Eigengames

Kirill Tamogashev,
Ivan Shchekotov,
Alexandra Senderovich

PCA: previous approaches

- PCA task: $\underset{\mathbf{U}\in\mathbb{R}^{d\times k},\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\text{minimize}} \left\|\mathbf{U}\mathbf{U}^T\mathbf{X}-\mathbf{X}\right\|_F^2$
- Full SVD of sample covariance matrix:

$$\mathcal{O}(\min\{nd^2, n^2d\})$$

- Streaming k-PCA:
  - Frequent Directions
    - Approximates the top-k subspace
    - Three steps: copy the row of A, rotate, shrink (to create a zero row)
  - Oja's algorithm
    - Approximates the top-k eigenvectors

$$w_{t+1} - w_t = \eta(w_t^T X_t)X_t; \qquad w_{t+1}^T w_{t+1} = 1,$$

Algorithm 4.1.1 FREQUENTDIRECTIONS

```
Input: ℓ, A ∈ R^{n×d}
B ← 0^{ℓ×d}
for i ∈ 1, ..., n do
    B_{ℓ,:} ← A_{i,:}
    [U, Σ, V] ← svd(B)
    C ← ΣV^T
    δ ← Σ^2_{ℓ,ℓ}
    B ← sqrt(Σ^2 − δI_ℓ) · V^T
return B
```

$$\|A - \pi_B^k(A)\|_F \leq (1+\varepsilon)\|A - A_k\|_F$$

Symmetric Generalized Eigenvalue Problem (SGEP)

Definition (SGEP)

Given matrices $A = A^\top$, $B = B^\top$, $B \succ 0$,

$$Av = \lambda Bv, \tag{1}$$

defines Symmetric Generalized Eigenvalue Problem (SGEP)

Examples:

- SVD/PCA: $A = X^\top X, B = I$
- Graph Laplacian: L - Laplacian, $A = L, B = I$
- CCA, ICA, etc.

Top-k SGEP Game ($\mathbf{\Gamma}$-Eigengame)

- **Players**: k players $i \in \{1, \ldots, k\}$
- **Strategy**: generalized eigenvectors $v_i \in \mathcal{S}^{d-1}$ associated with top-k generalized eigenvalues λ_i
- **Utility**: i-th player utility conditioned on players $(j < i)$:

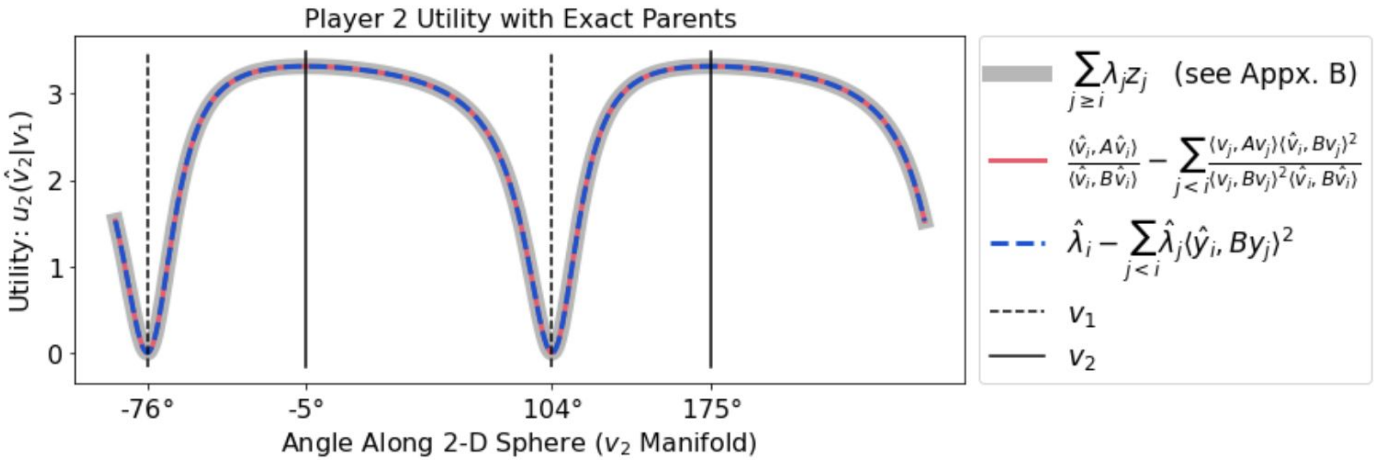
$$u_i\left(\hat{v}_i | \hat{v}_{j<i}\right) = \frac{\langle \hat{v}_i, A\hat{v}_i \rangle}{\langle \hat{v}_i, B\hat{v}_i \rangle} - \sum_{j<i} \frac{\langle \hat{v}_j, A\hat{v}_j \rangle \langle \hat{v}_i, B\hat{v}_j \rangle^2}{\langle \hat{v}_j, B\hat{v}_j \rangle^2 \langle \hat{v}_i, B\hat{v}_i \rangle}$$

