTK), partner site Frankfurt/Mainz,
a partnership between DKFZ and UCT Frankfurt-Marburg, Germany, Frankfurt am Main, Germany
German Cancer Research Center (DKFZ), Heidelberg, Germany
Goethe University Frankfurt, Germany
sebastian.gruber@dkfz.de

Pascal Tobias Ziegler
Goethe University Frankfurt, Germany

Florian Buettner
German Cancer Consortium (DKTK), partner site Frankfurt/Mainz,
a partnership between DKFZ and UCT Frankfurt-Marburg, Germany, Frankfurt am Main, Germany
German Cancer Research Center (DKFZ), Heidelberg, Germany
Frankfurt Cancer Institute (FCI), Germany
Goethe University Frankfurt, Germany

Abstract

The evaluation of image generators remains a challenge due to the limitations of traditional metrics in providing nuanced insights into specific image regions. This is a critical problem as not all regions of an image may be learned with similar ease. In this work, we propose a novel approach to disentangle the cosine similarity of mean embeddings into the product of cosine similarities for individual pixel clusters via central kernel alignment. Consequently, we can quantify the contribution of the cluster-wise performance to the overall image generation performance. We demonstrate how this enhances the explainability and the likelihood of identifying pixel regions of model misbehavior across various real-world use cases.

1 Introduction

The increasing prevalence of Artificial Intelligence (AI), particularly with the rise of sophisticated generative models like image generators, has brought about a transformative shift beyond the field of machine learning [Singh and Raza, 2021, Mirsky and Lee, 2021, Oppenlaender, 2022]. However, the evaluation of the outputs from these models, especially in the realm of image generation, continues to pose a significant challenge [Benny et al., 2021, Elasri et al., 2022, Xu et al., 2024]. Traditional evaluation metrics, like the maximum mean discrepancy (MMD) [Gretton et al., 2012], the Inception Score Criterion (ISC) [Salimans et al., 2016], the Fréchet Inception Distance (FID) [Heusel et al., 2017], or the Kernel Inception Distance (KID) [Bińkowski et al., 2018], fall short in providing a nuanced understanding of specific image regions, thereby limiting their effectiveness in assessing model performance comprehensively. The MMD is the squared distance between the mean embeddings of two distributions. A mean embedding is the expectation of a kernel-induced feature map based on the respective distribution. As we will see, we can decompose, which we refer to as *disentangle* based on [Vedral, 2002], mean embeddings as a tensor product under certain conditions. Depending on the kernel choice, the MMD can be used without an external model, but it cannot be

Interpretable AI: Past, Present and Future Workshop at NeurIPS 2024

decomposed in a meaningful way even when disentangled mean embeddings exist. This disallows more fine-grained interpretations of model performance by current evaluation approaches. In the context of this work, interpretation, interpretability, and explainability refer to performance and error assignment to different image regions, which increases human oversight and understanding, contrary to "black box" evaluation approaches only assessing entire images [Castelvecchi, 2016, Phillips et al., 2021, Longo et al., 2024].

In the present work, we **contribute** a novel approach based on disentangling the mean embedding of an image space into mean embeddings of independent clusters of pixels. Further, we show that the cosine similarity of the mean embeddings can be disentangled into the product of the cosine similarities for each respective cluster. This enables the evaluation and interpretation of the model performance of each cluster in isolation, significantly enhancing diagnostics of image generation and the likelihood of identifying the source pixel region of model misbehavior. We illustrate the gain in interpretability by monitoring the generalization performance of DCGAN [Radford et al., 2015] and DDPM [Ho et al., 2020] architectures trained on CelebA [Liu et al., 2015] and ChestMNIST [Yang et al., 2021] datasets.

2 Preliminaries on Mean Embeddings

In this section, we introduce the necessary background for mean embeddings and central kernel alignment. Given a nonempty set $\mathcal{X}$ and a positive semi-definite (p.s.d.) kernel $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there exists ϕ and an inner product $\langle .,. \rangle$ such that $k(x,y) = \langle \phi(x), \phi(y) \rangle$ [Schölkopf and Smola, 2002]. The associated RKHS is defined as the completion $\mathcal{H} = \overline{\operatorname{span}\{\phi(x) \mid x \in \mathcal{X}\}}$, where $\langle .,. \rangle_{\mathcal{H}} \coloneqq \langle .,. \rangle$ and $\|h\|_{\mathcal{H}} \coloneqq \sqrt{\langle h, h \rangle_{\mathcal{H}}}$ for $h \in \mathcal{H}$. For a distribution P with support $\mathcal{X}$ the **mean embedding** in $\mathcal{H}$ is defined via

$$\mu_P \coloneqq \mathbb{E}_{X \sim P}[\phi(X)]. \tag{1}$$

Given another distribution Q with similar support, Gretton et al. [2012] introduce the **maximum mean discrepancy** as the squared distance between mean embeddings μ_P and μ_Q defined by

$$\mathrm{MMD}_k^2(P,Q) \coloneqq \|\mu_P - \mu_Q\|_{\mathcal{H}}^2. \tag{2}$$

If k is a characteristic kernel, i.e. $\mu_{(.)}$ is an injective function for a set of distributions including P and Q, then it holds $P = Q \iff \mathrm{MMD}_k^2(P,Q) = 0$ [Gretton et al., 2012]. However, for the central research question of this paper, we will discover that the MMD cannot be disentangled into the MMD values of different input regions. Fortunately, there also exist other approaches to quantify the similarity of vectors μ_P and μ_Q in the RKHS $\mathcal{H}$. One of the most classical metrics is the cosine similarity defined between vectors $v, w \in \mathbb{R}^d$ via $\cos(v,w) = \frac{\langle v,w \rangle}{\|v\|\|w\|}$. Compared to the squared distance, it has the benefit of being easier to interpret as it lies within $[-1,1]$ with $v = w \implies \cos(v,w) = 1$. For general RKHS, the cosine similarity was already studied implicitly as the inner product of quantum mean embeddings in [Kübler et al., 2019]. To receive an analogous definition to the MMD, we define the cosine similarity between the mean embeddings μ_P and μ_Q as the **cosine mean similarity** given by

$$\mathrm{CMS}_k(P,Q) \coloneqq \frac{\langle \mu_P, \mu_Q \rangle_{\mathcal{H}}}{\|\mu_P\|_{\mathcal{H}} \|\mu_Q\|_{\mathcal{H}}}. \tag{3}$$

While the MMD can be seen as the generalization of the squared distance to possibly infinite dimensional RKHS, the CMS is the analogous generalization of the cosine similarity. If k is a c_0 universal kernel and P, Q are Borel probability measures, then it holds $P = Q \iff \mathrm{CMS}_k(P,Q) = 1$ [Kübler et al., 2019]. If k is c_0 universal, then it follows that it is also characteristic, but the opposite does not always hold [Sriperumbudur et al., 2011]. Consequently, CMS and MMD only share the uniqueness of their optimum for c_0 universal kernels. Examples of c_0 universal kernels are the RBF kernel $k_{\mathrm{rbf}}(x,y) = \exp\left(-\gamma \|x-y\|_2^2\right)$, the Laplacian kernel, the Matérn kernel, or any other characteristic and translation invariant kernel [Sriperumbudur et al., 2011]. Given datasets $\mathbf{X} = (X_1, \ldots, X_n) \overset{iid}{\sim} P$ and $\mathbf{Y} = (Y_1, \ldots, Y_m) \overset{iid}{\sim} Q$ of independently and identically distributed (iid) random variables, an empirical estimator for $\mathrm{CMS}_k(P,Q)$ can be defined via

$$\widehat{\mathrm{CMS}}_k(\mathbf{X}, \mathbf{Y}) \coloneqq \frac{\sum_{i=1}^n \sum_{j=1}^m k(X_i, Y_j)}{\sqrt{\sum_{i=1}^n \sum_{j=1}^n k(X_i, X_j)} \sqrt{\sum_{i=1}^m \sum_{j=1}^m k(Y_i, Y_j)}}. \tag{4}$$

