Comparative Analysis in Preimage Algorithms of Kernel PCA

Wojciech Czaja, Canran Ji[*]
Department of Mathematics
University of Maryland - College Park
College Park, MD 20742-4015, USA

Abstract

We study the kernel PCA (kPCA) pre-image problem in image denoising by benchmarking classical algorithms and introducing two neural network adversarial pre-imaging models, DCGAN-KPCAnet and WGAN-KPCAnet. Our results show that WGAN-KPCAnet delivers superior reconstruction results and is robust to noise compared to baselines.

1 Introduction

Recent work on the SINGLE pipeline for graphene liquid-cell electron microscopy has shown that kernel PCA (kPCA) denoising can markedly enhance signal-to-noise ratios and enable atomic-resolution reconstructions of nanocrystal dynamics[Ji, 2025, Meana-Pañeda et al., 2025, Reboul et al., 2021] . This study motivates our present focus on the inverse kPCA problem: recovering images from nonlinear kPCA embeddings.

Building on this motivation, we recall that the development of kPCA is rooted in over a century of research on dimensionality reduction. Principal component analysis (PCA) was introduced in [Pearson, 1901] and then independently re-developed and re-discovered by others, e.g., [Hotelling, 1933, Eckart and Young, 1936, Karhunen, 1947]. But it was almost a century later, when Schölkopf and his collaborators generalized the PCA algorithm, using the nonlinear kernel framework, [Schölkopf, 1997, Schölkopf et al., 1997, 1998a,b, Mika et al., 1999, Schölkopf and Smola, 2002], forming foundations of the modern nonlinear manifold learning theory. Thus, their construction is often considered to be a prototypical example of techniques called nonlinear dimension reduction methods. In the past 25 years, many novel nonlinear data representation algorithms have been proposed, often with superior performance and the ability to efficiently represent complex nonlinear data. However, one of the main advantages of kPCA is the fact that it allows for the existence of an approximate “inverse". This significantly broadens the applicability of kPCA, and also forms the basis for the pre-image reconstruction methods. It is the kPCA pre-image algorithms which are the object of interest in this paper. In Section 2, we analyze and compare the performance of 3 state-of-the-art algorithms for kPCA pre-image reconstruction. In Section 3, we propose 2 adversarial network-based pre-imaging models and compare their performance with the the baseline algorithms.

1.1 Mathematical formulation of kPCA

Let $\{x_i\}_{i=1}^n \subset \mathbb{R}^d$ denote the dataset of n input samples in d-dimensional space. Let $\phi : \mathbb{R}^d \to \mathcal{H}$ be a nonlinear feature map into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ associated with a Mercer kernel $k(\cdot,\cdot)$ satisfying $k(x,y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$.

Kernel PCA (kPCA) performs principal component analysis in $\mathcal{H}$ without explicitly computing $\phi(x_i)$. Let $K \in \mathbb{R}^{n\times n}$ be the Gram matrix with entries $K_{ij} = k(x_i, x_j)$, and let $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ denote the

[*] Corresponding Author, email: poloji@umd.edu

Proceedings of the 1st Conference on Topology, Algebra, and Geometry in Data Science(TAG-DS 2025).

centering matrix, where $\mathbf{1}\in\mathbb{R}^n$ represents the all-ones column vector. The centered kernel matrix is $K_c = HKH$. The eigenvalue problem

$$K_c\,\boldsymbol{v}_i = n\,\lambda_i\,\boldsymbol{v}_i, \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq 0,$$

yields eigenvalues λ_i and corresponding normalized eigenvectors $\boldsymbol{v}_i \in \mathbb{R}^n$, which represent the coefficients of the principal directions in the dual space.

By the representer theorem, each feature-space principal axis is

$$u_i = \frac{1}{\sqrt{n\lambda_i}} \sum_{\ell=1}^{n} v_{i\ell}\,\phi(x_\ell),$$

so that $\{u_i\}$ forms an orthonormal set in $\mathcal{H}$. The m-dimensional embedding of a sample x_j is

$$\mathbf{z}_j = \big(\langle u_1, \phi(x_j)\rangle, \ldots, \langle u_m, \phi(x_j)\rangle\big)^\top = \Big(\textstyle\sum_\ell v_{1\ell}k(x_\ell, x_j), \ldots, \sum_\ell v_{m\ell}k(x_\ell, x_j)\Big)^\top,$$

yielding a nonlinear representation $\mathbf{z}_j \in \mathbb{R}^m$ that captures the principal variance directions in $\mathcal{H}$ and is often used for denoising [Schölkopf et al., 1997].

1.2 Pre-image algorithms of kPCA

Given $\psi \in \mathcal{H}$, its pre-image x^* via ϕ is defined as

$$x^* = \arg\min_{x\in\mathbb{R}^d} \ \|\phi(x) - \psi\|^2.$$

In what follows we describe 3 state-of-the-art algorithms that solve this problem.

Fixed-point and kernel-ridge regression. For Gaussian k, an EM-like fixed-point update

$$x_{t+1} = \frac{\sum_{i=1}^n k(x_t, x_i)\,x_i}{\sum_{i=1}^n k(x_t, x_i)}$$

drives $\phi(x_t)\to\psi$. This iterative scheme originates from the work [Mika et al., 1999], which introduced a fixed-point formulation for the kPCA pre-image problem and demonstrated its effectiveness for feature-space denoising. Building upon this foundation, [Bakir et al., 2004] proposed two complementary strategies: a kernel ridge regression (KRR) approach that explicitly learns an inverse mapping $\Gamma' : \mathcal{H} \to \mathbb{R}^d$ via

$$\min_{\Gamma'} \sum_{i=1}^{n} \|x_i - \Gamma'(\phi(x_i))\|^2 + \lambda\|\Gamma'\|^2, \quad x^* \approx \Gamma'(\psi).$$

and forms the basis of the implementation adopted by `Sklearn`. In contrast, the refined fixed-point updatesometimes referred to here as Schölkopfs methodwas the algorithm incorporated into the SINGLE denoising pipeline.

