FROM OPTIMIZATION TO EQUILIBRATION: UNDERSTANDING AN EMERGING PARADIGM IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

A Dissertation Presented

by

IAN GEMP

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2019

College of Information and Computer Sciences

FROM OPTIMIZATION TO EQUILIBRATION: UNDERSTANDING AN EMERGING PARADIGM IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

A Dissertation Presented

by

IAN GEMP

Approved as to style and content by:

Sridhar Mahadevan, Chair

Phil Thomas, Member

Daniel Sheldon, Member

Mario Parente, Member

James Allan, Chair of the Faculty
College of Information and Computer Sciences

ACKNOWLEDGMENTS

I would like to begin by thanking my advisor, Sridhar Mahadevan. Sridhar introduced me to Variational Inequality theory, providing a framework for formalizing my ideas and enabling me to continue research on topics I find most interesting. I would also like to thank my thesis committee members, Phil Thomas, Dan Sheldon, and Mario Parente. I specifically want to thank Dan and Phil for taking the time to discuss my research and to provide me with support and advice. I also want to thank Mario and Darby Dyar for their mentorship during my research assistantships with them throughout the course of my graduate studies.

Thank you to the faculty and staff of the College of Information and Computer Sciences, who provided a welcoming and productive environment for my graduate studies. Special thanks are due to Susan Overstreet and Leeanne Leclerc, without whose support and patience I would be lost. Thanks also to the current and former members of the Autonomous Learning Lab, an excellent group of graduate students that I am proud to have as colleagues and friends.

Finally, and most importantly, I want to thank my family for their unfailing encouragement and support. More than anyone, you know and shared in the trials and tribulations I had to overcome to get here. Thank you.

ABSTRACT

FROM OPTIMIZATION TO EQUILIBRATION: UNDERSTANDING AN EMERGING PARADIGM IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

MAY 2019

IAN GEMP

B.Sc., UNIVERSITY OF NORTHWESTERN

B.Sc., UNIVERSITY OF NORTHWESTERN

M.Sc., UNIVERSITY OF NORTHWESTERN

M.Sc., UNIVERSITY OF MASSACHUSETTS, AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Sridhar Mahadevan

Many existing machine learning (ML) algorithms cannot be viewed as gradient descent on some single objective. The solution trajectories taken by these algorithms naturally exhibit rotation, sometimes forming cycles, a behavior that is not expected with (full-batch) gradient descent. However, these algorithms can be viewed more generally as solving for the equilibrium of a game with possibly multiple competing objectives. Moreover, some recent ML models, specifically generative adversarial networks (GANs) and its variants, are now explicitly formulated as equilibrium problems. Equilibrium problems present challenges beyond those encountered in optimization

such as limit-cycles and chaotic attractors and are able to abstract away some of the difficulties encountered when training models like GANs.

In this thesis, I aim to advance our understanding of equilibrium problems so as to improve state-of-the-art in GANs and related domains. In the following chapters, I will present work on

1. designing a no-regret framework for solving monotone equilibrium problems in online or streaming settings (with applications to Reinforcement Learning),

2. ensuring convergence when training a GAN to fit a normal distribution to data by *Crossing-the-Curl*,

3. improving state-of-the-art image generation with techniques derived from theory,

4. and borrowing tools from dynamical systems theory for analyzing the complex dynamics of GAN training.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

INTRODUCTION

Artificial Intelligence (AI) focuses on the design of agents that act rationally. The Maximum Expected Utility (MEU) principle formalizes the behavior of a rational agent as the solution to an optimization problem: $\max_{\text{action}} \mathbb{E}[U(\text{action})]$. This principle has pulled optimization to the center of attention in AI and Machine Learning (ML), however, a new paradigm is emerging. Many existing algorithms such as those in Reinforcement Learning (RL) or inference in graphical models can be viewed as solving for an equilibrium rather than an optimum. Moreover, some recent ML models, specifically generative adversarial networks (GANs) and its variants, are now explicitly formulated as equilibrium problems.

Equilibrium problems present their own set of unique difficulties. One common difficulty of equilibrium problems not shared by optimization is the existence of cyclic or oscillatory behavior during the solution process. Properties like these pose real challenges for ML researchers tackling an equilibrium approach to ML. In fact, the domains described above all exhibit important, practical problems caused by the nature of equilibration: "divergence of...TD" [35]; "One of the main problems with loopy belief propagation is nonconvergence...often due to oscillations" [60]; "[GANs are] known to be notoriously hard to train" [74].

Optimization has been studied within the context of ML for decades, leading to new algorithms and even the study of new problems such as Online Optimization. Traditionally, equilibration has not seen the same attention, however, it has risen to the forefront recently with the advent of GANs. GANs have highlighted our lack of theoretical and empirical understanding of these problems. In order to improve

upon the performance of current equilibration-based models, we must elevate our understanding of equilibration, especially within the context of ML.