Using kernels with images in practice usually follows either of two approaches: Either the image is flattened and each pixel is an entry in the vector provided as argument for the kernel [Schölkopf, 1997, Gruber and Buettner, 2024], or the image is encoded into a smaller-dimensional semantic vector space, which is then the argument for the kernel [Bińkowski et al., 2018]. While the latter approach gained more prominence in recent years [Benny et al., 2021, Xu et al., 2024], the encoding is usually learned by a neural network and not interpretable. In the following of this work, we focus on the former approach, which allows to disentangle the image provided that the chosen kernel is a product of pixel-wise kernels. Specifically, we assume that for flattened images $x = (x_1, \ldots, x_d)^\intercal \in \mathbb{R}^d$ and $y = (y_1, \ldots, y_d) \in \mathbb{R}^d$ with d pixels the image-wise kernel $k_{\text{img}} \colon \mathcal{X}_{\text{img}} \times \mathcal{X}_{\text{img}} \to \mathbb{R}$ with $\mathcal{X}_{\text{img}} \subseteq \mathbb{R}^d$ can be decomposed into a product $k_{\text{img}}(x, y) = k_{\text{pxl}}^{\otimes d}((x_1, \ldots, x_d), (y_1, \ldots, y_d)) = k_{\text{pxl}}(x_1, y_1) \cdots k_{\text{pxl}}(x_d, y_d)$ for a pixel-wise kernel $k_{\text{pxl}} \colon \mathcal{X}_{\text{pxl}} \times \mathcal{X}_{\text{pxl}} \to \mathbb{R}$ with $\mathcal{X}_{\text{pxl}} \subseteq \mathbb{R}$. Note that this assumes grey-scale images for simplicity, however colored images (with three color channels) can be easily represented by assuming $\mathcal{X}_{\text{img}} \subseteq \mathbb{R}^{3d}$ and $\mathcal{X}_{\text{pxl}} \subseteq \mathbb{R}^3$. The RBF and Laplacian kernels are such product kernels, since we can write $k_{\text{rbf}}(x, y) = \exp\left(-\gamma (x_1 - y_1)^2\right) \cdots \exp\left(-\gamma (x_d - y_d)^2\right)$. In general, these are special cases of product kernels discussed in [Szabó and Sriperumbudur, 2018]. The image-wise RKHS $\mathcal{H}_{\text{img}}$ and feature map $\phi_{\text{img}} \colon \mathcal{X}_{\text{img}} \to \mathcal{H}_{\text{img}}$ associated with k_{img} can also be decomposed into a tensor product space $\mathcal{H}_{\text{img}} = \underbrace{\mathcal{H}_{\text{pxl}} \otimes \cdots \otimes \mathcal{H}_{\text{pxl}}}_{d \text{ times}}$ of the pixel-wise RKHS $\mathcal{H}_{\text{pxl}}$ and a tensor product

$$\phi_{\text{img}}(x) = (\phi_{\text{pxl}} \otimes \cdots \otimes \phi_{\text{pxl}})(x_1, \ldots, x_d) = \bigotimes_{i=1}^{d} \phi_{\text{pxl}}(x_i) \tag{5}$$

of the pixel-wise feature maps $\phi_{\text{pxl}} \colon \mathcal{X}_{\text{pxl}} \to \mathcal{H}_{\text{pxl}}$ associated with k_{pxl} [Szabó and Sriperumbudur, 2018]. However, when we compute the respective mean embedding for a distribution P_{img} of a random image $X = (X_1, \ldots, X_d)$, then we cannot decompose it in general into pixel-wise mean embeddings since

$$\mu_{P_{\text{img}}} = \mathbb{E}_{X \sim P_{\text{img}}}[\phi_{\text{img}}(X)] = \mathbb{E}_{X \sim P_{\text{img}}}\left[\bigotimes_{i=1}^{d} \phi_{\text{pxl}}(X_i)\right] \neq \bigotimes_{i=1}^{d} \mathbb{E}_{X_i \sim P_i}[\phi_{\text{pxl}}(X_i)] \tag{6}$$

where P_i are the marginal distributions of pixel indices $i = 1 \ldots d$. Such a pixel-wise decomposition, which we also refer to as disentanglement, is usually not possible in practice since pixels are correlated. However, for interpretability purposes, we do not necessarily require a complete disentanglement of all individual pixels, but it may suffice to discover disentangled clusters of pixels. To quantify to what degree this is possible in practice, we require the following.

The cross-covariance matrix between an $\mathbb{R}^d$-valued random variable X and an $\mathbb{R}^{d'}$-valued random variable Y is defined by

$$\operatorname{Cov}(X, Y) \coloneqq \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])^\intercal] \in \mathbb{R}^{d \times d'}. \tag{7}$$

Any matrix in $\mathbb{R}^{d \times d'}$ may also be seen as a linear operator from $\mathbb{R}^{d'} \to \mathbb{R}^d$. Consequently, given another p.s.d. kernel k' with RKHS $\mathcal{H}'$, a generalization of the cross-covariance matrix to $\mathcal{H}$-valued and $\mathcal{H}'$-valued random variables $\phi(X)$ and $\phi'(Y)$ is given via the cross-covariance operator $\mathcal{C}_{XY} \colon \mathcal{H}' \to \mathcal{H}$ with

$$\mathcal{C}_{XY} \coloneqq \mathbb{E}[(\phi(X) - \mathbb{E}[\phi(X)]) \otimes (\phi'(Y) - \mathbb{E}[\phi'(Y)])]. \tag{8}$$

The Hilbert-Schmidt norm of an operator $\mathcal{C} \colon \mathcal{H}' \to \mathcal{H}$ of Hilbert spaces $\mathcal{H}$ and $\mathcal{H}'$ with orthonormal bases $a_1, a_2, \ldots$ and $b_1, b_2, \ldots$ is defined via $\|\mathcal{C}\|_{\text{HS}} = \sum_{ij} \langle a_i, \mathcal{C} b_j \rangle_{\mathcal{H}}$ [Gretton et al., 2005]. It reduces to the Frobenius norm if both spaces are finite-dimensional Euclidean spaces. For $g \in \mathcal{H}$ and $h \in \mathcal{H}'$ it holds $\|g \otimes h\|_{\text{HS}} = \|g\|_{\mathcal{H}} \|h\|_{\mathcal{H}'}$, from which follows that

$$\begin{aligned} \|\mathcal{C}_{XY}\|_{\text{HS}}^2 = {} & \mathbb{E}_{X,Y,X^{\text{c}},Y^{\text{c}}}[k(X, X^{\text{c}}) k'(Y, Y^{\text{c}})] - \mathbb{E}_{X,X^{\text{c}},Y^{\text{c}}}[k(X, X^{\text{c}}) \mathbb{E}_Y[k'(Y, Y^{\text{c}})]] \\ & - \mathbb{E}_{Y,X^{\text{c}},Y^{\text{c}}}[\mathbb{E}_X[k(X, X^{\text{c}})] k'(Y, Y^{\text{c}})] + \mathbb{E}_{X,X^{\text{c}}}[k(X, X^{\text{c}})] \mathbb{E}_{Y,Y^{\text{c}}}[k'(Y, Y^{\text{c}})], \end{aligned} \tag{9}$$