Both approaches operate by learning or iteratively refining mappings in feature space, a strategy that has also proven effective in broader data-fusion and reconstruction settings. [Cloninger et al., 2017, Czaja et al., 2024]

MDS-based inversion. Assuming isotropic kernels, estimate $\tilde{d}_i = \|\psi - \phi(x_i)\|$, invert the kernel-induced distance to obtain input-space distances, then recover x^* with classical MDS [Kwok and Tsang, 2004].

In practice, this involves constructing the approximate distance matrix $D = [d_{ij}]$ where $d_{ij} = f^{-1}(\tilde{d}_{ij})$ for the kernel-dependent mapping f, centering it via double subtraction, and solving the corresponding eigenproblem to recover the coordinates of the unknown point x^*.

Because the optimization is linear and non-iterative, this method avoids local minima and provides a globally consistent embedding, although its accuracy degrades when the kernel is not strictly isotropic or when noise significantly perturbs the feature-space distances.

Nyström-hybrid. With $K_c = U\widetilde{\Lambda}U^\top$, approximate features by $\hat{\phi}(x) = \widetilde{\Lambda}^{-1/2}U^\top k_c(x)$. For target ψ, set $\hat{k}_x = \widetilde{\Lambda}^{-1/2}U^\top(\psi/\|\psi\|)$ and recover

$$x^* = \arg\min_x \|k_c(x) - \hat{k}_x\|^2,$$

or, when $\hat{k}_x(i) \geq 0$, use the closed form $x^* = \frac{\sum_i \hat{k}_x(i)\, x_i}{\sum_i \hat{k}_x(i)}$ [Arias et al., 2007].

This method combines the Nyström out-of-sample extension with the local averaging idea from direct approximation. By projecting ψ into the kernel eigenspace before reconstruction, it ensures consistency between training and test features and yields smoother, more stable pre-images.

Empirically, the hybrid scheme is efficient, robust under moderate noise, and tends to outperform purely iterative approaches in reconstruction fidelity.

2 Experiments evaluation of classical pre-image algorithms

We compare three Kernel PCA (kPCA) pre-image reconstruction methods: Sapiro's method [Arias et al., 2007, Kwok and Tsang, 2004], a kernel ridge regression-based inverse mapping (popularized by its Sklearn implementation), and Schölkopf's algorithm [Bakir et al., 2004], on classical image datasets, MNIST, CIFAR-10, and SVHN. For MNIST we consider two settings: *raw* (no trimming/normalization) and *normalized* (noise trimming and intensity normalization). CIFAR-10 and SVHN are converted to grayscale and evaluated in the normalized setting. For each dataset we use two kernels (RBF and cosine) and three sets of retained principal components (PCs: 20, 120, 220). We report average PSNR (dB), SSIM, and Pearson correlation coefficient (PCC) on clean inputs and on inputs corrupted by additive Gaussian noise.

Preprocessing and embeddings. For each dataset/kernel, we form centered Gram matrices and compute kPCA embeddings with $m \in \{20, 120, 220\}$. Reconstructions are evaluated on clean and noisy inputs under identical conditions across methods. Normalized settings apply outlier trimming and rescale intensities to the original dynamic range before inverse mapping.

Methods and experimental settings. Sapiro's method implements a Nyström-based hybrid inversion [Arias et al., 2007] and related distance-matching ideas [Kwok and Tsang, 2004]. The Sklearn-based method learns an inverse map $\Gamma' : \mathcal{H} \to \mathbb{R}^d$ via kernel ridge regression, then predicts $\Gamma'(\psi)$ at test time. Schölkopf's algorithm uses a fixed-point update for the RBF kernel (or a kernel-distance inversion where applicable), with iteration caps and tolerance-based stopping.

Implementation Credits. The implementation of Sapiros algorithm was adapted from code originally written by Alexander Cloninger and Timothy Doster, with additional contributions from Jeremiah Emidih, during their graduate studies at the University of Maryland under the mentorship of Dr. Wojciech Czaja. Their codebase provided the foundation for the Nyström-hybrid inversion and distance-matching routines [Cloninger, 2014, Doster, 2014, Cloninger and Czaja, 2014, Emidih and Czaja, 2017]. The implementation of the Schölkopf pre-image algorithm was completed with the assistance of Dr. Cong Tuan Son Van (National Cancer Institute, NIH). The kernel-ridge baseline follows `sklearn.decomposition.KernelPCA` and its inverse transform, credited to the Scikit-learn developers, particularly Andreas Müller and Arnaud Joly.