In my thesis, I aim at both theoretical and empirical advances in equilibration within the context of ML. Theoretical advances necessarily focus on simpler domains where analysis is tractable. The hope is that advances here will transfer to useful heuristics in more complex domains. I also develop tools for analyzing more complex domains—these tools should be useful, especially when intuitions derived from a simpler theoretical understanding fail.

The first contribution of the thesis is ***Online Monotone Equilibration* (OME)**, a framework for studying monotone equilibrium problems in an online (possibly adversarial) scenario. As comparison, the Online Optimization framework can be used for studying optimization-based machine learning models that are expected to learn *as* they consume data from a (possibly adversarial) data source. This is particularly useful in this era of *Big Data* where models must process data in a streaming fashion to ensure realistic training times. An analogous framework for equilibration is lacking. The OME framework subsumes the well known Online Convex Optimization framework and defines a notion of *regret* that applies to both optimization and equilibrium problems. A close inspection of this framework motivates an algorithm presented in the second chapter.

The second contribution of the thesis is an analysis of the ***Linear-Quadratic GAN* (LQ-GAN)**, as well as an algorithm with convergence guarantees for equilibrating this model. The Linear-Quadratic GAN has recently been proposed as an important test problem for equilibration. Solving the LQ-GAN is equivalent to fitting a multivariate-Gaussian to data, making this a fundamental generative modeling problem as well. Despite the simplicity of the task, this model is deceptively complex. Technically speaking, the corresponding equilibrium problem is not even quasi-monotone. Despite this challenge, our analysis reveals that there exists an el-

egant solution to this problem that may generalize to more complex domains; we call the successful technique ***Crossing-the-Curl***. A specific aspect of our solution technique supports a practical approach for training neural network based GANs that we explore in the next chapter.

The third contribution of the thesis is ***Generative Multi-Adversarial Networks*** **(GMAN)**, a framework that extends GANs to multiple discriminators. A GAN is modeled as a two-player minimax game, but equilibration is more generally studied with N players. We show that introducing more discriminators into the standard GAN framework reduces variance of the minimax objective, improves the quality of the resulting samples that are generated, and accelerates convergence of the GAN to a steady-state minimax loss. In the final chapter, we examine the dynamics at the end of training and study whether convergence of the loss implies convergence of the weights.

The fourth contribution of the thesis applies an analytical tool, ***Lyapunov Exponent Computation*** **(LEC)**, to very large, stochastic equilibrium problems. Lyapunov exponents (LEs) are vectors that concisely summarize the behavior of a dynamical system (DS). For instance, if an LE associated with a certain region of a DS contains N leading zeros, all initial states in that region of the DS converge to an N-torus. Information like this could prove valuable to judging the tractability of GAN variants, studying weight initialization, and more. The number of weights learned in a GAN is in the thousands and millions. Lyapunov Exponent (LE) computation is typically used to study the dynamics of deterministic physical systems (i.e., 3-D). Extending LEC to GANs requires streamlining traditional LEC by way of approximations.

The chapters are organized in order of the contributions listed above. Each chapter begins by discussing the purpose (importance) of the research topic and then presents

work towards the intended contribution. The chapter then concludes with a summary of results.

CHAPTER 1

TECHNICAL BACKGROUND AND MOTIVATING PROBLEMS

In this chapter, we will provide some useful tools and results from convex analysis, online convex optimization, game theory, GANs, variational inequalities, and monotone operator theory.

1.1 Optimization

We will now formalize and abstract the meaning of a continuous optimization problem. Discrete or combinatorial optimization lies outside the scope of this thesis. Let $f : \mathcal{X} \to \mathbb{R}$ be a function that maps each input in the set $\mathcal{X}$ to a real number. The set $\mathcal{X}$ may be $\mathbb{R}^n$ or it may be a subset of $\mathbb{R}^n$. In the case that $\mathcal{X} \subset \mathbb{R}^n$, the optimization problem is called a constrained optimization problem. The optimization problem is to find the $x \in \mathcal{X}$ that minimizes $f(x)$, written as

$$\min_{x \in \mathcal{X}} f(x), \tag{1.1}$$

where $f(x)$ is called the objective function. Note that we can equivalently formulate this problem as a maximization problem with

$$\max_{x \in \mathcal{X}} -f(x). \tag{1.2}$$

We denote any x that minimizes $f(x)$ as x^*, also written as

$$x^* \in \arg\min_{x \in \mathcal{X}} f(x). \tag{1.3}$$

As an example, consider the ordinary least squares method for linear regression. This is one of the most common and well studied techniques in discriminative prediction. Assume we have a dataset of n input-output pairs, (x, y), where $x \in \mathbb{R}^{m+1}$ is a vector augmented with a 1, and $y \in \mathbb{R}^p$ is also a vector. The goal is to learn the parameters of a function, $f(x) = Ax$, that minimize the sum of squared errors between the true outputs and the outputs predicted by the function. Let X be a matrix whose columns consist of the x's and let Y be a matrix whose columns consist of the y's. Let $A \in \mathbb{R}^{p\times(m+1)}$ be a matrix containing the parameters of $f(x)$. Note that we can consider A to be the vector $a \in \mathbb{R}^{pm+p}$ reshaped. Then we can formulate our goal of fitting a line to the data as the following optimization problem