$$= \hat{\lambda}_i - \sum_{j<i} \hat{\lambda}_j \langle \hat{y}_i, B\hat{y}_j \rangle,$$

where $\hat{y}_i = \frac{\hat{v}_i}{\|\hat{v}_i\|_B}$, $\hat{\lambda}_i = \frac{\langle \hat{v}_i, A\hat{v}_i \rangle}{\langle \hat{v}_i, B\hat{v}_i \rangle}$ (gen. Rayleigh quotient),
$\|z\|_B = \sqrt{\langle z, Bz \rangle}$

Utilities as periodic functions

Proposition 1 (Utility Shape). *Each player's utility is periodic in the angular deviation (θ) along the sphere. Its shape is sinusoidal, but with its angular axis (θ) smoothly deformed as a function of B. Most importantly, every local maximum is a global maximum.*



Deriving algorithm

Gradient of player's utility (up to scaling constant):

$$\frac{(\hat{v}_i^\top B\hat{v}_i)A\hat{v}_i - (\hat{v}_i^\top A\hat{v}_i)B\hat{v}_i}{\langle \hat{v}_i, B\hat{v}_i\rangle^2} - \sum_{j<i} \frac{\hat{\lambda}_j}{\langle \hat{v}_j, B\hat{v}_j\rangle}(\hat{v}_i^\top B\hat{v}_j)\frac{\left[\langle \hat{v}_i, B\hat{v}_i\rangle B\hat{v}_j - \langle \hat{v}_i, B\hat{v}_j\rangle B\hat{v}_i\right]}{\langle \hat{v}_i, B\hat{v}_i\rangle^2}.$$

(i) $\hat{\lambda}_j\langle \hat{v}_i, B\hat{v}_j\rangle = \langle \hat{v}_i, A\hat{v}_j\rangle$ if player i's parents match their true solutions, i.e., $\hat{v}_{j<i} = v_{j<i}$,

(ii) $\sqrt{\langle \hat{v}_j, B\hat{v}_j\rangle} = ||\hat{v}_j||_B$ is strictly positive and real-valued because $B \succ 0$,

Simplified update:

$$\tilde{\nabla}_i = \overbrace{(\hat{v}_i^\top B\hat{v}_i)A\hat{v}_i - (\hat{v}_i^\top A\hat{v}_i)B\hat{v}_i}^{\text{reward}} - \sum_{j<i} \overbrace{(\hat{v}_i^\top A\hat{y}_j)\left[\langle \hat{v}_i, B\hat{v}_i\rangle B\hat{y}_j - \langle \hat{v}_i, B\hat{y}_j\rangle B\hat{v}_i\right]}^{\text{penalty}}. \qquad (7)$$

Lemma 1 (Well-posed Utilities). *Given exact parents and assuming the top-k eigenvalues of* $B^{-1}A$ *are distinct and positive, the maximizer of player* i*'s utility is the unique generalized eigenvector* v_i *(up to sign, i.e.,* $-v_i$ *is also valid).*

Theorem 1 (Nash Property). *Assuming the top-k generalized eigenvalues of the generalized eigenvalue problem* $Av = \lambda Bv$ *are positive and distinct, their corresponding generalized eigenvectors form the unique, strict Nash equilibrium of* Γ*-EigenGame.*

Lemma 2. *The direction* $\tilde{\nabla}_i$ *defined in equation (7) is a steepest ascent direction on utility* $u_i(\hat{v}_i|\hat{v}_{j<i})$ *given exact parents* $\hat{v}_{j<i} = v_{j<i}$.

Theorem 2 (Deterministic / Full-batch Global Convergence). *Given a symmetric matrix* A *and symmetric positive definite matrix* B *where the top-k eigengaps of* $B^{-1}A$ *are positive along with a square-summable, not summable step size sequence* η_t *(e.g.,* $1/t$*), Algorithm 1 converges to the top-k generalized eigenvectors asymptotically* ($\lim_{T\to\infty}$) *with probability* 1.