2.1 Experimental results

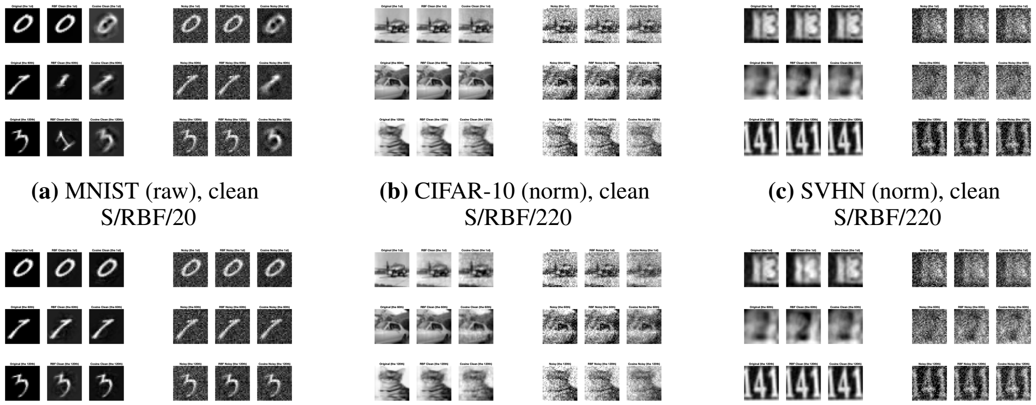
We summarize all experiments across datasets, kernels, principalcomponent counts (PCs), and noise conditions with a single aggregate table of best values and one illustrative figure panel. The table reports, for each dataset and clean/noisy setting, the highest PSNR, SSIM, and PCC achieved over all methods and PCs, together with the winning configuration. The figure panel provides qualitative examples that reflect these trends without attempting exhaustive coverage.

Table 1 reports, for each dataset and noise condition, the best reconstruction quality obtained across all combinations of classical methods, kernels, and principal-component counts. Because the full experimental grid is large, this compact summary provides a clear picture of the highest achievable performance on each dataset under our evaluation budget.

Table 1: Overall best reconstruction performance (PSNR in dB, SSIM, PCC) across datasets and noise conditions. Each entry reports the best value and the achieving configuration.

Dataset	Best PSNR (dB)	Best SSIM	Best PCC
Without noise			
MNIST (raw)	34.2829 (Sapiro, RBF, PC=220)	0.87190 (Sapiro, RBF, PC=220)	0.99487 (Schölkopf, Cosine, PC=220)
MNIST (norm)	30.4618 (Sapiro, RBF, PC=220)	0.66136 (Sklearn, RBF, PC=220)	0.99487 (Schölkopf, Cosine, PC=220)
CIFAR-10 (norm)	32.0257 (Schölkopf, RBF, PC=20)	0.92645 (Schölkopf, RBF, PC=220)	0.98085 (Schölkopf, RBF, PC=220)
SVHN (norm)	31.7435 (Schölkopf, RBF, PC=120)	0.97229 (Schölkopf, RBF, PC=220)	0.99711 (Schölkopf, Cosine, PC=220)
With noise			
MNIST (raw)	18.5474 (Sapiro, RBF, PC=220)	0.48545 (Schölkopf, Cosine, PC=20)	0.86955 (Schölkopf, RBF, PC=120)
MNIST (norm)	17.9833 (Schölkopf, RBF, PC=20)	0.30864 (Sapiro, RBF, PC=120)	0.91387 (Schölkopf, RBF, PC=120)
CIFAR-10 (norm)	19.9966 (Schölkopf, RBF, PC=120)	0.52890 (Sklearn, Cosine, PC=220)	0.85676 (Sklearn, Cosine, PC=220)
SVHN (norm)	19.7507 (Schölkopf, RBF, PC=120)	0.50728 (Schölkopf, Cosine, PC=20)	0.78210 (Sklearn, Cosine, PC=20)

Observations. Schölkopfs inversion with the RBF kernel dominates for PSNR and SSIM on CIFAR-10 and SVHN, while MNIST sometimes favors Sapiro with RBF at high PC for PSNR. Under noise, Schölkopf+RBF is generally strongest for PSNR, with Cosine occasionally giving the best PCC values via Sklearn. Overall, Schölkopfs method shows the most consistent robustness.



(a) MNIST (raw), clean S/RBF/20

(b) CIFAR-10 (norm), clean S/RBF/220

(c) SVHN (norm), clean S/RBF/220

(d) MNIST (raw), noisy S/RBF/120

(e) CIFAR-10 (norm), noisy S/RBF/120

(f) SVHN (norm), noisy S/RBF/120

Figure 1: Representative reconstructions across datasets under clean (top) and noisy (bottom) conditions. Abbreviations: S = Schölkopf, RBF = Gaussian kernel; numbers indicate principal components (PC). Each label sits directly beneath its image by construction.

2.2 Analysis

Tables 2–3 summarize reconstruction trends across algorithms (Sapiro, Sklearn, Schölkopf), kernels (RBF, Cosine), and principal component (PC) settings. Best values are in **bold**. In contrast to Table 1, which collapses the full experimental grid to a single best-performing configuration per dataset and noise level, Tables 2 and 3 reorganize the same results by algorithm and by clean versus noisy condition.

Tables 2 and 3 regroup the same experimental results by algorithm. For each method we report its best-performing combination of kernel and principalcomponent count among the tested values $\{20, 120, 220\}$. These PC values were chosen to span low-, mid-, and higher-dimensional embeddings under the conference page limits. A strict fixed-PC comparison across these choices is reported in Section 5.4 of [Ji, 2025], and the same qualitative ranking of methods is observed there.

Table 2: Aggregated results (without noise): best reconstruction metrics (PSNR in dB, SSIM, PCC). Each entry shows the top value for the algorithm on the dataset across all kernels and PC counts, with kernel and PC in parentheses.