$$\min_{a\in\mathbb{R}^{pm+p}} ||Y - AX||_2^2 = \min_{A\in\mathbb{R}^{p\times(m+1)}} \sum_{i=1}^{n}\sum_{j=1}^{p}\left(Y_{ji} - \sum_{k=1}^{m+1} A_{jk}X_{ki}\right)^2 \tag{1.4}$$

where $||\cdot||$ represents a norm. The squared Euclidean norm is given by $||\cdot||_2^2$. We typically write this as an optimization over A noting the equivalence between a and A:

$$\min_{A\in\mathbb{R}^{p\times(m+1)}} ||Y - AX||_2^2. \tag{1.5}$$

1.1.1 Convex Optimization

The example above actually has two very important properties. The first is that the set $\mathcal{X}$ is convex. A set $\mathcal{X}$ is convex if and only if for every $x_0 \in \mathcal{X}$, $x_f \in \mathcal{X}$, and $t \in [0, 1]$,

$$(1 - t)x_0 + tx_f \in \mathcal{X}. \tag{1.6}$$

In other words, $\mathcal{X}$ is convex if any line segment connecting any two vectors in $\mathcal{X}$ lies completely in $\mathcal{X}$.

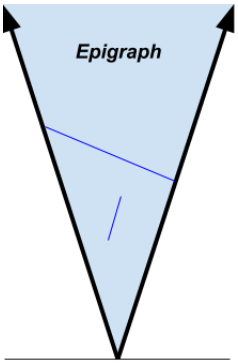


Figure 1.1: This figure depicts the convex function $f(x) = |x|$ and its epigraph: the set defined by the space above the function. Note that any line segment connecting two points in the epigraph is wholly contained in the epigraph—therefore the epigraph of $|x|$ is convex, therefore $f(x)$ is convex.

The second important property is that the objective function is convex. The convexity of a function can actually be defined similarly to above using its epigraph (see Figure 1.1), but we will choose an alternate definition to better match other properties that we will present later in this proposal. Before we define convexity for a function, we first define its subdifferential. The subdifferential of a convex function, $f : \mathcal{X} \to \mathbb{R}$, at x, denoted $\partial f(x)$, is the set of all subgradients at x, i.e., $\partial f(x) = \{z : \forall x' \in \mathcal{X},\ \langle z, x' - x \rangle \leq f(x') - f(x)\}$; in other words, a first order Taylor series expansion about x using $z \in \partial f(x)$ as the first derivative ensures that the Taylor series approximation does not overestimate the function anywhere else. Finally, a function, f, is *convex* if

$$\begin{aligned} &\forall x \in \mathcal{X}, x' \in \mathcal{X}, z \in \partial f(x), z' \in \partial f(x'), \\ &\langle z - z', x - x' \rangle \geq 0. \end{aligned} \tag{1.7}$$

By appealing to finite difference approximations, we can also view this as requiring the function to have positive semi-definite Hessian, $H \succeq 0$. Let $v = x - x'$ and note that a Hessian-vector product can be approximated as $H \cdot v \approx z - z'$ where x, x', z, and z' are defined as before. Then $\langle z - z', x - x' \rangle \geq 0$ is implicitly requiring

$v^\top Hv \geq 0$ for all v but is well defined even when H is not available, i.e., f is not twice differentiable.

A necessary first order condition for optimality of x^* is

$$\exists z^* \in \partial f(x^*) \text{ such that } \langle z^*, x' - x^* \rangle \geq 0 \quad \forall x' \in \mathcal{X}. \tag{1.8}$$

For convex optimization problems, this is also a sufficient condition. If the subdifferential of f contains only a single subgradient at each x, $\partial f(x)$ is more commonly referred to as the gradient, written $\nabla f(x)$.

Convex optimization problems are of particular interest because there exist many techniques to solve them that come with convergence guarantees. One of the most popular algorithms for solving continuous convex optimization problems is projected subgradient descent,

Algorithm 1 Projected Subgradient Descent (PSGD)

input: A scalar learning rate schedule, e.g., $\eta = \frac{1}{\sqrt{k}}$
$x_1 = 0$
for all $k = 1, 2, \ldots$ **do**
  $x_{k+1} = P_{\mathcal{X}}(x_k - \eta z_k)$ where $z_k \in \partial f(x_k)$
end for

where $P_{\mathcal{X}}(\xi)$ denotes the projection of ξ onto the set $\mathcal{X}$. Note that the projection operation is also defined as an optimization problem

$$P_{\mathcal{X}}(\xi) = \arg\min_{x \in \mathcal{X}} ||\xi - x||. \tag{1.9}$$

Projected subgradient descent has a convergence rate of $\mathcal{O}(\frac{1}{\sqrt{k}})$. This means that $f(x_k^{\text{best}}) - f(x^*)$ is $\mathcal{O}(\frac{1}{\sqrt{k}})$, which implies that we need $\mathcal{O}(\frac{1}{\epsilon^2})$ iterations to achieve $f(x_k^{\text{best}}) - f(x^*) \leq \epsilon$ where x_k^{best} is the iterate with the lowest error so far, i.e., $x_k^{\text{best}} = x_{k^*}$ where $k^* = \arg\min_{\{1,\ldots,k\}} f(x_k)$.