where $(X^{\text{c}}, Y^{\text{c}})$ is an i.i.d. copy of (X, Y) [Gretton et al., 2005]. Then, Gretton et al. [2005] show that it holds

$$\|\mathcal{C}_{XY}\|_{\text{HS}} = 0 \iff \mathbb{E}[\phi(X) \otimes \phi'(Y)] = \mu_{\mathbb{P}_X} \otimes \mu'_{\mathbb{P}_Y} \tag{10}$$

with $\mu'_{\mathbb{P}_Y} = \mathbb{E}[\phi'(Y)]$. In consequence, they refer to $\mathrm{HSIC}_{k,k'}(\mathbb{P}_{XY}) \coloneqq \|\mathcal{C}_{XY}\|^2_{\mathrm{HS}}$ as **Hilbert-Schmidt independence criterion** (HSIC). Equation 9 indicates that we can estimate the HSIC in practice via the kernel trick, even when $\mathcal{H}$ or $\mathcal{H}'$ are infinite-dimensional. For two sets of samples $\mathbf{X} \coloneqq \{X_1, \ldots, X_n\}$ and $\mathbf{Y} \coloneqq \{Y_1, \ldots, Y_n\}$ with i.i.d. $(X_1, Y_1), \ldots, (X_n, Y_n) \sim \mathbb{P}_{XY}$ an estimator is given by

$$\mathrm{HSIC}_{k,k'}(\mathbf{X}, \mathbf{Y}) \coloneqq \mathrm{tr}\left(\mathbf{K}_X \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^\intercal\right)\mathbf{K}_Y \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^\intercal\right)\right), \tag{11}$$

where $[\mathbf{K}_X]_{ij} \coloneqq k(X_i, X_j)$, $[\mathbf{K}_Y]_{ij} \coloneqq k'(Y_i, Y_j)$, $\mathbf{1} = (1, \ldots, 1)^\intercal$ is the vector of 1's, and I is the unit matrix [Cortes et al., 2012].
However, when we evaluate HSIC in practice, the estimated values will never be precisely zero. This makes comparing HSIC values across different random variables problematic since their ranges may have different magnitudes. Consequently, we use the normalized version of HSIC referred to as **central kernel alignment** (CKA) [Cortes et al., 2012, Chang et al., 2013], which is defined by

$$\mathrm{CKA}_{k,k'}(\mathbb{P}_{XY}) \coloneqq \frac{\|\mathcal{C}_{XY}\|^2_{\mathrm{HS}}}{\|\mathcal{C}_{XX}\|_{\mathrm{HS}} \|\mathcal{C}_{YY}\|_{\mathrm{HS}}}. \tag{12}$$

It has the form of a squared correlation coefficient and by the Cauchy-Schwartz inequality lies within $[0, 1]$ [Chang et al., 2013]. Even though $\mathrm{CKA}_{k,k'}$ and CMS_k may appear similar in form, they measure very different things: While $\mathrm{CKA}_{k,k'}$ measures the independence between random variables according to their joint distribution, CMS_k compares how similar their marginal distributions are via their mean embedding. For two sets of samples $\mathbf{X} \coloneqq (X_1, \ldots, X_n)$ and $\mathbf{Y} \coloneqq (Y_1, \ldots, Y_n)$ with i.i.d. $(X_1, Y_1), \ldots, (X_n, Y_n) \sim \mathbb{P}_{XY}$ an estimator for $\mathrm{CKA}_{k,k'}(\mathbb{P}_{XY})$ is given by re-using the HSIC estimator via

$$\widehat{\mathrm{CKA}}_{k,k'}(\mathbf{X}, \mathbf{Y}) \coloneqq \frac{\mathrm{HSIC}_{k,k'}(\mathbf{X}, \mathbf{Y})}{\sqrt{\mathrm{HSIC}_{k,k'}(\mathbf{X}, \mathbf{X})\,\mathrm{HSIC}_{k,k'}(\mathbf{Y}, \mathbf{Y})}}. \tag{13}$$

Similar to the HSIC, it holds

$$\mathrm{CKA}_{k,k'}(\mathbb{P}_{XY}) = 0 \iff \mathbb{E}_{X,Y\sim\mathbb{P}_{XY}}[\phi(X) \otimes \phi'(Y)] = \mu_{\mathbb{P}_X} \otimes \mu'_{\mathbb{P}_Y}. \tag{14}$$

In the next section, we state our theoretical main contribution, which uses the CKA to disentangle the CMS value of an entire target domain into the product of CMS values in sub-domains.

3 Cosine Similarity of Disentangled Mean Embeddings

We can now state the main theoretical contribution of this work, which describes when we are allowed to disentangle the image-wise CMS into the product of more fine-grained cluster-wise CMS values.

Theorem 1. *Assume for random variables $X = (X_1, \ldots, X_d)^\intercal$ and $Y = (Y_1, \ldots, Y_d)^\intercal$ with outcomes in a space $\mathcal{X}^d$ and for a p.s.d. kernel $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there exists a partition $\mathbf{I}$ of the indices $\{1 \ldots d\}$ such that for all $I, I' \in \mathbf{I}$ it holds $\mathrm{CKA}_{k^{\otimes|I|}, k^{\otimes|I'|}}\left(\mathbb{P}_{X_I X_{I'}}\right) = 0 = \mathrm{CKA}_{k^{\otimes|I|}, k^{\otimes|I'|}}\left(\mathbb{P}_{Y_I Y_{I'}}\right)$ with $X_I \coloneqq (X_i)^\intercal_{i\in I}$ and $Y_{I'} \coloneqq (Y_i)^\intercal_{i\in I'}$. Then, we have*

$$\mathrm{CMS}_{k^{\otimes d}}(\mathbb{P}_X, \mathbb{P}_Y) = \prod_{I\in\mathbf{I}} \mathrm{CMS}_{k^{\otimes|I|}}(\mathbb{P}_{X_I}, \mathbb{P}_{Y_I}). \tag{15}$$

The proof located in Appendix D is mostly based on Equation 14. Theorem 1 indicates, that, by finding appropriate clusters, we can track the individual cluster-wise CMS values without losing information about the overall image-wise CMS. The cluster-wise CMS values then help to identify the cluster(s) responsible for certain behavior of the overall image-wise CMS, improving the interpretability of the model performance and the training dynamics. One practical constraint is that the estimated CKA values are likely not exactly zero since estimators cannot be expected to be perfectly precise and there may often be an infinitesimal correlation between pixels. This indicates that the assumptions of Theorem 1 will be, strictly speaking, violated to some degree in practice. However, it is straightforward to verify if the disentanglement is meaningful by comparing both sides of Equation 15, as we will see in Section 4.