	MNIST (raw)	MNIST (norm)	CIFAR-10 (norm)	SVHN (norm)
Sapiro	**PSNR=34.28 (RBF, 220)**	**PSNR=30.46 (RBF, 220)**	PSNR=18.79 (RBF, 220)	PSNR=15.02 (RBF, 220)
	SSIM=0.872 (RBF, 220)	SSIM=0.595 (RBF, 120)	SSIM=0.667 (RBF, 220)	SSIM=0.629 (RBF, 220)
	PCC=0.952 (RBF, 220)	PCC=0.952 (RBF, 220)	PCC=0.832 (RBF, 220)	PCC=0.862 (RBF, 220)
Sklearn	PSNR=19.21 (RBF, 220)	PSNR=21.18 (RBF, 220)	PSNR=21.57 (RBF, 220)	PSNR=14.74 (RBF, 220)
	SSIM=0.673 (RBF, 220)	**SSIM=0.661 (RBF, 220)**	SSIM=0.852 (RBF, 220)	SSIM=0.683 (RBF, 220)
	PCC=0.925 (RBF, 220)	PCC=0.925 (RBF, 220)	PCC=0.933 (RBF, 220)	PCC=0.962 (Cosine, 220)
Schölkopf	PSNR=29.22 (RBF, 20)	PSNR=28.17 (RBF, 20)	**PSNR=32.03 (RBF, 20)**	**PSNR=31.74 (RBF, 120)**
	SSIM=0.811 (RBF, 20)	SSIM=0.577 (RBF, 20)	**SSIM=0.926 (RBF, 220)**	**SSIM=0.972 (RBF, 220)**
	PCC=0.995 (Cosine, 220)	**PCC=0.995 (Cosine, 220)**	**PCC=0.987 (Cosine, 220)**	**PCC=0.997 (Cosine, 220)**

Table 3: Aggregated results (with noise): best reconstruction metrics (PSNR in dB, SSIM, PCC). Each entry shows the top value for the algorithm on the dataset across all kernels and PC counts, with kernel and PC in parentheses.

	MNIST (raw)	MNIST (norm)	CIFAR-10 (norm)	SVHN (norm)
Sapiro	**PSNR=18.55 (RBF, 220)**	PSNR=17.66 (RBF, 220)	PSNR=16.09 (RBF, 220)	PSNR=15.55 (RBF, 220)
	SSIM=0.336 (RBF, 220)	**SSIM=0.309 (RBF, 120)**	SSIM=0.389 (RBF, 220)	SSIM=0.300 (RBF, 220)
	PCC=0.830 (RBF, 220)	PCC=0.884 (RBF, 220)	PCC=0.707 (RBF, 220)	PCC=0.612 (RBF, 220)
Sklearn	PSNR=17.69 (RBF, 220)	PSNR=17.50 (RBF, 220)	PSNR=18.01 (Cosine, 220)	PSNR=16.27 (Cosine, 220)
	SSIM=0.315 (RBF, 220)	SSIM=0.288 (RBF, 220)	**SSIM=0.529 (Cosine, 220)**	SSIM=0.412 (Cosine, 20)
	PCC=0.822 (RBF, 220)	PCC=0.878 (RBF, 220)	**PCC=0.857 (Cosine, 220)**	**PCC=0.782 (Cosine, 20)**
Schölkopf	PSNR=18.22 (RBF, 120)	**PSNR=17.98 (RBF, 20)**	**PSNR=20.00 (RBF, 120)**	**PSNR=19.75 (RBF, 120)**
	SSIM=0.485 (Cosine, 20)	SSIM=0.298 (Cosine, 120)	SSIM=0.526 (RBF, 120)	**SSIM=0.507 (Cosine, 20)**
	PCC=0.870 (RBF, 120)	**PCC=0.911 (Cosine, 20)**	PCC=0.835 (RBF, 120)	PCC=0.764 (Cosine, 20)

Without noise. Across datasets without additive noise, as summarized in Table 2, Schölkopfs method yields the best overall performance, dominating on CIFAR-10 and SVHN across PSNR, SSIM, and PCC. On MNIST (raw), Sapiro attains the highest PSNR and SSIM, while Schölkopf achieves the strongest PCC. For MNIST (normalized), Sklearn provides the best SSIM, whereas Schölkopf leads in PSNR and PCC.

With noise. In the presence of noise, as summarized in Table 3, all methods exhibit degraded performance. Schölkopf remains the most effective overall, attaining the highest PSNR on three datasets and the best SSIM on two. Sapiro achieves the strongest SSIM on both MNIST variants, while Sklearn obtains the highest PCC on CIFAR-10 and SVHN.

Aggregate comparison. Overall, Schölkopf consistently delivers the most competitive results, especially on CIFAR-10 and SVHN. Sapiro provides strong PSNR/SSIM on MNIST (raw) and occasionally matches Schölkopf on normalized MNIST. Sklearn rarely dominates but yields the top SSIM (clean MNIST) and PCC (with noise on CIFAR-10 and SVHN).

Statistical significance testing. For each metric (PSNR, SSIM, PCC), we ran a oneway ANOVA across algorithms (Sapiro, Sklearn, Schölkopf) over 48 scenarios (4 datasets × 3 principalcomponent settings × 2 noise levels × 2 kernels) at $\alpha = 0.05$, followed by Dunnett post-hoc tests with Schölkopf as the control. All omnibus tests reject the global null of equal means.

Table 4: Oneway ANOVA omnibus p-values at $\alpha = 0.05$.

