SCIENCE HUB FOR HUMANITY
AND ARTIFICIAL INTELLIGENCE
UCLA amazon

UCLA Samueli School of Engineering

Insights from GAN Training with Kernel Discriminators

Evan Becker

01/25/2024

This talk includes joint work with:
Parthe Pandit (UCSD+IIT Bombay),
Sundeep Rangan (NYU),
Alyson K. Fletcher (UCLA)

Motivation

Problem: Generative Adversarial Networks (GANs) are popular but hard to train

- Alternating min-max optimization for a non-convex objective is not fully understood

thispersondoesnotexist.com

Contribution: Simple yet expressive framework to analyze convergence and failure modes

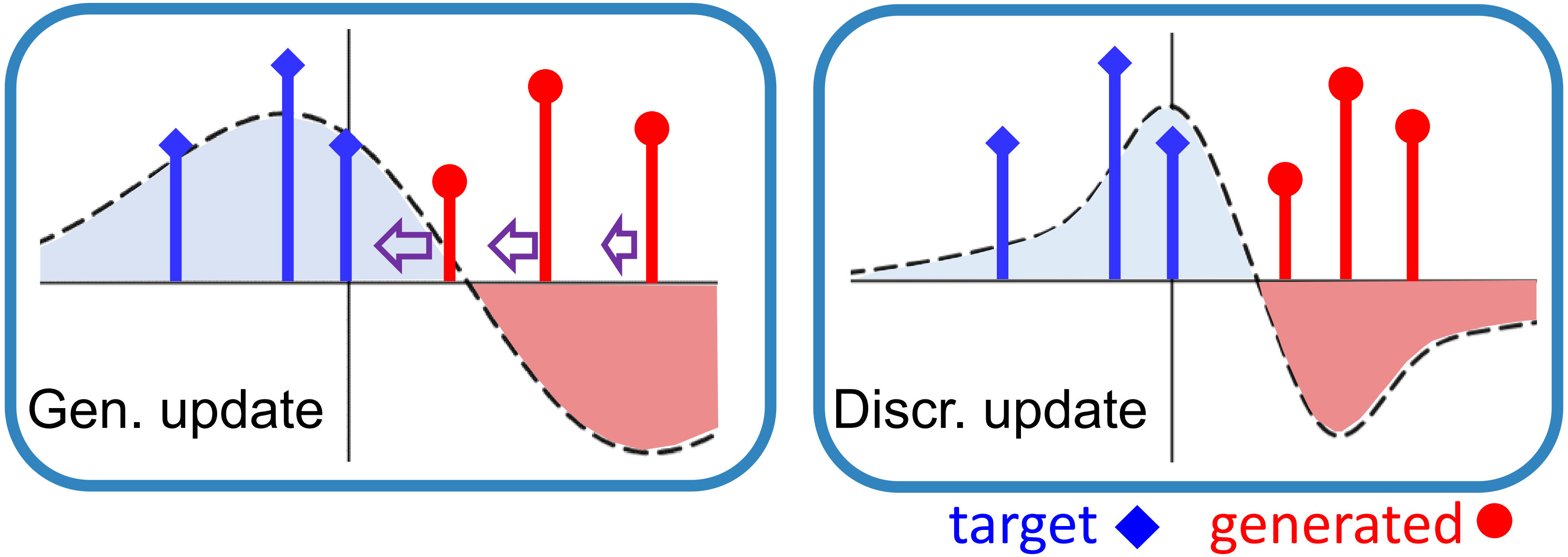
- MMD-GAN objective trained with gradient descent ascent
- Generator is unconstrained, discriminator is a kernel model

Problem setup

Target and Generated Distributions: Directly parameterized by set of points (δ Dirac delta function):

$$\mathbb{P}_r(\boldsymbol{x}) = \sum_{i=1}^{N_r} p_i \delta(\boldsymbol{x} - \boldsymbol{x}_i), \quad \mathbb{P}_g(\boldsymbol{x}) = \sum_{j=1}^{N_g} \widetilde{p}_j \delta(\boldsymbol{x} - \widetilde{\boldsymbol{x}}_j),$$

Discriminator: $f : \mathcal{X} \to \mathbb{R}$ from a reproducing kernel hilbert space (RKHS) $\mathcal{H}_K$ with positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$.