Algorithm 1 Monitoring Cosine Similarity of Disentangled Mean Embeddings

Input: Training data $D^{\text{tr}} \in \mathbb{R}^{n_{\text{tr}} \times wh}$ and test data $D^{\text{te}} \in \mathbb{R}^{n_{\text{te}} \times wh}$ with n_{tr} and n_{te} number of flattened pixels with resolution $w \times h$, generator G_t with training iterations $t \in \mathbb{N}$ and n' image generations, p.s.d. kernel $k \colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$.
Initialize empty correlation matrix $M \in \mathbb{R}^{wh \times wh}$.
for $i = 1, \ldots, wh$ **do**
  for $j = 1, \ldots, wh$ **do**
    { Compute centered kernel alignment between pixel i and j}
    $M_{ij} \leftarrow \widehat{\text{CKA}}_{k,k} \left(\{D^{\text{tr}}_{li}\}_{l=1 \ldots n_{\text{tr}}}, \left\{ D^{\text{tr}}_{lj} \right\}_{l=1 \ldots n_{\text{tr}}} \right)$ according to Eq. 13
  end for
end for
$\{I_1, \ldots, I_C\} \leftarrow \text{HierarchicalClustering}\,(M)$
for $t = 1, \ldots$ **do**
  $G_{t+1} \leftarrow \text{TrainingIteration}\,(G_t)$
  $\hat{D} \leftarrow$ generate n' images with G_{t+1}
  ImageSimilarity $\leftarrow \widehat{\text{CMS}}_{k^{\otimes wh}} \left(\left\{ D^{\text{te}}_{ij} \right\}_{i=1 \ldots n_{\text{te}}, j=1 \ldots wh}, \{\hat{D}_{ij}\}_{i=1 \ldots n', j=1 \ldots wh} \right)$
  for I, c **in** enumerate $\{I_1, \ldots, I_C\}$ **do**
    ClusterSimilarity$_c \leftarrow \widehat{\text{CMS}}_{k^{\otimes |I|}} \left(\left\{ D^{\text{te}}_{ij} \right\}_{i=1 \ldots n_{\text{te}}, j \in I}, \{\hat{D}_{ij}\}_{i=1 \ldots n', j \in I} \right)$
  end for
  Output ImageSimilarity, $\{\text{ClusterSimilarity}_c\}_{c=1 \ldots C}$
end for

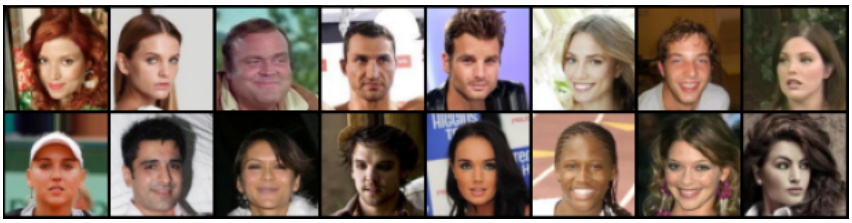
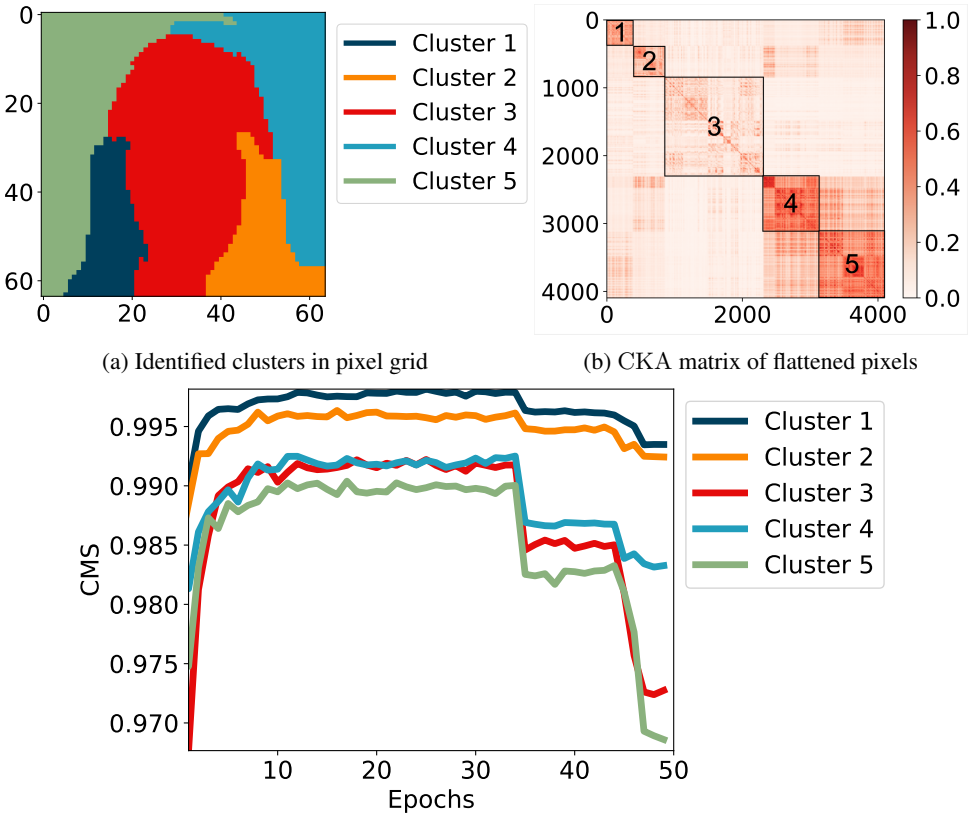


Figure 1: Samples of the CelebA dataset. Most faces are centered of similar size and similar angles. The clusters identified in Figure 2 match this observation.

If we want to turn Theorem 1 into a practical algorithm, we are facing a runtime problem: The number of possible partitions for a set grows exponentially with the set size [Berend and Tassa, 2010], which makes evaluating the CKA for all possible partitions of an image grid infeasible. As a workaround, we only compute the CKA between all pairwise pixels, which has $O\left(d^2\right)$ runtime complexity. We then perform hierarchical clustering to identify clusters of pixels with high pairwise CKA values. Further, we only compute the CKA values based on the training data as the generated images converge to this distribution during training. As we will see in the experiments, this simple approach works sufficiently well to find meaningful clusters. The whole algorithm to disentangle the CMS for monitoring the training performance of an image generator is presented in Algorithm 1, where we use the CMS estimator of Equation 4 and the CKA estimator of Equation 13.

4 Experiments

The source code for the following experiments is located at `https://github.com/MLO-lab/Disentangling_Mean_Embeddings`. We run experiments on the CelebA dataset [Liu et al., 2015], which consists of 200.000 colored celebrity images with resolution 64 x 64, and on the ChestMNIST dataset [Yang et al., 2021] consisting of 112.120 gray-scale chest scans with 28 x 28 resolution. Since both datasets show centered faces/chests at a similar angle, we can expect to identify various clusters of pixels that can be interpreted in a meaningful way (c.f. Figure 1 and Figure 6 for samples). For CelebA, we train 20 seeds of the DCGAN architecture [Radford et al., 2015] on randomly sampled 90% of the original set, and use the other 10% for evaluation. As errors, we consider the CMS and



(a) Identified clusters in pixel grid

(b) CKA matrix of flattened pixels

(c) CMS values throughout training

Figure 2: **Top-Left:** The identified clusters match how a human may separate the image structure of CelebA: There are two clusters for the background (Clusters 4 & 5), two clusters for long hair or alternating head angles (Clusters 1 & 2), and one central cluster for the head and neck (Cluster 3). **Top-Right:** The correlation matrix in terms of the CKA values indicates how well the clusters can be separated. The blocks on the diagonal are ordered by cluster number. As can be seen, most clusters are fairly independent of the other clusters (especially Cluster 3). Only clusters 4 and 5 show a relatively strong dependence on each other, which is expected since these often express the same background in the images (c.f. Figure 1). **Bottom:** Unlike the other errors, we can decompose the image-wise CMS into the CMS of different clusters according to the CKA. This offers novel insights into model performance. For example, we can detect that Cluster 3 and 5 degrade more in the training collapses than the other clusters.

MMD based on the RBF kernel with γ set to the inverse of the median of the pairwise Euclidean distances between training instances, which is a heuristic based on [Schölkopf and Smola, 2002]. Further, we evaluate the Inception Score Criterion (ISC) [Salimans et al., 2016], Fréchet Inception Distance (FID) [Heusel et al., 2017], and the Kernel Inception Distance (KID) [Bińkowski et al., 2018]. We average all errors across all seeds. More details are given in Appendix B.