Algorithm 1 Deterministic / Full-batch γ-EigenGame

Given: $A \in \mathbb{R}^{d \times d}$ and $B \in \mathbb{R}^{d \times d}$, step size sequence η_t, and number of iterations T.
$\hat{v}_i \sim \mathcal{S}^{d-1}$, i.e., $\hat{v}_i \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$; $\hat{v}_i \leftarrow \hat{v}_i / ||\hat{v}_i|$ for all i
for $t = 1 : T$ **do**
    parfor $i = 1 : k$ **do**
        $\hat{y}_j = \frac{\hat{v}_j}{\sqrt{\langle \hat{v}_j, B\hat{v}_j \rangle}}$
        $\texttt{rewards} \leftarrow (\hat{v}_i^\top B \hat{v}_i) A \hat{v}_i - (\hat{v}_i^\top A \hat{v}_i) B \hat{v}_i$
        $\texttt{penalties} \leftarrow \sum_{j<i} (\hat{v}_i^\top A \hat{y}_j) \big[\langle \hat{v}_i, B\hat{v}_i \rangle B \hat{y}_j - \langle \hat{v}_i, B\hat{y}_j \rangle B \hat{v}_i \big]$
        $\tilde{\nabla}_i \leftarrow \texttt{rewards} - \texttt{penalties}$
        $\hat{v}_i' \leftarrow \hat{v}_i + \eta_t \tilde{\nabla}_i$
        $\hat{v}_i \leftarrow \frac{\hat{v}_i'}{||\hat{v}_i'||}$
    end parfor
end for
return all $\hat{v}_i$

Algorithm 2 Stochastic γ-EigenGame

Given: paired data streams $X_t \in \mathbb{R}^{b \times d_x}$ and $Y_t \in \mathbb{R}^{b \times d_y}$, number of parallel machines M per player (minibatch size per machine $b' = \frac{b}{M}$), step size sequences η_t and γ_t, scalar ρ lower bounding $\sigma_{min}(B)$, and number of iterations T.
$\hat{v}_i \sim \mathcal{S}^{d-1}$, i.e., $\hat{v}_i \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$; $\hat{v}_i \leftarrow \hat{v}_i / ||\hat{v}_i|$ for all i
$[B\hat{v}]_i \leftarrow \hat{v}_i^0$ for all i
for $t = 1 : T$ **do**
  parfor $i = 1 : k$ **do**
    parfor $m = 1 : M$ **do**
      Construct A_{tm} and B_{tm} (*unbiased estimates using independent data batches)
      $\hat{y}_j = \frac{\hat{v}_j}{\sqrt{\max(\langle \hat{v}_j, [B\hat{v}]_j \rangle, \rho)}}$
      $[B\hat{y}]_j = \frac{[B\hat{v}]_j}{\sqrt{\max(\langle \hat{v}_j, [B\hat{v}]_j \rangle, \rho)}}$
      $\texttt{rewards} \leftarrow (\hat{v}_i^\top B_{tm} \hat{v}_i) A_{tm} \hat{v}_i - (\hat{v}_i^\top A_{tm} \hat{v}_i) B_{tm} \hat{v}_i$
      $\texttt{penalties} \leftarrow \sum_{j<i} (\hat{v}_i^\top A_{tm} \hat{y}_j) \big[\langle \hat{v}_i, B_{tm} \hat{v}_i \rangle [B\hat{y}]_j - \langle \hat{v}_i, [B\hat{y}]_j \rangle B_{tm} \hat{v}_i \big]$
      $\tilde{\nabla}_{im} \leftarrow \texttt{rewards} - \texttt{penalties}$
      $\nabla_{im}^{Bv} = (B_{tm} \hat{v}_i - [B\hat{v}]_i)$
    end parfor
    $\tilde{\nabla}_i \leftarrow \frac{1}{M} \sum_m [\tilde{\nabla}_{im}]$
    $\hat{v}_i' \leftarrow \hat{v}_i + \eta_t \tilde{\nabla}_i$
    $\hat{v}_i \leftarrow \frac{\hat{v}_i'}{||\hat{v}_i'||}$
    $\nabla_i^{Bv} \leftarrow \frac{1}{M} \sum_m [\nabla_{im}^{Bv}]$
    $[B\hat{v}]_i \leftarrow [B\hat{v}]_i + \gamma_t \nabla_i^{Bv}$
  end parfor
end for
return all $\hat{v}_i$