Metric	Global p-value
PSNR	9.5089×10^{-5}
SSIM	2.7778×10^{-3}
PCC	2.7923×10^{-7}

Dunnett comparisons test $H_{0,j} : \mu_{\text{Schölkopf}} = \mu_j$ vs. $H_{a,j} : \mu_{\text{Schölkopf}} > \mu_j$ for $j \in \{\text{Sapiro}, \text{Sklearn}\}$. Table 5 reports one-sided p-values and estimated mean differences $(\widehat{\mu}_{\text{Schölkopf}} - \widehat{\mu}_j)$. All reported differences are positive, and all Dunnett p-values are below 0.05, indicating significantly higher means for Schölkopfs algorithm on all three metrics.

Table 5: Dunnett post-hoc results: one-sided p-values and estimated mean differences ($\widehat{\mu}_{\text{Schölkopf}} - \widehat{\mu}_j$) at $\alpha = 0.05$.

Metric	vs Sapiro p	$\widehat{\mu}_{\text{Schölkopf}} - \widehat{\mu}_{\text{Sapiro}}$	vs Sklearn p	$\widehat{\mu}_{\text{Schölkopf}} - \widehat{\mu}_{\text{Sklearn}}$
PSNR	9.1689×10^{-4}	2.9845	4.7501×10^{-5}	3.7047
SSIM	1.7326×10^{-3}	0.1264	4.7124×10^{-3}	0.1135
PCC	4.8945×10^{-8}	0.2106	7.7250×10^{-4}	0.1253

Conclusion and summary. Schölkopfs pre-image reconstruction provides the most consistent high-fidelity performance across datasets and noise conditions, with superior PSNR, SSIM, and PCC in the majority of scenarios. Sapiros method attains the highest SSIM or PSNR on clean MNIST variants, and Sklearns inverse mapping yields strong PCC on CIFAR-10 and SVHN under noise. The ANOVA omnibus tests (Table 4) and Dunnett post-hoc comparisons (Table 5) confirm, at $\alpha = 0.05$, that Schölkopfs algorithm achieves significantly higher mean PSNR, SSIM, and PCC than both Sapiro and Sklearn under the reported settings.

3 Inverse kernel PCA reconstruction via generative models

In this section we propose two alternative kPCA pre-image reconstruction schemes based on generative models and compare them against the classical constructions already analyzed in Sections 1 and 2. For any noisy input $\tilde{x}$ we write $z = \Psi(\tilde{x}) \in \mathbb{R}^m$ for its $m$dimensional kPCA code, and let Φ denote its projection in feature space onto the first m principal directions. The preimage problem then amounts to recovering an image whose feature representation is closest to Φ. Rather than solving this optimization directly, we learn a parametric mapping $G_\theta : \mathbb{R}^m \to \mathbb{R}^d$ such that $G_\theta(z)$ approximates the corresponding clean image.

Adversarial training framework. Let p_{data} be the empirical distribution of the clean data $\{x_i\}$, and let p_z be the distribution of the latent codes $\{z_i\}$ produced by kPCA on noisy samples. We introduce a discriminator network $D_\xi : \mathbb{R}^d \longrightarrow [0, 1]$, where ξ are discriminator parameters, which aims to distinguish real samples $x_i \sim p_{\text{data}}$ from generated reconstructions $G_\theta(z)$ with $z \sim p_z$. The generator G_θ is then trained to fool D_ξ while simultaneously minimizing a pixelwise reconstruction error. Concretely, the adversarial loss and reconstruction loss are defined as follows.

Generator Loss. The generator G_θ minimizes

$$\mathcal{L}_G(\theta, \xi) \;=\; -\mathbb{E}_{z\sim p_z}\Big[\log D_\xi\big(G_\theta(z)\big)\Big] \;+\; \lambda\,\mathbb{E}_{(z,x)\sim p_{\text{pair}}}\Big[\|\,G_\theta(z) \;-\; x\|_2^2\Big],$$

where $\lambda > 0$ is a tradeoff parameter and p_{pair} is the joint distribution pairing each latent code z with its corresponding clean sample x. The first term encourages $G_\theta(z)$ to lie on the manifold of real images, and the second term enforces fidelity to the true preimage in the Euclidean sense.

Discriminator Loss. The discriminator D_ξ is trained to solve

$$\mathcal{L}_D(\xi, \theta) \;=\; -\mathbb{E}_{x\sim p_{\text{data}}}\big[\log D_\xi(x)\big] \;-\; \mathbb{E}_{z\sim p_z}\Big[\log\big(1 - D_\xi\big(G_\theta(z)\big)\big)\Big].$$

At each iteration, ξ is updated by descending $\nabla_\xi \mathcal{L}_D(\xi, \theta)$, and θ is updated by descending $\nabla_\theta \mathcal{L}_G(\theta, \xi)$. This adversarial optimization follows the original GAN framework in Goodfellow et al. [2014].