When the projected subgradient algorithm is run on an unconstrained, differentiable optimization problem, i.e., without the projection operator and using gradients instead of subgradients, it is more commonly known as gradient descent. We will refer to the projected subgradient algorithm more concisely as gradient descent from now on. We refer the reader to the book by Boyd and Vandenberghe [2004] for a comprehensive review of convex optimization.

1.1.2 Online Optimization

Online optimization is important for training machine learning models on streaming datasets when we require the model to begin making predictions before seeing all of the data. Fore example, consider the least squares linear regression problem again,

$$\min_{A \in \mathbb{R}^{p \times (m+1)}} \sum_{i=1}^{n} \sum_{j=1}^{m} (\sum_{k=1}^{m+1} A_{jk} X_{ki} - Y_{ji})^2. \tag{1.10}$$

We can rewrite this more generally as

$$\min_{x \in \mathcal{X}} R(x) + \sum_{i=1}^{n} f_i(x). \tag{1.11}$$

where $R(x) = 0$ in the above least squares problem. This particular form occurs throughout machine learning with datasets whose samples are assumed to be independent and identically distributed (i.i.d.). We typically want to minimize the sum of errors for each data point in the dataset (just like least squares linear regression). In this case, $f_i(x)$ is typically $f(x, data_i)$ where x represents the parameters (e.g., A) and $data_i$ represents an (x, y) pair. The function, $R(x)$, is called a regularizer and was introduced to bias the solution x^* towards an x with properties more desirable to the specific problem at hand. We will consider a natural choice for $R(x)$ later.

Now, assume we solve the least squares regression problem by minimizing the above objective over n samples. We then decide to employ our linear regressor in

production to start taking advantage of its predictive power. However, we continue to receive more data. Should we choose to re-solve our least squares problem once we meet some criteria such as "our dataset doubled in size" or "our predictions no longer seem accurate"? Online optimization presents a framework for tackling this dilemma and suggests a solution where we can train our regressor by simply adjusting its parameters a small amount as every new (x, y) pair is observed.

If each f_i is convex and $R(x) = \frac{1}{2\eta}||x||_2^2$, it can be shown that there exists a natural extension of gradient descent that is actually equivalent to resolving the least squares linear regression problem on the entire dataset as the number of samples goes to infinity. This framework is known as Online Convex Optimization, and it generalizes beyond least squares linear regression to any sequence of convex losses. Online Convex Optimization is presented below along with the algorithm described, online gradient descent (note that in this case, x denotes the parameters being learned).

Framework 1 Online Convex Optimization (OCO)

input: A convex set $\mathcal{X}$
for all $t = 1, 2, \ldots$ **do**
  predict a vector $x_t \in \mathcal{X}$
  receive a convex loss function $f_t : \mathcal{X} \to \mathbb{R}$
  suffer loss $f_t(x_t)$
end for

Algorithm 2 Online Gradient Descent (OGD)

input: A scalar learning rate $\eta > 0$
$x_1 = 0$
for all $t = 1, 2, \ldots$ **do**
  $x_{t+1} = x_t - \eta z_t$ where $z_t \in \partial f_t(x_t)$
end for

In the context of online learning, we often measure performance with *regret*. Regret measures how much worse off we are by making our predictions online rather than

waiting for all the data to arrive and computing the best parameters offline in batch form. Regret is defined as

$$regret = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x) = \sum_{t=1}^{T} f_t(x_t) - f_t(x^*). \tag{1.12}$$

We suggest the book by Shalev-Shwartz [2011] as a reference for online convex optimization.

1.2 Equilibration and Game Theory

As discussed in the introduction, not all machine learning problems are formulated and solved as optimization problems. We mentioned two in which solving a machine learning problem amounted to finding an equilibrium point: reinforcement learning and generative adversarial networks. We now formalize the notion of an equilibrium within the framework of games.