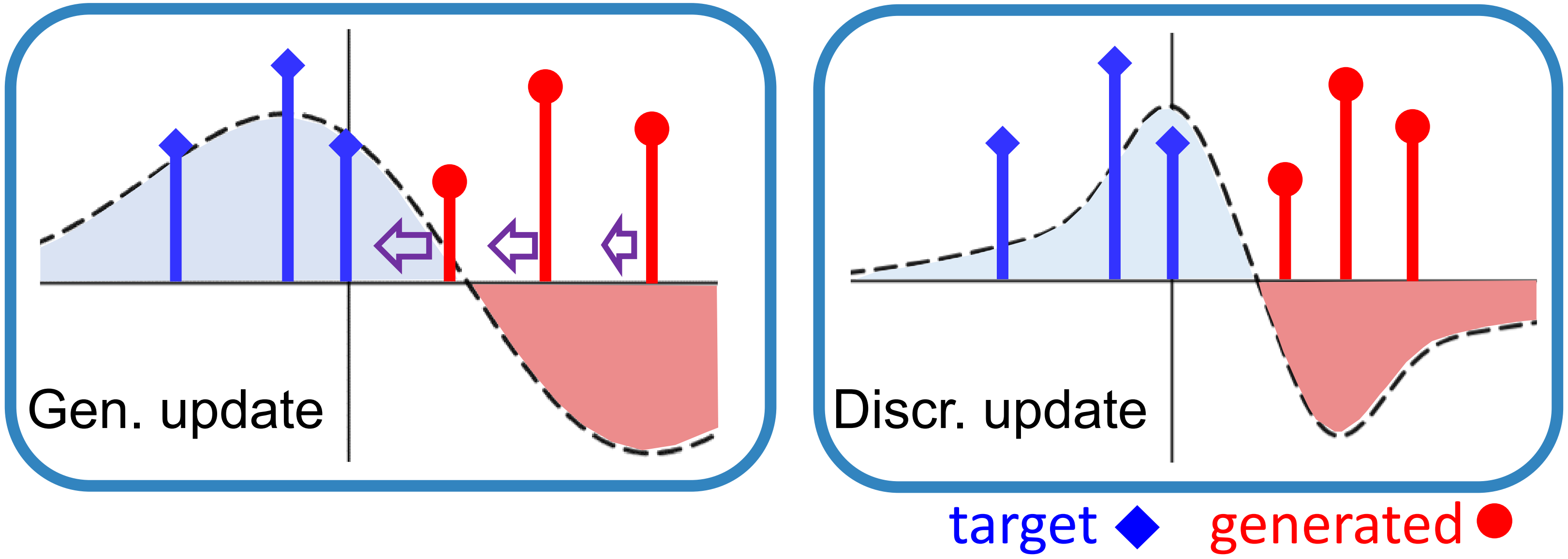


Problem setup

Optimization: $\min_{\widetilde{\boldsymbol{X}}} \max_{f \in \mathcal{H}} \mathcal{L}(f, \widetilde{\boldsymbol{X}})$ with loss defined as