Wasserstein GAN (WGAN) Variant. In the WGAN formulation Arjovsky et al. [2017], the discriminator (often called the critic) D_ξ is unconstrained in its output, and the losses become

$$\begin{aligned}
\mathcal{L}_D^{\text{W}}(\xi, \theta) \;&=\; -\mathbb{E}_{x\sim p_{\text{data}}}\big[D_\xi(x)\big] \;+\; \mathbb{E}_{z\sim p_z}\big[D_\xi\big(G_\theta(z)\big)\big],\\
\mathcal{L}_G^{\text{W}}(\theta, \xi) \;&=\; -\mathbb{E}_{z\sim p_z}\big[D_\xi\big(G_\theta(z)\big)\big] \;+\; \lambda_*\,\mathbb{E}_{(z,x)\sim p_{\text{pair}}}\Big[\|\,G_\theta(z) \;-\; x\|_2^2\Big].
\end{aligned}$$

Under appropriate Lipschitzcontinuity constraints (enforced via weight clipping or gradient penalty), this yields a more stable training procedure and a meaningful notion of distance between distributions in terms of the EarthMovers distance.

Deep Convolutional GAN (DCGAN). The DCGAN model employs a deep convolutional generator and a multilayer perceptron discriminator. All experiments use $n_{\text{samples}} = 20000$ MNIST images normalized to $[0, 1]$, corrupted by additive Gaussian noise of standard deviation $\sigma = 0.5$, then split without shuffling into 80% noisy training and 20% clean test sets.

Generator. A latent vector $z \in \mathbb{R}^{3000}$ is first mapped by a fullyconnected layer to a $256 \times 7 \times 7$ feature map, followed by

$$\begin{aligned}&\text{ConvTranspose2d}(256 \to 128,\ 4 \times 4,\ \text{stride} = 2,\ \text{pad} = 1),\ \text{BatchNorm},\ \mathrm{ReLU},\\ &\text{ConvTranspose2d}(128 \to 64,\ 4 \times 4,\ \text{stride} = 2,\ \text{pad} = 1),\ \text{BatchNorm},\ \mathrm{ReLU},\\ &\text{Conv2d}(64 \to 1,\ 3 \times 3,\ \text{stride} = 1,\ \text{pad} = 1),\ \tanh.\end{aligned}$$

Discriminator. A fullyconnected network maps flattened $28 \times 28 = 784$ inputs through linear layers of sizes 512, 256, 128 (each followed by LeakyReLU with slope 0.2) and a final sigmoid output.

Training. Batch size 128, for 1500 epochs on GPU if available. Optimizers are Adam with

$$\begin{aligned}&\text{Generator: } lr = 1 \times 10^{-4},\ \beta = (0.5, 0.999),\\ &\text{Discriminator: } lr = 2 \times 10^{-4},\ \beta = (0.5, 0.999).\end{aligned}$$

The loss combines binary crossentropy adversarial term and reconstruction term

$$L_G = \mathbb{E}\big[\mathrm{BCE}(D(G(z)), 1)\big] \ +\ \lambda_{\text{recon}}\ \mathrm{MSE}\big(G(z), x\big), \quad \lambda_{\text{recon}} = 10.$$

At each epoch, PSNR and SSIM are computed on the first 16 test samples; the model with highest average PSNR (and separately SSIM) is recorded.

Wasserstein GAN (WGAN). The WGAN uses the same generator architecture. The discriminator (critic) differs by omitting the sigmoid activation and producing a realvalued score. Its loss is

$$L_D = \mathbb{E}\big[D(G(z))\big] \ -\ \mathbb{E}\big[D(x)\big], \qquad L_G = -\,\mathbb{E}\big[D(G(z))\big].$$

After each discriminator update, all its weights are clipped to $[-0.01, 0.01]$. We perform $n_{\text{critic}} = 5$ critic updates per generator update and adjust the Adam hyperparameters to $\beta = (0.5, 0.9)$. All other settings (batch size, λ_{recon}, number of epochs = 1500, data split, noise level) remain as in the DCGAN.

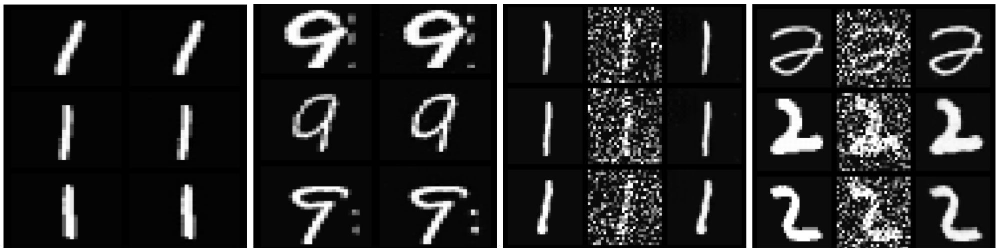
3.1 Quantitative results on MNIST

For completeness, we report the MNIST results on a per-digit basis. The overall mean is included in the final row, and the digit-wise values provide a more granular view of the reconstruction behavior.

Quantitative evaluation. Table 6 summarizes the reconstruction performance of DCGAN–KPCAnet and WGAN–KPCAnet on MNIST after 500 epochs, reporting only the averages of the best 5 and worst 5 PSNR values per digit. Without noise, DCGAN–KPCAnet achieves the highest overall quality, with a mean average of the best 5 reconstructions at 39.34 dB and worst 5 at 29.57 dB, slightly outperforming WGAN–KPCAnet at 38.00 dB and 28.84 dB. When noise is introduced, performance drops significantly. In this setting, WGAN–KPCAnet becomes more robust, reaching 33.88 dB on the best 5 and 27.50 dB on the worst 5, while DCGAN–KPCAnet falls to 18.62 dB and 13.00 dB.

Table 6: MNIST: comparison of average PSNR (dB) values across digits for dcgan–kpcanet and wgan–kpcanet after 500 epochs. For each digit, only the *average of best 5* and *average of worst 5* reconstructions are reported. Results are shown separately for clean and noisy settings. The last row shows column-wise means.