We will begin at the intersection of optimization and game theory—1-player games. In a 1-player game, player 1's goal is to minimize its loss function, $f^{(1)}(x^{(1)})$ by adjusting the variables under its control, $x^{(1)}$. Player 1 can solve this as an optimization problem

$$\min_{x^{(1)} \in \mathcal{X}^{(1)}} f^{(1)}(x^{(1)}). \tag{1.13}$$

If we generalize this to an N-player game, player 1's loss may now additionally depend on other players' variables, $f^{(1)}(x^{(1)})$ becomes $f^{(1)}(x^{(1)}, \ldots, x^{(N)})$. We write the vector containing all N players' variables concisely as x. In this case, x must belong to the product space of the N players' sets, i.e., $\mathcal{X} = \mathcal{X}^{(1)} \times \cdots \times \mathcal{X}^{(N)}$. We denote the vector containing all the player variables except player i as $x^{(-i)}$. Now we can say

that, given all other players' variables are fixed, the goal of each player i is to solve the optimization problem

$$\min_{x^{(i)} \in \mathcal{X}^{(i)}} f^{(i)}(x). \tag{1.14}$$

This is where optimization falls short in describing the problem. If player 1 solves this problem with all other players fixed, the solution will likely be one that exploits all the other players. For example, player 2 may be very displeased with the solution found by player 1 and choose to fix player 1's and all other players' variables and find a solution with lower $f^{(2)}$. Essentially, the problem with games is pleasing all the players simultaneously. One such notion that captures this ideal is the Nash equilibrium. The vector of player variables, x^*, constitutes a Nash equilibrium [83] if no single player i can reduce their loss by deviating from $x^{*(i)}$ with all other player variables fixed. More formally, let $\tilde{x}^{(i)}$ be the vector of player strategies where player i plays any $x^{(i)} \in \mathcal{X}^{(i)}$ and player $j \neq i$ plays $x^{*(j)}$. Then the vector of player variables, $x^* \in \mathcal{X}$, is a Nash equilibrium if for all i and all $x^{(i)} \in \mathcal{X}^{(i)}$, $f^{(i)}(\tilde{x}^{(i)}) \geq f^{(i)}(x^*)$. We denote the set of all Nash equilibria by $\mathcal{X}^*$.

1.2.1 Generative Adversarial Networks

As an example application of Nash equilibria to ML, consider generative adversarial networks (GANs). The goal of a GAN is to learn a function capable of transforming a noisy random variable, z, into a distribution that matches the true distribution of some data source, $p_{data}(x)$. Typically the transformation function used is a neural network. The problem of learning this function is framed as a two-player minimax game, a special type of game where $f^{(1)} = -f^{(2)}$. In this game, the transformation function is referred to as the data generator, G, and the other player is referred to as the discriminator, D. The role of the discriminator is to predict whether or not the data generated by G came from $p_{data}(x)$. The game can be written as follows

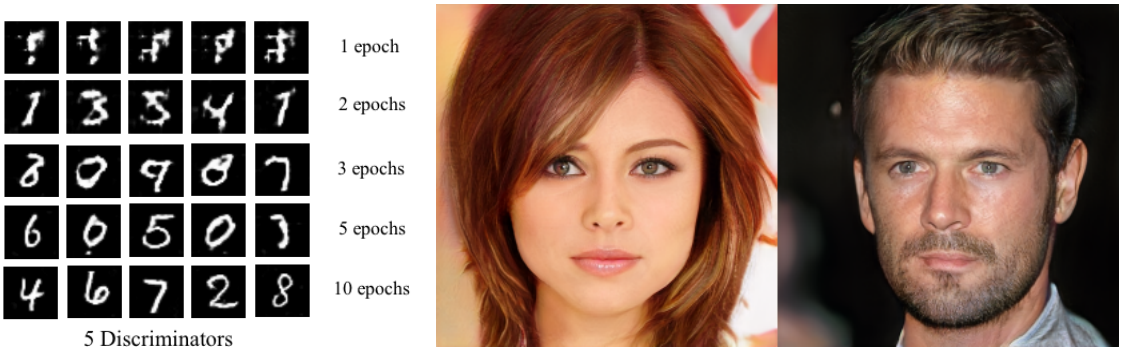


Figure 1.2: Our work presented in Chapter 4 can be used to accelerate training and improve the quality of generated samples. On the left, we show samples drawn from a generator G trained using techniques from this thesis on the MNIST handwritten digits dataset [64]. The rows show how the generated samples improve in quality throughout the training epochs. Our work motivated followup work [57] that led to the extremely high sample quality on the right for the CelebA dataset [69]. Note that the images shown are not of real celebrities—they were formed by the generator.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\Big[\log D(x)\Big] + \mathbb{E}_{z\sim p_z(z)}\Big[\log(1-D(G(z)))\Big]. \quad (1.15)$$

Goodfellow [40] was able to show that the equilibrium point of this game is the minimizer of the following optimization problem for the generator

$$\min_G -\log(4) + 2\cdot JSD(p_{data}||p_G) \quad (1.16)$$

where JSD denotes the Jensen-Shannon divergence, and p_G is the distribution defined by applying the generator's transformation function to $p_z(z)$. The Jensen-Shannon divergence is always positive except when its two arguments are equal, so this optimization problem is only solved when $p_{data} = p_G$, i.e., when the generating distribution matches the true distribution, which achieves the goal.