In Figure 2, we show the identified clusters for CelebA on the left, and the respective pairwise CKA values of the (flattened) pixels on the right, which we refer to as CKA matrix. The indices in the CKA matrix are arranged according to the clusters. As can be seen, Cluster 3, which represents the face, is fairly independent of the other clusters. However, Cluster 4 and 5 share a lot of dependence, which is not surprising, since these represent the background. The training curves for the DCGAN architecture are depicted in Figure 3. There, on the left, we compare the image-wise CMS with the product of the cluster-wise CMS to verify that Theorem 1 holds approximately. On the right in Figure 3, we show the ISC, FID, KID, and MMD for comparison. All metrics capture similar trends in most cases. The benefits of our approach become apparent in Figure 2c, where we plot the cluster-wise CMS values for the detected clusters. Here, we can determine how much each cluster influences the image-wise CMS throughout training. Especially Cluster 3 and 5, which represent the

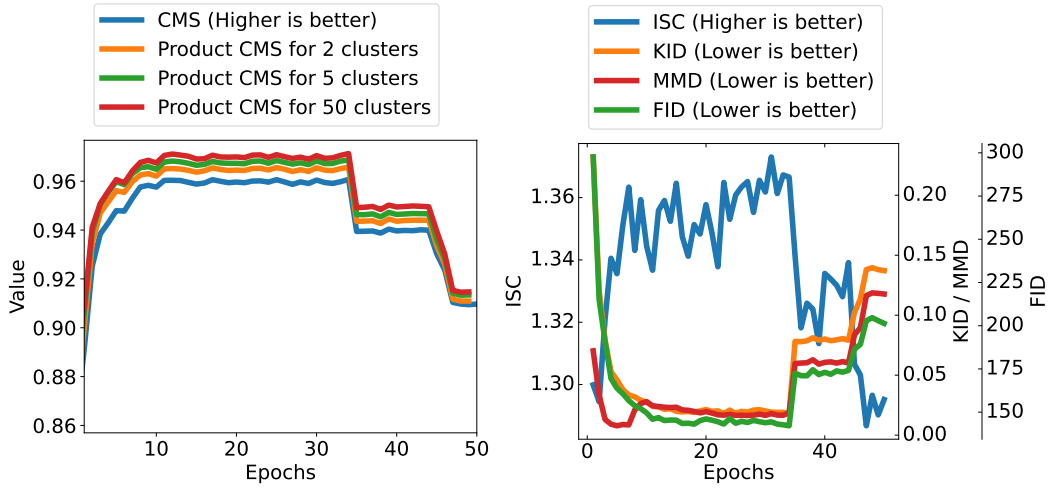


Figure 3: Different errors throughout the training of a DCGAN model on the CelebA dataset. All lines are an average of 20 seeds. **Left:** The CMS (higher is better) shows how the average training run improves until epoch 10. After epoch 30, some models collapse, and after epoch 45 additional collapses occur. Computing the product of the cluster-wise CMS values according to our methodology shows a close match with the normal CMS, indicating the correctness of the clusters. **Right:** The ISC does match the other errors but shows erratic behavior. The KID and FID resemble the CMS quite closely. The MMD also shows a similar trajectory as the other errors but indicates a minimum around epochs 5-7. Generated samples of the training runs match these observations (c.f. Figure 5).

head and left background area, are striking: They degrade the worst among all clusters after the two training collapses. This indicates, that these pixel regions have the highest influence on the worsening image-wise CMS.
In Appendix B, we discuss the results for ChestMNIST in more detail. Specifically, we compare the DCGAN with the DDPM architecture [Ho et al., 2020] in Figure 4. There, we discover how a major performance drop during the training runs with the DCGAN architecture can be assigned to the background of the images. On the contrary, the DDPM architecture fits all regions quickly except the background region, which requires further iterations.
Overall, such an analysis is only possible with our approach and the CMS as error, since the other errors (MMD, FID, KID, ISC) cannot be disentangled similarly.

Limitations. Our approach offers novel insights, but it assumes mean embeddings are a meaningful representation of the images based on a user-defined product kernel. While kernels scale well to higher dimensions, they will still degrade at some resolution [Gretton et al., 2012]. Further, computing the clusters is computationally expensive and we may not expect to find perfectly independent clusters in practice.

5 Conclusion

In this work, we introduced a novel approach to disentangle the mean embedding of an image space into mean embeddings of approximately independent pixel clusters. We also proved when the cosine similarity of the mean embeddings can be disentangled into the product of the cosine similarities for each respective cluster. This enables the evaluation and interpretation of the generalization performance of each cluster in isolation, significantly enhancing the explainability and the likelihood of identifying model misbehavior. We demonstrated the improved interpretability by monitoring the training of various architectures on the CelebA and ChestMNIST datasets according to the MMD, ISC, FID, KID, and our approach.

References

Yaniv Benny, Tomer Galanti, Sagie Benaim, and Lior Wolf. Evaluation metrics for conditional image generation. *International Journal of Computer Vision*, 129:1712–1731, 2021.

Daniel Berend and Tamir Tassa. Improved bounds on bell numbers and on moments of sums of random variables. *Probability and Mathematical Statistics*, 30(2):185–205, 2010.

Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018.

Davide Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.

Billy Chang, Uwe Kruger, Rafal Kustra, and Junping Zhang. Canonical correlation analysis based on hilbert-schmidt independence criterion and centered kernel target alignment. In *International Conference on Machine Learning*, pages 316–324, 2013.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13:795–828, 2012.

Mohamed Elasri, Omar Elharrouss, Somaya Al-Maadeed, and Hamid Tairi. Image generation: A review. *Neural Processing Letters*, 54(5):4609–4646, 2022.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International Conference on Algorithmic Learning Theory*, pages 63–77, 2005.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

Sebastian G. Gruber and Florian Buettner. A bias-variance-covariance decomposition of kernel scores for generative models. In *International Conference on Machine Learning*, pages 16460–16501, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Jonas M Kübler, Krikamol Muandet, and Bernhard Schölkopf. Quantum mean embedding of probability distributions. *Physical Review Research*, 1(3):033159, 2019.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

Luca Longo, Mario Brcic, Federico Cabitza, Jaesik Choi, Roberto Confalonieri, Javier Del Ser, Riccardo Guidotti, Yoichi Hayashi, Francisco Herrera, Andreas Holzinger, et al. Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Information Fusion*, 106:102301, 2024.

Yisroel Mirsky and Wenke Lee. The creation and detection of deepfakes: A survey. *ACM computing surveys (CSUR)*, 54(1):1–41, 2021.

Jonas Oppenlaender. The creativity of text-to-image generation. In *International Academic Mindtrek Conference*, pages 192–202, 2022.

P Jonathon Phillips, P Jonathon Phillips, Carina A Hahn, Peter C Fontana, Amy N Yates, Kristen Greene, David A Broniatowski, and Mark A Przybocki. Four principles of explainable artificial intelligence. 2021.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in Neural Information Processing Systems*, 29, 2016.

Bernhard Schölkopf. *Support vector learning*. PhD thesis, Oldenbourg München, Germany, 1997.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

Nripendra Kumar Singh and Khalid Raza. Medical image generation using generative adversarial networks: A review. *Health Informatics: A Computational Perspective in Healthcare*, pages 77–96, 2021.

Bharath K Sriperumbudur, Kenji Fukumizu, and Gert RG Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *Journal of Machine Learning Research*, 12(7), 2011.