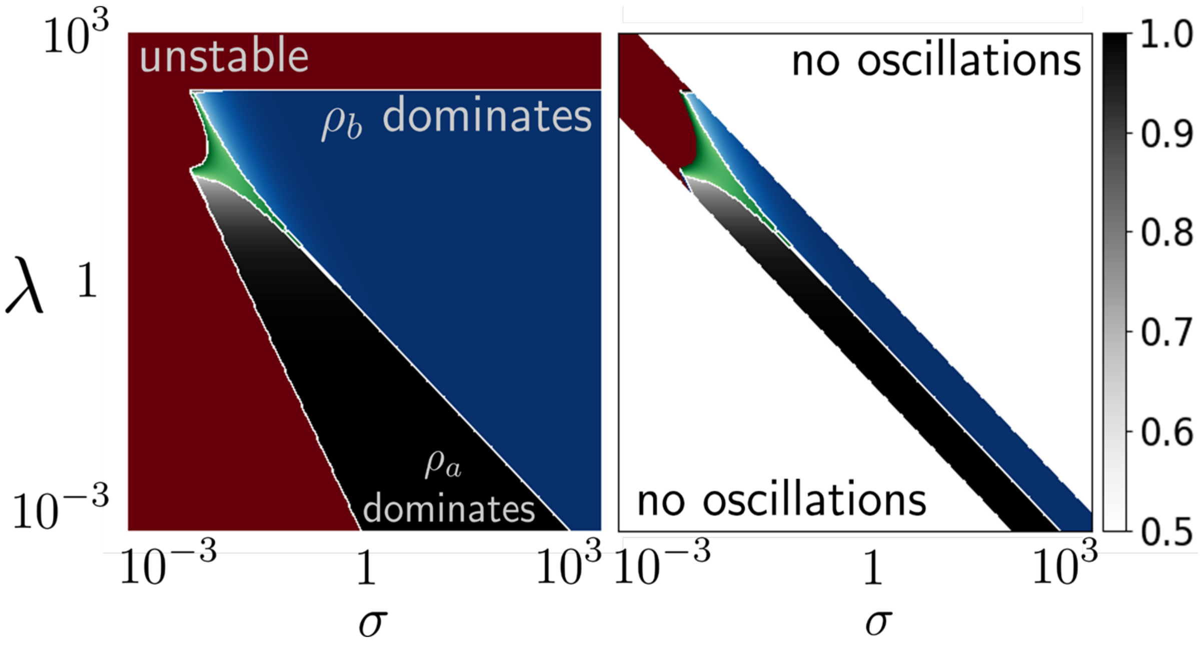
$$\mathcal{L}(f, \widetilde{\boldsymbol{X}}) := \sum_{i=1}^{N_r} p_i f(x_i) - \sum_{i=1}^{N_g} \widetilde{p}_i f(\widetilde{x}_j) - \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2 .$$

Note: Just maximizing the objective over $f \in \mathcal{H}$ results in the maximum mean discrepancy (MMD) between distributions, which is a common two-sample test statistic



First-Order Analysis

Idea: Look at some local region around each true point $\boldsymbol{x}_i$



Dominating eigenvalues when using an RBF kernel discriminator

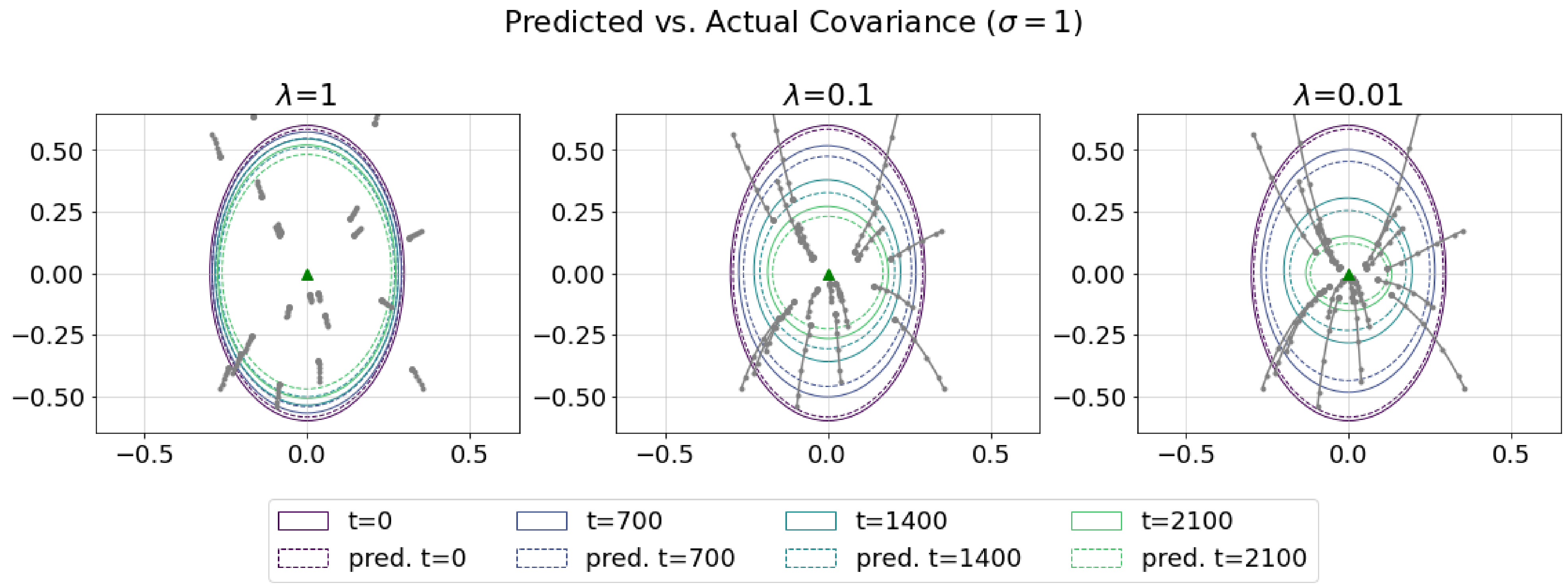
Insight 1 (Becker et al. 22): Existence of good local minima (exact convergence) and bad (mode collapse).