In this way, GANs formulate the solution to an optimization problem as the solution to an equilibrium problem. In general, it is not yet clear why one would prefer one formulation to another, but empirical results with GANs have shown them to learn qualitatively more accurate distributions of the data. Later on, we give a

theoretical motivation for why games may provide more representational power than optimization.

1.2.2 Dynamical Systems

This thesis will focus on the interactions between the different players' learning algorithms when employed together in a game. In general, we will analyze the vector field implied by the player's learning algorithms. We will assume player i's learning rule can be written as $x_{k+1}^{(i)} = x_k^{(i)} - \alpha F^{(i)}(x_k)$ where $\alpha > 0$ is a learning rate hyperparameter. We can then represent the simultaneous learning of all N agents as follows:

$$F(x) = [F^{(1)}(x), \ldots, F^{(N)}(x)] \tag{1.17}$$

$$x_{k+1} = x_k - \alpha F(x_k). \tag{1.18}$$

For instance, if $F^{(i)}(x_k) = \nabla_{x^{(i)}} f^{(i)}(x_k)$ where ∇_v is Feynman notation for taking the gradient with respect to the variable v only, then this update represents learning using simultaneous gradient descent. We will refer to $-F$ as the *dynamics* of the game and F as the (vector) field or map represented by the game. The connection to dynamical systems can be understood informally as reformulating the update in Equation (1.18) and taking the limit as the learning rate goes to zero. First, replace α with Δt for convenience. Then let $t = k\Delta t$ so that x indexed by k, x_k, corresponds to x evaluated at time t, $x^{\Delta t}(t)$, where Δt denotes the conversion factor between index and time. Then by definition,

$$x_{k+1} = x_k - \Delta t F(x_k) \tag{1.19}$$

$$\Rightarrow x_{k+1} - x_k = -\Delta t F(x_k) \tag{1.20}$$

$$\Rightarrow \frac{x_{k+1} - x_k}{\Delta t} = -F(x_k) \text{ for all } \Delta t \neq 0 \tag{1.21}$$

$$\Rightarrow \frac{x^{\Delta t}(t + \Delta t) - x^{\Delta t}(t)}{\Delta t} = -F(x^{\Delta t}(t)) \tag{1.22}$$

$$\Rightarrow \lim_{\Delta t \to 0} \frac{x^{\Delta t}(t + \Delta t) - x^{\Delta t}(t)}{\Delta t} = -F(x^{\Delta t}(t)) \tag{1.23}$$

$$\Rightarrow \frac{dx^{\Delta t}}{dt} = \dot{x}^{\Delta t} = -F(x^{\Delta t}). \tag{1.24}$$

This derivation is informal because a bijection between the natural numbers (indices) and real numbers (times) does not exist. Nevertheless, it should appeal to intuition and the connection is formalized in the book by Nagurney and Zhang [1996]. Unless stated otherwise, we will assume simultaneous gradient descent is used for learning. We suggest the book by Strogatz [2018] as an excellent introduction to dynamical systems.

1.2.3 Variational Inequalities and Monotone Operator Theory

We now show how a specific class of equilibrium problems subsume continuous, convex optimization problems. Just like in convex optimization problems, we will assume $\mathcal{X}$ is a convex set.

The equilibrium problem can be formalized with the theory of variational inequalities (VIs) [43]. The VI problem is to find x^* such that

$$\langle F(x^*), x' - x^* \rangle \geq 0 \quad \forall x' \in \mathcal{X}, \tag{1.25}$$

where $F : \mathcal{X} \to \mathbb{R}^m$. Notice that this is a simple generalization of the sufficient condition for the minimum of a convex function (i.e., replace F in Equation (1.8) with ∇f). Also x^* is a solution to the VI if and only if $x^* = P_{\mathcal{X}}(x^* - \eta F(x^*))$ where $P_{\mathcal{X}}$ is a projection on to $\mathcal{X}$. Figure 1.3 provides a geometric interpretation

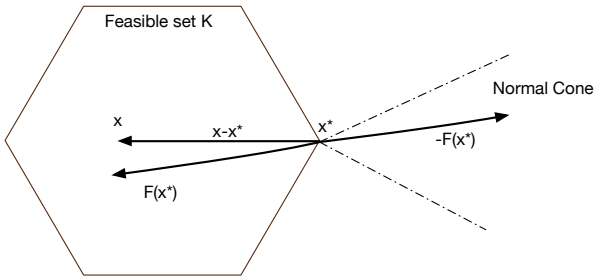


Figure 1.3: This figure provides a geometric interpretation of the variational inequality $VI(F, \mathcal{X})$. The mapping F defines a vector field over the feasible set $\mathcal{X}$ such that at the solution point x^*, the vector $F(x^*)$ is directed inwards at the boundary, and $-F(x^*)$ is an element of the normal cone $C(x^*)$ of $\mathcal{X}$ at x^* where the normal cone $C(x^*)$ at the vector x^* of a convex set $\mathcal{X}$ is defined as $C(x^*) = \{y \in \mathbb{R}^n | \langle y, x - x^* \rangle \leq 0, \forall x \in \mathcal{X}\}$.

of a variational inequality. In summary, the VI problem is to find an $x^* \in \mathcal{X}$ such that attempting to perturb x^* by $-F(x^*)$ either reveals that x^* is "stuck" against the boundary or $F(x^*) = 0$ meaning x^* is a stationary point with respect to F.