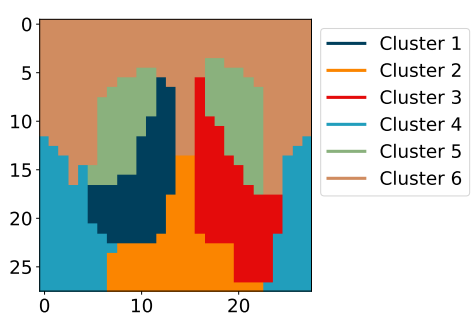
Zoltán Szabó and Bharath K Sriperumbudur. Characteristic and universal tensor product kernels. *Journal of Machine Learning Research*, 18(233):1–29, 2018.

Vlatko Vedral. The role of relative entropy in quantum information theory. *Reviews of Modern Physics*, 74(1):197, 2002.

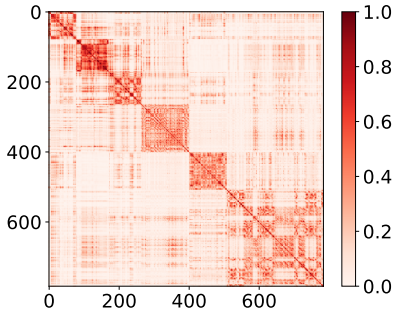
Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *International Symposium on Biomedical Imaging*, pages 191–195, 2021.

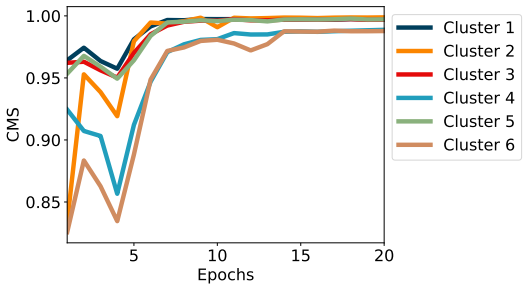
Wojciech Zaremba, Arthur Gretton, and Matthew Blaschko. B-test: A non-parametric, low variance kernel two-sample test. *Advances in Neural Information Processing Systems*, 26, 2013.



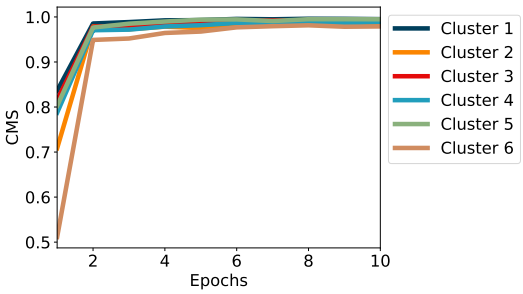
(a) Identified clusters in pixel grid



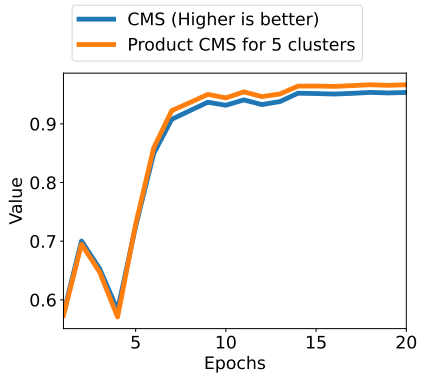
(b) CKA matrix of flattened pixels



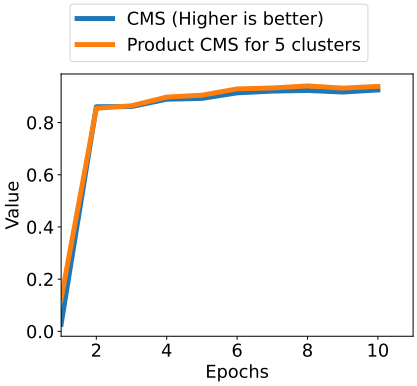
(c) CMS values throughout training (DCGAN)



(d) CMS values throughout training (DDPM)



(e) CMS values throughout training (DCGAN)

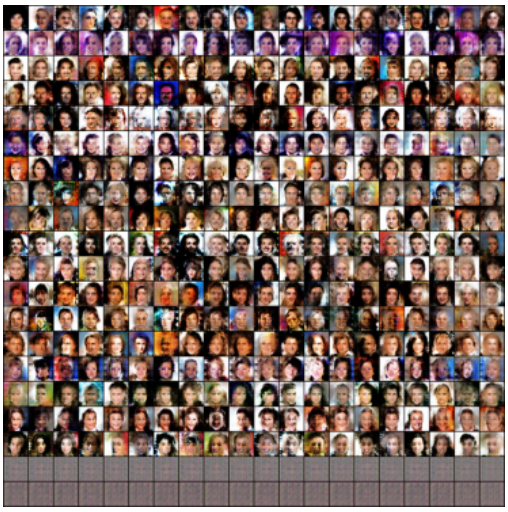


(f) CMS values throughout training (DDPM)

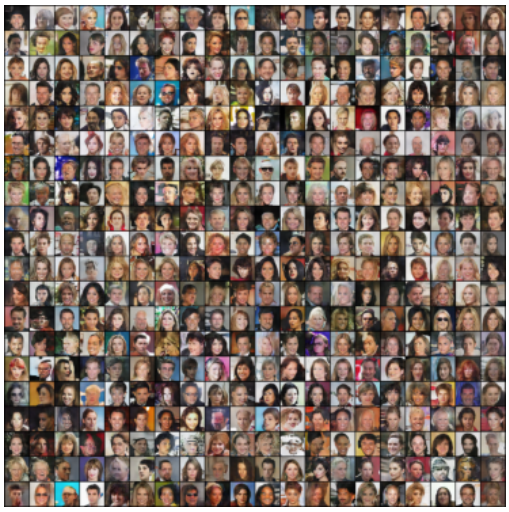
Figure 4: **Top-Left:** The identified clusters for ChestMNIST match how a human may separate the image structure: There are three clusters for the lung area, one for the abdomen, and two for the upper chest and background. **Top-Right:** The correlation matrix in terms of the CKA values indicates how well the clusters can be separated. The blocks on the diagonal are ordered by cluster number. As can be seen, most clusters are fairly independent. Cluster 6 could be further separated. **Mid:** Comparing the cluster-wise CMS values throughout training of DCGAN and DDPM architectures shows the difficulty of learning each cluster. The DCGAN architectures have a performance drop mostly due to Cluster 5 and 6 around Epoch 4. **Lower:** The cluster-wise CMS values successfully represent the image-wise CMS.

A Overview

In the following, we discuss additional experiment results and details in Appendix B, practical time and space complexity of Algorithm 1 in Appendix C, and missing proofs in Appendix D.