Naive Complexity:

$$\mathcal{O}(bdk^2)$$

Parallelized Complexity:

$$\mathcal{O}(dk)$$

Experiments

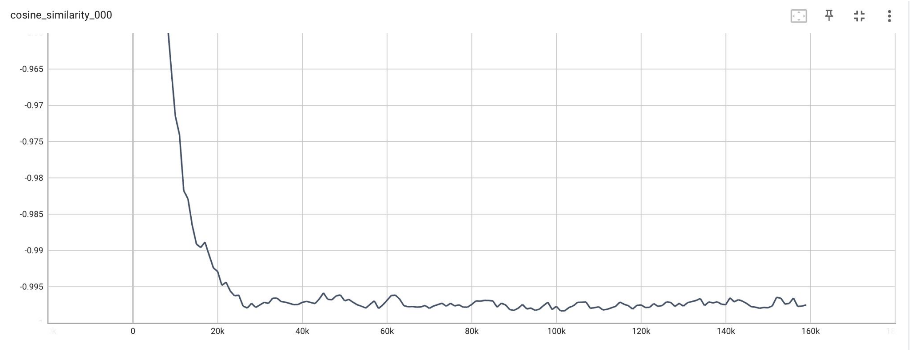
We run two kinds of experiments:

1. Basic synthetic data generated from normal distribution
2. Images from mnist dataset

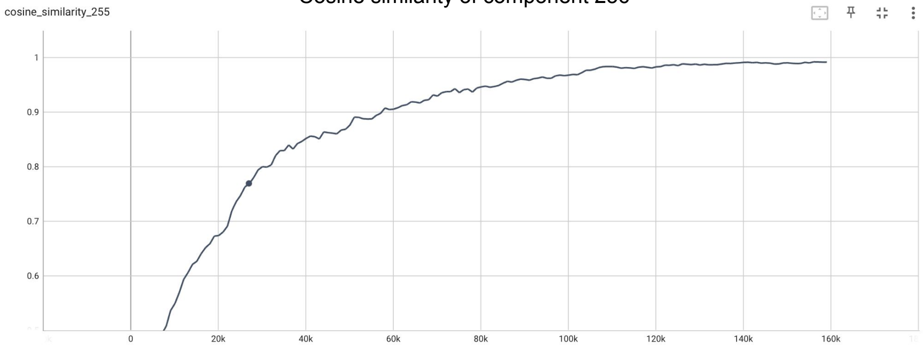
We also track cosine similarity as a loss function whenever possible

Synthetic data: convergence

Cosine similarity of component 1



Cosine similarity of component 256



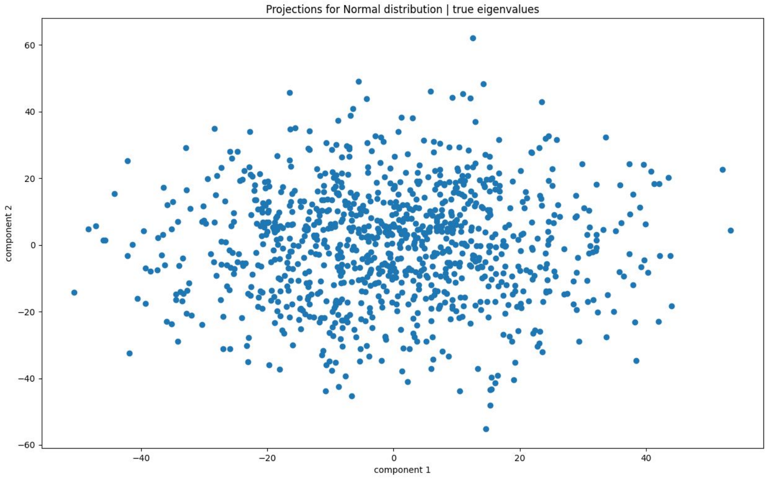
Valuable observations:

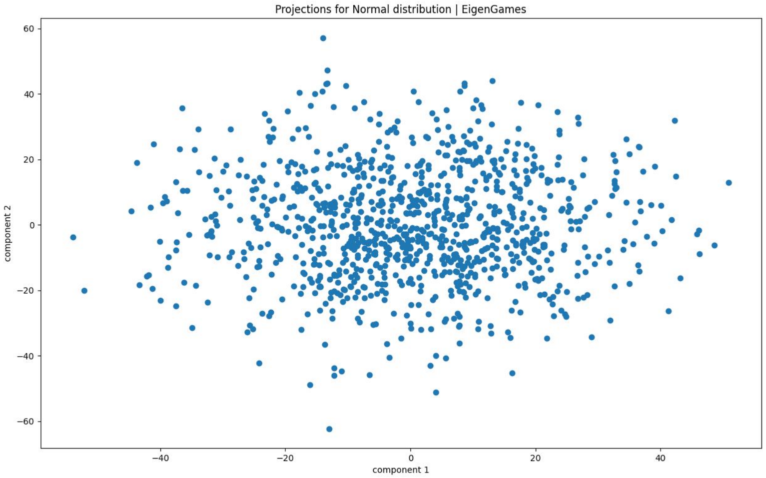
1. Algorithm is able to recover the true eigenvectors up to a sign

2. We can observe how the prediction of eigenvectors converge hierarchically

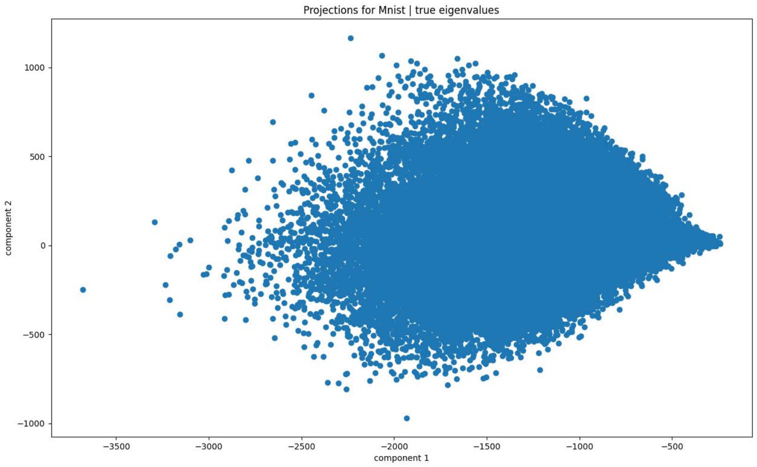
3. Algorithm takes time to converge. Therefore, there is no advantage on running it on simple datasets that can fit into a memory

Synthetic data: projections





Mnist projections



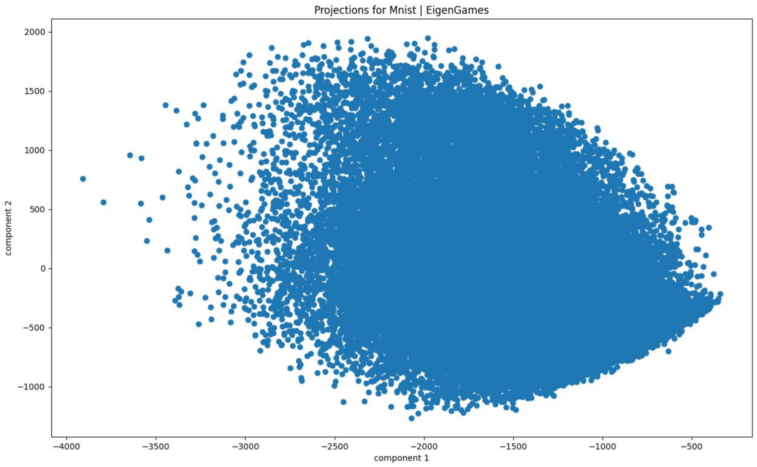
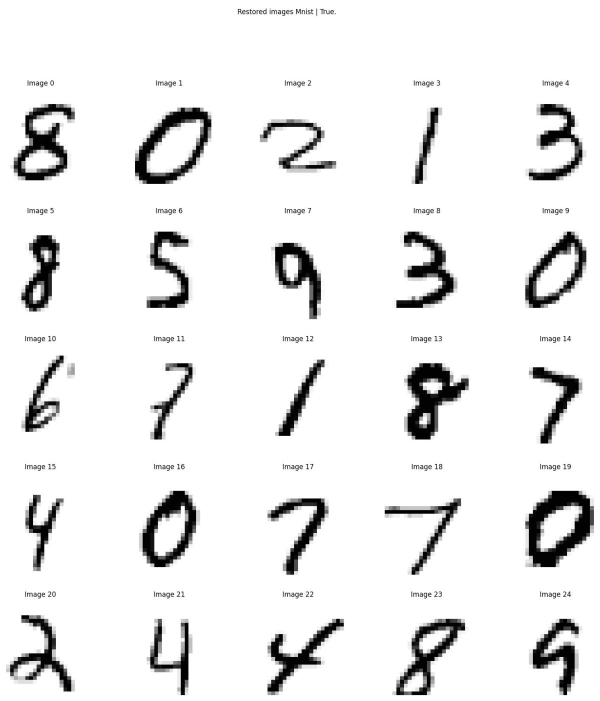