If F satisfies the following property, it is monotone:

$$\langle F(\hat{x}) - F(x'), \hat{x} - x' \rangle \geq 0 \quad \forall \hat{x}, x' \in \mathcal{X}. \tag{1.26}$$

Notice also that this is a simple generalization of convexity (i.e., replace F in Equation (1.7) with ∇f).

Therefore, to solve the convex optimization problem

$$\min_{x \in \mathcal{X}} f(x) \tag{1.27}$$

we can instead, equivalently solve the equilibrium problem that is to find x^* such that

$$\langle \nabla f(x^*), x' - x^* \rangle \geq 0 \quad \forall x' \in \mathcal{X}. \tag{1.28}$$

Nagurney and Zhang [1996] formalize the connection between projected dynamical systems and VIs and provide a good introduction to VI theory.

The motivation for considering monotone equilibrium problems rather than convex optimization problems is that F does not have to be the gradient of any function. We will exploit this generalization in Chapter 2, which will represent the first contribution of this thesis.

1.3 Motivating Problems

As stated in the introduction, equilibrium problems have recently risen to the forefront of ML research primarily due to the advent of GANs. Variants on the original GAN formulation [40] have achieved state-of-the-art results in numerous domains and applications. GANs have been successfully applied to image-to-image translation [124], pose transfer [72, 49], image super-resolution [65], text-to-image translation [122], image inpainting [28, 89, 118], and image anomaly detection [101]. Besides image modeling tasks, GANs have also been applied to simulating high particle physics [26], imitation learning for reinforcement learning (RL) [45], hybrid model-based RL [10], improving variational autencoders [31], drug discovery [54], and more. Each of these advances leverages the adversarial training paradigm presented in the original GAN work. Therefore, any gains made in understanding adversarial training more generally, i.e., equilibrium problems, can be shared to improve performance on each of these tasks. We present one such contribution in Chapter 4. We also provide tools for better visualizing adversarial training dynamics in Chapter 5 and explore a fundamental GAN variant in Chapter 3 that illustrates the difficulty of adversarial training.

Aside from GANs, equilibrium problems also appear in reinforcement learning (RL), distributed network resource allocation, and market economy models. For example, Li et al. [2018] present work on a market economy model where parameters of the model may drift [67]. In Chapter 2, we present a framework that guarantees that the economy will track the drifting equilibrium within a certain degree of accuracy.

This framework applies more generally to some algorithms in RL and some resource allocation policies as mentioned above.

CHAPTER 2

ONLINE MONOTONE EQUILIBRATION

2.1 Purpose of Research

As mentioned in the introduction, online optimization is important for training machine learning models on streaming datasets when we require the model to begin making predictions before seeing all of the data. However, we argued that many machine learning models are formulated as equilibrium problems. Therefore, we aim to develop a framework for solving a specific class of equilibrium problems online. The framework we develop, Online Monotone Equilibration (OME), subsumes the popular Online Convex Optimization framework.

Spoiler: The study of the online setting (OME) leads to a new extragradient algorithm with applications to Reinforcement Learning, specifically policy evaluation with linear value functions. Our proposed framework also provides an alternative derivation of *Crossing-the-Curl* (also known as *Symplectic Gradient Adjustment* [11]), a recently proposed algorithm for solving GANs.

2.2 Introduction

The primary focus of this chapter is on solving monotone Variational Inequality (VI) problems online. In order to develop an online framework suitable for evaluating and designing algorithms, we need a way to measure performance. As stated in the technical background of Chapter 1, the VI problem is to find x^* such that

$$\langle F(x^*), x' - x^* \rangle \geq 0 \quad \forall x' \in \mathcal{X} \tag{2.1}$$

where $F : \mathcal{X} \to \mathbb{R}^m$ and $\mathcal{X}$ is a convex set. F is monotone if and only if it satisfies the following property:

$$\langle F(\hat{x}) - F(x'), \hat{x} - x' \rangle \geq 0 \quad \forall \hat{x}, x' \in \mathcal{X}. \tag{2.2}$$

Notice that unlike optimization problems, the problem definition of a VI does not readily admit a performance metric for suboptimal predictions, x. In response, VI research has developed gap functions. A *gap function* is a function $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ which satisfies $\psi(x) \geq 0$ for all $x \in \mathcal{X}$ and $\psi(x^*) = 0$ if and only if x^* solves $\text{VI}(F, \mathcal{X})$. These are more commonly referred to as merit functions or loss functions in the optimization and machine learning literature. Numerous gap functions have been developed satisfying the properties above [32]. Despite their wide use, we opt for designing a new gap function for our purposes. We do this because gap functions are approximate in the following sense. We stated in the technical background that $\text{VI}(\nabla f, \mathcal{X})$ is equivalent to the optimization problem $\min_{x \in \mathcal{X}} f(x)$. For this reason, we desire a performance metric for $\text{VI}(F, \mathcal{X})$ that gracefully falls back to $f(x)$ when $F = \nabla f$ and gap functions generally do not satisfy this property.