Digit	DCGAN Avg Best 5 (clean)	DCGAN Avg Worst 5 (clean)	WGAN Avg Best 5 (clean)	WGAN Avg Worst 5 (clean)	DCGAN Avg Best 5 (noisy)	DCGAN Avg Worst 5 (noisy)	WGAN Avg Best 5 (noisy)	WGAN Avg Worst 5 (noisy)
0	38.39	29.15	36.88	28.42	18.61	13.61	32.80	26.83
1	45.54	31.82	43.06	31.53	19.93	12.32	37.35	30.00
2	37.85	29.03	36.49	28.51	18.18	12.76	33.12	26.46
3	38.71	29.10	36.89	28.96	18.26	13.22	33.16	26.97
4	38.67	30.31	37.77	29.64	18.33	13.26	33.59	28.06
5	37.94	29.22	36.77	28.36	18.12	13.59	33.17	27.06
6	38.98	29.75	37.36	27.41	18.51	13.07	33.98	26.90
7	40.18	29.88	38.78	28.95	19.84	13.27	34.85	26.85
8	37.63	29.20	36.68	28.15	17.64	12.65	33.05	27.18
9	39.49	28.69	38.87	28.50	18.73	13.24	34.76	27.42
Mean	39.34	29.57	38.00	28.84	18.62	13.00	33.88	27.50



(a) DCGANdigit 1 (clean): best

(b) DCGANdigit 9 (clean): worst

(c) WGANdigit 1 (noisy): best

(d) WGANdigit 2 (noisy): worst

Figure 2: Representative DCGANKPCAnet (clean) and WGANKPCAnet (noisy) reconstructions on MNIST after 500 epochs. Each label is placed directly beneath its subfigure.

Figure 2 illustrates representative reconstruction examples. The top row shows DCGAN–KPCAnet on clean MNIST digits, while the bottom row highlights WGAN–KPCAnet on noisy digits. These examples confirm the quantitative trends: DCGAN excels in noise-free settings, but WGAN demonstrates greater robustness when noise is present.

3.2 Comparison with Schölkopfs pre-image algorithm

Table 7: Per-digit and overall average **PSNR** (dB) for **WGANKPCAnet** (Cosine, PC=300) vs. **Schölkopfs** pre-image (RBF, PC=90) on 400 MNIST inputs.

Digit	WGANKPCAnet	Schölkopf
0	21.31	16.22
1	24.82	18.22
2	20.26	16.03
3	20.94	16.38
4	21.19	16.52
5	20.66	16.25
6	21.19	16.44
7	22.12	16.86
8	20.42	16.40
9	21.70	16.94
Overall	21.46	16.66

Both algorithms are evaluated on the same set of 400 MNIST inputs. WGANKPCAnet employs a cosine kernel with PC=300, leveraging a high component count and larger feature dimension to maximize reconstruction fidelity, whereas Schölkopfs pre-image algorithm uses an RBF kernel with PC=90 (approximately 25% of the sample size), which was found to be the most stable and accurate choice for classical kernel-based recovery under the same low-sample conditions.

Although the kernel/PC choices differ, these configurations reflect the respective operating regimes in which each method performs optimally: the classical solver degrades for large PCs in heterogeneous sample settings, whereas the generative inverse model benefits from higher-dimensional latent representations. Thus, the comparison in Table 7 is a best-performing vs. best-performing evaluation rather than a fixed-kernel, fixed-PC experiment. A more exhaustive fixed-PC comparison over multiple component counts is provided in Section 5.4 of [Ji, 2025], and the resulting ordering between WGANKPCAnet and Schölkopfs method agrees with the ranking reported here.

As shown in Table 7, WGANKPCAnet achieves an overall average PSNR of 21.46 dB, outperforming Schölkopfs algorithm by 4.80 dB. The largest per-digit improvement occurs for digit 1 (+6.60 dB), and even the smallest gain for digit 7 (+1.24 dB) confirms consistent superiority across all digit classes.

Importantly, WGANKPCAnet not only surpasses the classical solver in the high-dimensional reconstruction regime (PC=300), but also remains superior in lower-dimensional settings (PC=90), underscoring the robustness of generative inverse solvers across a range of feature dimensions. This comparison extends the preceding section, where Schölkopfs algorithm emerged as the strongest classical baseline, by showing that neural-network-based inverse recovery can exceed classical performance even under modest-component constraints. Future work will study the interaction between kernel choice, component-to-sample ratio, and dataset heterogeneity in both frameworks.

4 Conclusion and summary

We investigated the kPCA pre-image problem through a systematic comparison of deterministic algorithms (Sapiro, Sklearn inverse mapping, Schölkopf) and adversarial generators conditioned on kPCA codes (DCGANKPCAnet, WGANKPCAnet). Across MNIST, CIFAR-10, and SVHN, Schölkopfs fixed-point method with an RBF kernel is the most reliable deterministic baseline, attaining the strongest PSNR/SSIM on CIFAR-10 and SVHN and competitive PCC overall. On MNIST, Sapiros Nyström-hybrid occasionally yields the highest clean reconstructions at large PC, while the Sklearn inverse sometimes leads in SSIM (clean MNIST) and PCC (noisy CIFAR-10/SVHN). Oneway ANOVA and Dunnett tests corroborate these trends, indicating significantly higher mean PSNR, SSIM, and PCC for Schölkopfs method relative to the other deterministic baselines under our evaluation.