Image reconstructions using EigenGames PCA



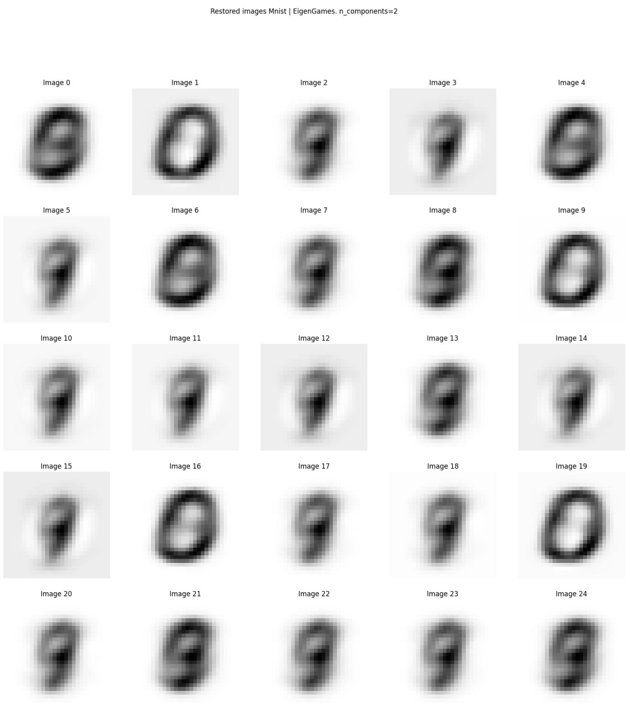


Image reconstructions using EigenGames PCA

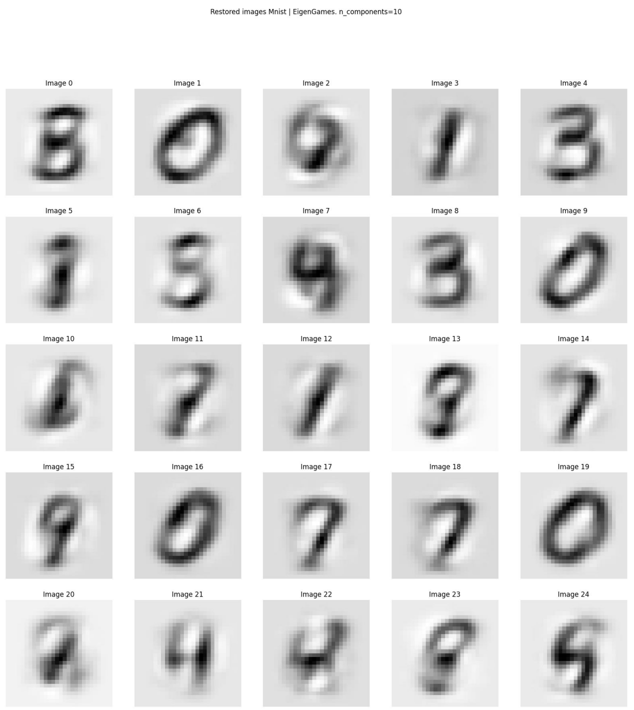
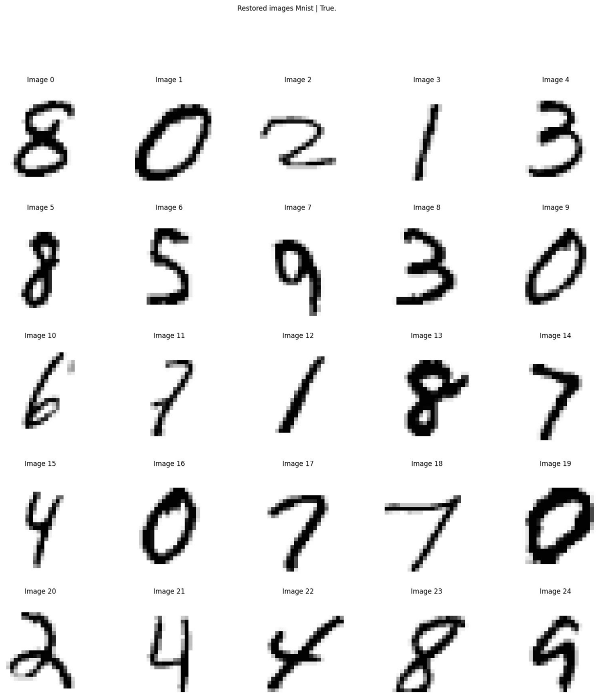