(a) Epoch 1



(b) Epoch 21

(c) Epoch 41

(d) Epoch 49

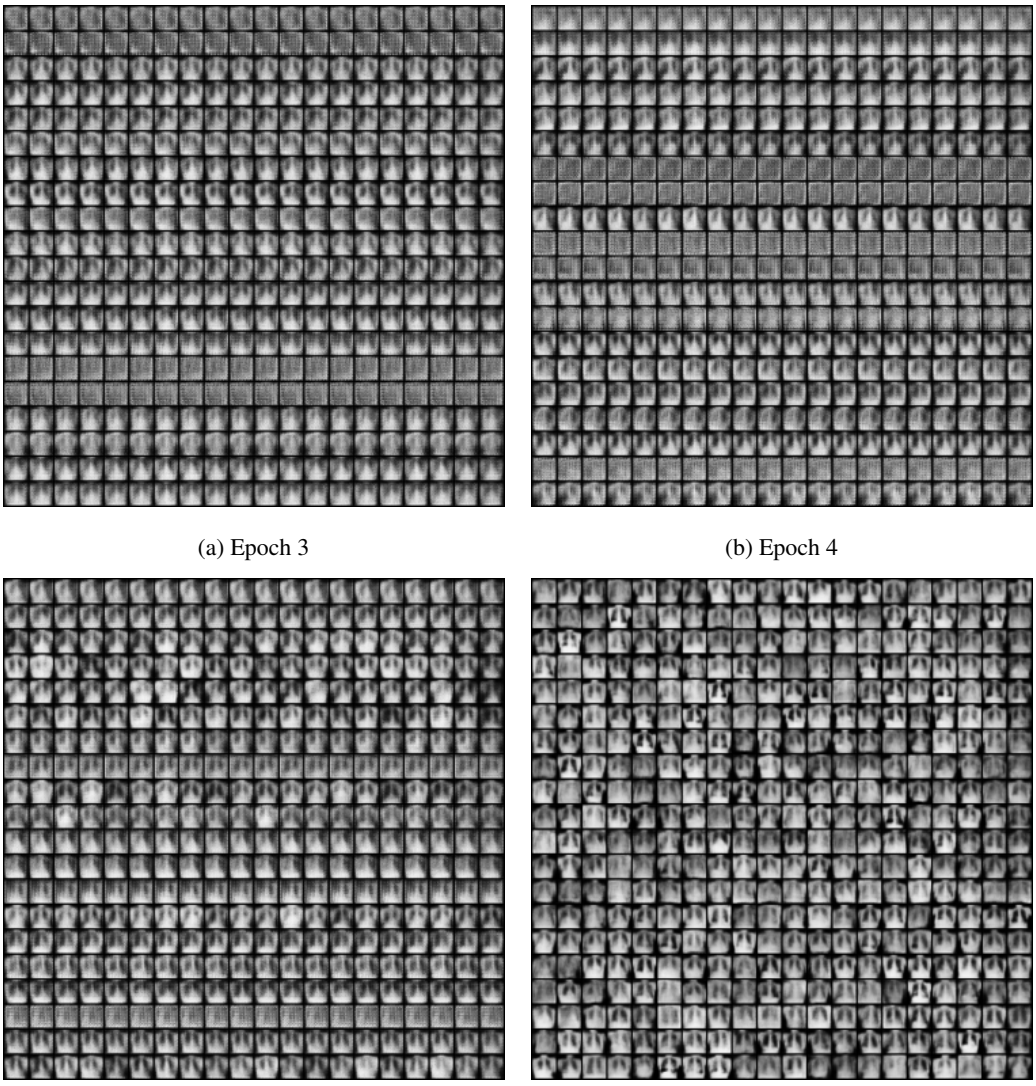
Figure 5: Generated samples of the twenty training runs (each row is a seed, each column a sample of a respective seed). Initially, all models are improving their fit. At 21 epochs, no further improvements are visible. At 41 epochs, the training of two models collapsed. At 49 epochs, the training of two additional models collapsed. The collapses are visible in all evaluation metrics in Figure 3, but only with our approach we can quantify the extend to which the individual pixel regions are affected.

B Additional Experimental Results and Details

In this section, we give more experimental results and details, which are missing in the main part.

B.1 ChestMNIST and additional CelebA Figures

We train 20 seeds of the DCGAN and DDPM architecture on the provided training set of ChestMNIST. Generated samples of the DCGAN architecture are presented in Figure 6. In Figure 4, we show the corresponding plots of ChestMNIST as in Figure 2. Specifically, we also discover a meaningful cluster partition of the image grid: It becomes clear in what regions the lungs, the abdomen, and the background are located. The CKA matrix shows that the large background region in brown is sparse and may be split up in additional clusters. The cluster-wise CMS values show how each architecture



(a) Epoch 3

(b) Epoch 4

(c) Epoch 5

(d) Epoch 10

Figure 6: Generated samples of the twenty training runs (each row is a seed, each column a sample of a respective seed) of the DCGAN architecture on ChestMNIST.

behaves during training: The DCGAN models suffer from a performance drop after 4 epochs, which can be assigned to Cluster 4 and 6 (the background). The DDPM models are more stable but learning Cluster 6 takes longer than the other clusters. The image-wise CMS values at the bottom of Figure 4 indicate that the cluster-wise CMS values are representative for the image-wise CMS.

We also show samples of generated images in Figure 5, which matches the observed trends of the evaluation metrics in Figure 3.

B.2 Experimental Details

All models were trained on a machine equipped with an AMD Ryzen 9 3950X CPU, an Nvidia RTX 4090 GPU, and 128GB of RAM. However, it is important to note that such high-end hardware is not strictly necessary for training these models; similar results can be obtained on less powerful systems, albeit with potentially longer training times. We use a random split of 90% of the CelebA dataset for training, utilizing the original model architecture as described in [Radford et al., 2015]. The specific

split can be reproduced via our source code. Each model and training run is initialized with a unique random seed ranging from 0 to 20.

The models for CelebA were trained with a consistent set of hyperparameters: a batch size of 128, generator and discriminator feature maps set to 64, a learning rate of 0.0002, and Adam optimizer β values of (0.5, 0.999). Binary Cross-Entropy (BCE) loss was used for training both the generator and discriminator. These settings were uniformly applied across all models.

For ChestMNIST we also train with a consistent set of hyperparameters: For the DCGAN architecture a batch size of 128, generator and discriminator feature maps set to 64, a learning rate of 1e-05, and Adam optimizer β values of (0.5, 0.999). Binary Cross-Entropy (BCE) loss was used for training both the generator and discriminator. These settings were uniformly applied across all models.
For the DDPM architecture, we base our implementation on `https://github.com/tcapelle/Diffusion-Models-pytorch` with similar hyperparameters.

Similar to the CKA, we also compute the MMD and CMS via their kernel representations according to the following. Given two datasets $\mathbf{X} = (X_1, \dots X_n) \overset{iid}{\sim} \mathbb{P}_X$ and $\mathbf{Y} = (Y_1, \dots Y_m) \overset{iid}{\sim} \mathbb{P}_Y$, we use $\widehat{\|\mu_{\mathbb{P}_X}\|^2_{\mathcal{H}}} \coloneqq \frac{1}{n^2} \sum_{i,j=1}^{n} k(X_i, X_j)$ as estimator for $\|\mu_{\mathbb{P}_X}\|^2_{\mathcal{H}}$, $\widehat{\|\mu_{\mathbb{P}_Y}\|^2_{\mathcal{H}}} \coloneqq \frac{1}{m^2} \sum_{i,j=1}^{m} k(X_i, X_j)$ as estimator for $\|\mu_{\mathbb{P}_Y}\|^2_{\mathcal{H}}$, and $\widehat{\langle \mu_{\mathbb{P}_X}, \mu_{\mathbb{P}_Y} \rangle_{\mathcal{H}}} \coloneqq \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k(X_i, Y_j)$ as estimator for $\langle \mu_{\mathbb{P}_X}, \mu_{\mathbb{P}_Y} \rangle_{\mathcal{H}}$.
Based on [Gretton et al., 2012] and [Kübler et al., 2019] we use these as plugins for the MMD estimator

$$\widehat{\mathrm{MMD}^2} \coloneqq \widehat{\|\mu_{\mathbb{P}_X}\|^2_{\mathcal{H}}} + \widehat{\|\mu_{\mathbb{P}_Y}\|^2_{\mathcal{H}}} - 2\widehat{\langle \mu_{\mathbb{P}_X}, \mu_{\mathbb{P}_Y} \rangle_{\mathcal{H}}} \tag{16}$$

and the CMS estimator

$$\widehat{\mathrm{CMS}} \coloneqq \frac{\widehat{\langle \mu_{\mathbb{P}_X}, \mu_{\mathbb{P}_Y} \rangle_{\mathcal{H}}}}{\widehat{\|\mu_{\mathbb{P}_X}\|^2_{\mathcal{H}}} \widehat{\|\mu_{\mathbb{P}_Y}\|^2_{\mathcal{H}}}}. \tag{17}$$

To reduce the runtime complexity, the MMD and CMS estimator are computed based on mini-batches of size 150 and then averaged across all blocks similar to [Zaremba et al., 2013]. We use a total of 1200 CelebA and 1350 ChestMNIST test instances for computing the errors at each epoch. The CKA is computed on mini-batches of size 100, and we used 1000 CelebA training instances and 2000 ChestMNIST training instances for its computation.