For learned pre-images, DCGANKPCAnet provides the best clean MNIST reconstructions on average, whereas WGANKPCAnet exhibits greater robustness under additive noise. In a direct comparison on MNIST, WGANKPCAnet also surpasses Schölkopfs method in average PSNR over all digits. Overall, deterministic Schölkopf+RBF is preferred when a classical solver is required, while WGANKPCAnet is advantageous when robustness to noise or higher reconstruction fidelity on MNIST-like data is the priority. Future work will extend the evaluation to color inputs, broader corruption models, and additional kernels and stability-enhanced adversarial objectives.

While this study focuses on GAN-based pre-image mappings, alternative learned decoders such as variational or diffusion-based approaches may also be applicable. For instance, recent work by Tom Goldstein shows that diffusion architectures can be flexibly steered by external constraints, suggesting a plausible pathway for kPCA-conditioned diffusion decoders [Bansal et al., 2024]. Our experiments were restricted to MNIST as a controlled setting; extending the evaluation to more complex natural or scientific imaging datasets is a natural direction for future work. Although training the generative model is more expensive than running a classical solver once, the amortized per-sample cost after training is substantially lower, which is advantageous when large numbers of pre-image evaluations are required.

Acknowledgments and Disclosure of Funding

This research was supported by the UMD-NCI Partnership for integrative cancer research through its seed grant program, which funded the collaborative work carried out with the Center for Structural Biology (CSB), National Cancer Institute, NIH. The authors gratefully acknowledges Dr. Hans Elmlund for guidance and for the collaborative environment provided by his group.

References

P. Arias, G. Randall, and G. Sapiro. Connecting the Out-of-Sample and Pre-Image Problems in Kernel Methods. volume 29, 2007.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 2017.

G. H. Bakir, J. Weston, and B. Schölkopf. Learning to find pre-images. In *Advances in Neural Information Processing Systems 16*, pages 449–456. Max-Planck-Gesellschaft, MIT Press, jun 2004.

A. Bansal, H. M. Chu, A. Schwarzschild, R. Sengupta, M. Goldblum, J. Geiping, and T. Goldstein. Universal guidance for diffusion models. In *Proceedings of the International Conference on Learning Representations (ICLR 2024)*, 2024.

A. Cloninger. *Exploiting Data-Dependent Structure for Improving Sensor Acquisition and Integration*. Ph.D. dissertation, University of Maryland, College Park, 2014.

A. Cloninger and W. Czaja. LIDAR image recovery by incorporating heterogeneous imaging modalities. In *SPIE Vol. 9121, Remote Sensing*, page 91210C, 2014.

A. Cloninger, W. Czaja, and T. Doster. The pre-image problem for Laplacian Eigenmaps utilizing L1 regularization with applications to data fusion. *Inverse Problems*, 33(7), 2017.

W. Czaja, J. Emidih, B. Kolstoe, and R. G. Spencer. Hyperspectral reconstruction of skin through fusion of scattering transform features. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW 2024)*, pages 105–106. IEEE, 2024.

T. J. Doster. *Harmonic Analysis Inspired Data Fusion for Applications in Remote Sensing*. Ph.D. dissertation, University of Maryland, College Park, 2014.

C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3): 211–218, 1936.

J. Emidih and W. Czaja. Heterogeneous cancer cell line data fusion for identifying novel response determinants in precision medicine. In *Lecture Notes in Computer Science, vol. 10253, Computation, Physics and Beyond (IWCPB 2017)*, pages 414–419. Springer, 2017.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

C. Ji. *Uncertainty in Directional Representations, Preimages of Kernel Transformations and Applications*. Ph.D. dissertation, University of Maryland, College Park, 2025.

K. Karhunen. *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. Number 37 in A I. Annales Academiae Scientiarum Fennicae, 1947.

J. T. Kwok and I. W. Tsang. The Pre-Image Problem in Kernel Methods. *IEEE Transactions on Neural Networks*, 15(6):1517 – 1525, 2004.

R. Meana-Pañeda, C. Ji, C. T. S. Van, P. Ercius, H. Elmlund, and W. Czaja. Time-resolved atomic-resolution brownian tomography of single nanocrystals reveals size-dependent dynamics. *Science Advances*, 11, 2025.

S. Mika, B. Schölkopf, A. J. Smola, K-R. Müller, M. Scholz, and G. Rätsch. Kernel pca and de-noising in feature spaces. In *Advances in Neural Information Processing Systems 11*, pages 536–542. MIT Press, 1999.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2 (11):559–572, 1901.

C. F. Reboul, J. Heo, C. Machello, S. Kiesewetter, B. Kim, S. Kim, D. Elmlund, P. Ercius, J. Park, and H. Elmlund. SINGLE: Atomic-resolution structure identification of nanocrystals by graphene liquid cell EM. *Science Advances*, 7(5), 2021.

B. Schölkopf. *Support Vector Learning*. Oldenbourg Verlag, Munich, 1997.

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, 2002.

B. Schölkopf, A. Smola, and K-R. Müller. Kernel principal component analysis. pages 583–588, 1997.

B. Schölkopf, P. Knirsch, A. Smola, and C. Burges. Fast approximation of support vector kernel expansions, and an interpretation of clustering as approximation in feature spaces. In P. Levi et al., editors, *DAGM'98: Mustererkennung*, pages 124–132, Berlin, 1998a. Springer.

B. Schölkopf, A. J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. In *Advances in Neural Information Processing Systems 10*, pages 583–589. MIT Press, 1998b.