Image reconstructions using EigenGames PCA



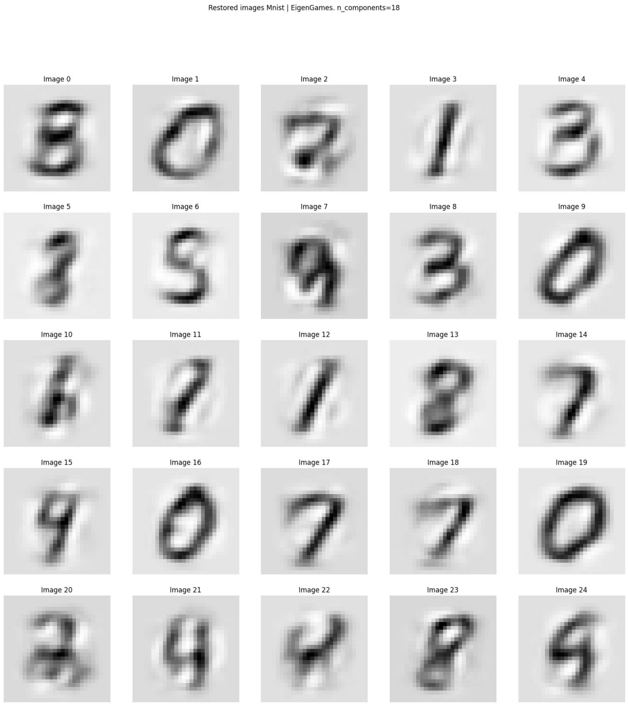


Image reconstructions using EigenGames PCA

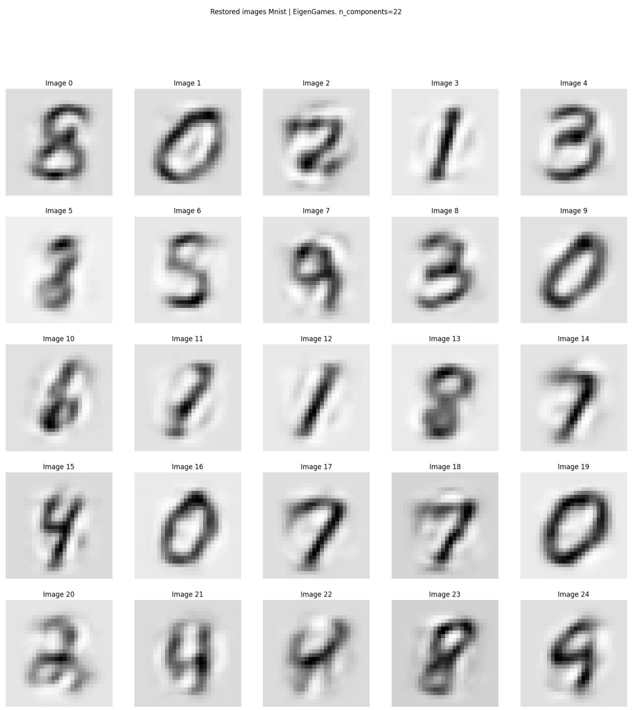
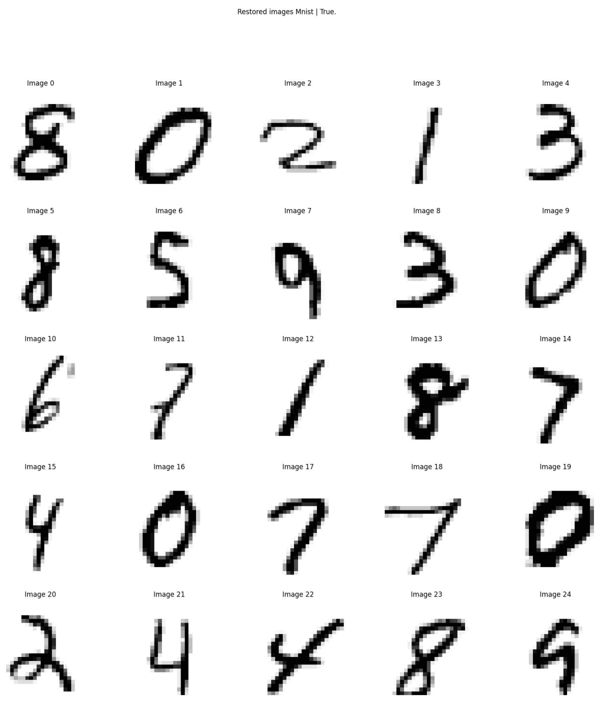