C Practical Time and Space Complexity

In the following, we discuss the time and space complexity of Algorithm 1. Let n_{tr} be the size of the training data, and $d = wh$ the number of pixels. The CKA estimator of Equation 13 has a runtime complexity of $O\left(n_{\mathrm{tr}}^3\right)$ and a space complexity of $O\left(n_{\mathrm{tr}}^2\right)$ due to multiplication of the kernel matrices. In Algorithm 1, we compute the CKA estimator in a nested for-loop over the number of pixels, resulting in a runtime complexity of $O\left(d^2 n_{\mathrm{tr}}^3\right)$. The space complexity does not increase since we do not require the respective kernel matrices after computing each CKA value. We can reduce the runtime complexity via the following. We may split the training data into mini-batches of size m_{CKA} and then average the CKA values across all mini-batches. This results in a runtime complexity of $O\left(d^2 m_{\mathrm{CKA}}^2 n_{\mathrm{tr}}\right)$ and space complexity of $O\left(m_{\mathrm{CKA}}^2\right)$. Further, we may use less training data since the estimator may converge with less data than available. In our case, we used mini-batches of size $m_{\mathrm{CKA}} = 100$ and $n_{\mathrm{tr}} = 1000$ training data for CelebA and $m_{\mathrm{CKA}} = 100$ and $n_{\mathrm{tr}} = 2000$ for ChestMNIST. The quadratic scaling with the number of pixels can be reduced using a window of pixels as kernel inputs. However, this was not necessary in our case.

The CMS estimator has a runtime and space complexity identical to the CKA estimator assuming $n \geq n'$, where n' is the number of generated images per iteration. We use the same mini-batch approach as for the CKA estimator, where m_{CMS} is the number of instances in a mini-batch. We also only use a subset of size n_{te} of the test data. In our experiments, we use $n_{\mathrm{te}} = 1200$ and $m_{\mathrm{CMS}} = 150$ for CelebA, and $n_{\mathrm{te}} = 1350$ and $m_{\mathrm{CMS}} = 150$ for ChestMNIST. The runtime and space complexities of the CMS estimator with respect to the number of pixels depend on the kernel choice. They can be neglected for the RBF and Laplacian kernel. We assume the number of chosen clusters is rather small (for example < 10), so we omit it as a variable.

D Missing Proofs

In the following, we present the proof for Theorem 1.

Proof. First, note that we can disentangle the mean embedding $\mu_{\mathbb{P}_X}$ due to the assumption $\mathrm{CKA}_{k^{\otimes|I|},k^{\otimes|I'|}}\left(\mathbb{P}_{X_I X_{I'}}\right) = 0$ and Equivalence 14 via

$$\begin{aligned}\mu_{\mathbb{P}_X} &= \mathbb{E}_{X\sim\mathbb{P}_X}\left[\bigotimes_{i=1}^{d}\phi\left(X_i\right)\right] = \mathbb{E}_{X\sim\mathbb{P}_X}\left[\bigotimes_{I\in\mathbf{I}}\bigotimes_{i\in I}\phi\left(X_i\right)\right] \\ &\overset{\text{Assumption}}{=} \bigotimes_{I\in\mathbf{I}}\mathbb{E}_{X\sim\mathbb{P}_X}\left[\bigotimes_{i\in I}\phi\left(X_i\right)\right] = \bigotimes_{I\in\mathbf{I}}\mu_{\mathbb{P}_{X_I}}.\end{aligned} \tag{18}$$

The same steps apply for $\mu_{\mathbb{P}_Y}$ as well. Further, note that for any $g_1, g_2 \in \mathcal{H}$ and $h_1, h_2 \in \mathcal{H}'$ we have $\langle g_1 \otimes h_1, g_2 \otimes h_2\rangle_{\mathcal{H}\otimes\mathcal{H}'} = \langle g_1, g_2\rangle_{\mathcal{H}}\langle h_1, h_2\rangle_{\mathcal{H}'}$. Now, we can use the disentangled mean embeddings to also disentangle the overall CMS into a product of cluster-wise CMS as stated in Theorem 1, since

$$\begin{aligned}\mathrm{CMS}_{k^{\otimes d}}\left(\mathbb{P}_X, \mathbb{P}_Y\right) &= \frac{\langle\mu_{\mathbb{P}_X}, \mu_{\mathbb{P}_Y}\rangle_{\mathcal{H}^{\otimes d}}}{\|\mu_{\mathbb{P}_X}\|_{\mathcal{H}^{\otimes d}}\|\mu_{\mathbb{P}_Y}\|_{\mathcal{H}^{\otimes d}}} = \frac{\left\langle\bigotimes_{I\in\mathbf{I}}\mu_{\mathbb{P}_{X_I}}, \bigotimes_{I\in\mathbf{I}}\mu_{\mathbb{P}_{Y_I}}\right\rangle_{\mathcal{H}^{\otimes|I|}}}{\left\|\bigotimes_{I\in\mathbf{I}}\mu_{\mathbb{P}_{X_I}}\right\|_{\mathcal{H}^{\otimes|I|}}\left\|\bigotimes_{I\in\mathbf{I}}\mu_{\mathbb{P}_{Y_I}}\right\|_{\mathcal{H}^{\otimes|I|}}} \\ &= \prod_{I\in\mathbf{I}}\frac{\left\langle\mu_{\mathbb{P}_{X_I}}, \mu_{\mathbb{P}_{Y_I}}\right\rangle_{\mathcal{H}^{\otimes|I|}}}{\left\|\mu_{\mathbb{P}_{X_I}}\right\|_{\mathcal{H}^{\otimes|I|}}\left\|\mu_{\mathbb{P}_{Y_I}}\right\|_{\mathcal{H}^{\otimes|I|}}} = \prod_{I\in\mathbf{I}}\mathrm{CMS}_{k^{\otimes|I|}}\left(\mathbb{P}_{X_I}, \mathbb{P}_{Y_I}\right).\end{aligned} \tag{19}$$

□

Based on the property that CMS always lies within $[-1, 1]$, Theorem 1 directly leads to the following fact relevant for interpretation.

Corollary 1. *Under the same assumptions as in Theorem 1, it holds for all $I \in \mathbf{I}$ that*

$$\left|\mathrm{CMS}_{k^{\otimes d}}\left(\mathbb{P}_X, \mathbb{P}_Y\right)\right| \leq \left|\mathrm{CMS}_{k^{\otimes|I|}}\left(\mathbb{P}_{X_I}, \mathbb{P}_{Y_I}\right)\right|. \tag{20}$$

In other words, the CMS of the whole image grid can never surpass the CMS of any cluster. This important fact tells us that we may never expect a smaller similarity between prediction and target in any cluster compared to the overall similarity. If this property is violated in practice, we will have to be wary of violated assumptions, which may affect the correctness of interpretations based on Theorem 1.