Insight 2 (Becker et al. 23): Analyze the *rate* of convergence by looking at eigenvalue functions. Can prescribe hyperparameters to achieve fastest local convergence.

hyperparameters: λ regularization, σ kernel width, η_g, η_d learning rates, $\Delta_i := p_i - \sum_{j \in N_i} \widetilde{p}_j$ local probability mass difference

Second-Order Analysis

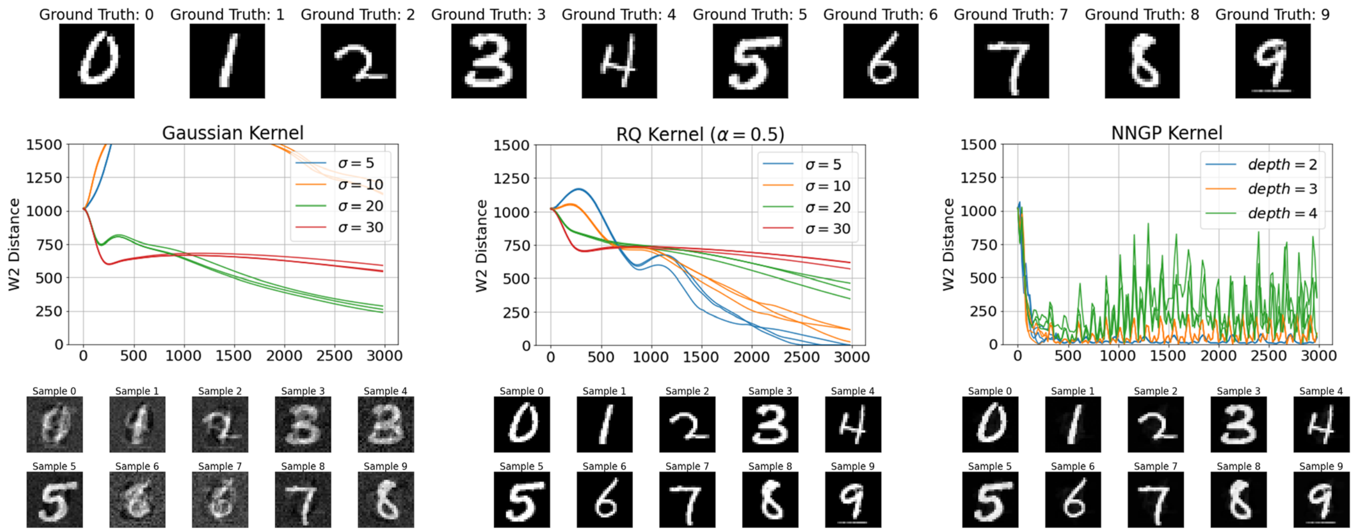
Idea: Look at the first two moments (mean and variance) of the generated distribution



Insight 3: Smaller kernel width σ increases 'momentum' of convergence in second order analysis (stronger effect than linearized analysis predicts)

hyperparameters: λ regularization, σ kernel width, η_g, η_d learning rates, $\Delta_i := p_i - \sum_{j \in N_i} \widetilde{p}_j$ local probability mass difference

Key takeaways:



Why do we care?

- Neural networks can be thought of as kernel machines whose kernel shape evolves over time (evolves very little in NTK regime)
- Reducing effective kernel width of the discriminator during training promotes fast convergence to good minima, and is already done heuristically (Karras et. al 2018)!
- Min-max optimization is used outside of GANs, and techniques used here could be used to study other systems

Thank you!

Questions?

Email: evbecker@ucla.edu