Image reconstructions using EigenGames PCA



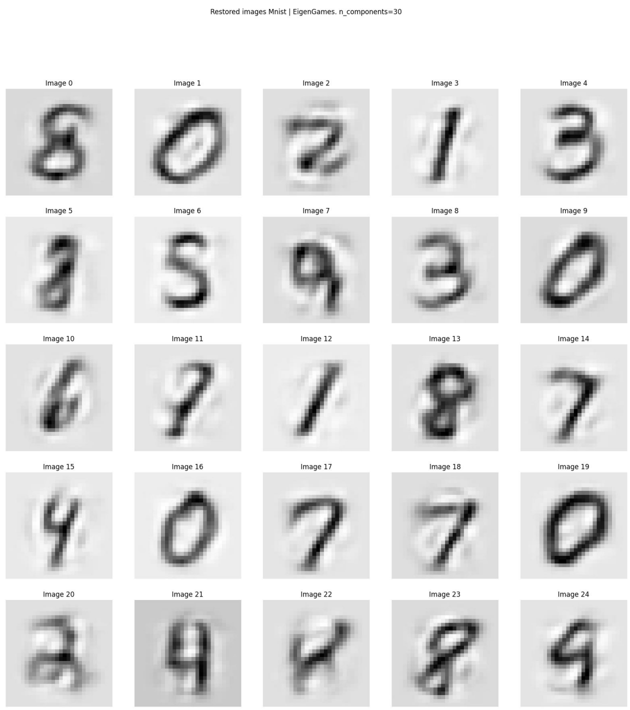
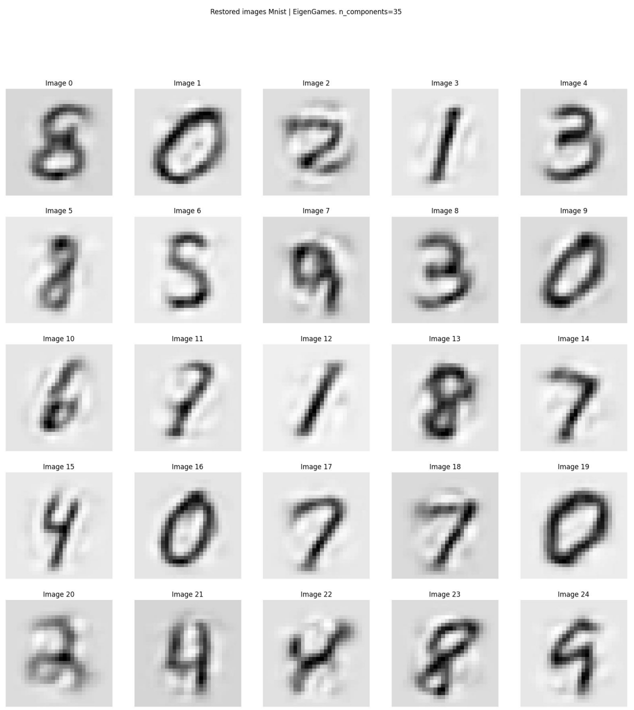


Image reconstructions using EigenGames PCA



References

- [Gemp2021] EigenGame: PCA as a Nash Equilibrium, https://arxiv.org/abs/2010.00554
- [Gemp2022] EigenGame Unloaded: When Playing Games is Better Than Optimizing, https://arxiv.org/abs/2102.04152
- [Gemp2023] The Symmetric Generalized Eigenvalue Problem as a Nash Equilibrium, https://arxiv.org/abs/2206.04993