In this work, introduce a new performance metric for VIs formulated as a path integral. We show that this formulation facilitates the design of an Online Monotone Equilibration framework equipped with no-regret algorithms. We also illuminate the boundaries between monotone equilibrium problems and other well known types of problems from the game theory literature. To demonstrate the utility of this framework for machine learning applications, we perform an online analysis of the family of GTD algorithms [106] for reinforcement learning. In summary, our primary contributions are

- the definition of online monotone equilibrium problems,
- the definition of our path integral regret, with accompanying linear bounds,

- algorithms that achieve sublinear regret,
- and examples of a variety of monotone equilibrium problems of interest.

2.3 Performance Metric

In order for a performance metric to be admissible for $\text{VI}(F, \mathcal{X})$, it must equal 0 at $x = x^*$ and it must be greater than 0 everywhere else. We also require that the performance metric reduces to $f(x)$ if $F = \nabla f$.

Consider the following path integral:

$$\int_{z:x^* \to x} \langle F(z), dz \rangle. \tag{2.3}$$

where $x^* \to x$ denotes a straight line path from $x^* \in \mathcal{X}^*$ to $x \in \mathcal{X}$. By definition, this path integral equals zero when $x = x^*$, i.e. start = end. Next consider the following useful integral upper bound over monotone maps (see Remark 3.10 in the work of Romano et al. [1993] for a more rigorous proof).

Lemma 1 (Path Integral Bound)**.** *The path integral over a monotone map is bounded by its linear approximations, i.e.,* $\langle F(a), b - a \rangle \leq \int_{x:a \to b} \langle F(x), dx \rangle \leq \langle F(b), b - a \rangle$.

Proof. Let $x_{i+1} - x_i = \frac{x_n - x_0}{n} \ \forall \ x_0, x_n$ and recall the definition of monotonicity, $\langle F(x_{i+1}) - F(x_i), x_{i+1} - x_i \rangle \geq 0 \ \forall \ x_i, x_{i+1}$ which implies

$$\langle F(x_i), \frac{x_n - x_0}{n} \rangle \leq \langle F(x_{i+1}), \frac{x_n - x_0}{n} \rangle \tag{2.4}$$

$$\implies \langle F(x_i), \frac{x_n - x_0}{n} \rangle \leq \langle F(x_j), \frac{x_n - x_0}{n} \rangle \ \forall \ j \geq i. \tag{2.5}$$

Also,

$$\langle F(x_0), x_n - x_0\rangle = \langle F(x_0), \sum_{i=0}^{n-1} \frac{x_n - x_0}{n}\rangle \tag{2.6}$$

$$= \sum_{i=0}^{n-1} \langle F(x_0), \frac{x_n - x_0}{n}\rangle \tag{2.7}$$

$$\leq \sum_{i=0}^{n-1} \langle F(x_i), \frac{x_n - x_0}{n}\rangle \tag{2.8}$$

$$= \int_{x:x_0 \to x_n} \langle F, dx\rangle \text{ as } n \to \infty, \tag{2.9}$$

and vice versa for the reverse direction, which implies

$$\langle F(x_0), x_n - x_0\rangle \leq \int_{x:x_0 \to x_n} \langle F, dx\rangle \leq \langle F(x_n), x_n - x_0\rangle. \tag{2.10}$$

□

If the map is strictly monotone, then the $\leq$'s can be strengthened to $<$'s. Therefore, for strictly and strongly monotone maps, we have

$$0 \leq \langle F(x^*), x - x^*\rangle < \int_{z:x^* \to x} \langle F(z), dz\rangle \tag{2.11}$$

where the first inequality follows from the definition of x^* being a solution to VI$(F, \mathcal{X})$. Finally, notice that when $F = \nabla f$,

$$\int_{z:x^* \to x} \langle \nabla f(z), dz\rangle = f(x) - f(x^*) = f(x), \tag{2.12}$$

and we recover $f(x)$ via the fundamental theorem of calculus for path integrals. We have assumed $f(x^*) = 0$ without loss of generality. From now on, we will refer to the path integral $\int_{z:x^* \to x} \langle F(z), dz\rangle$ with $f(x)$ as we have just shown that it is equivalent to $f(x)$ in the case of optimization and an explicit objective function is left unspecified in the VI problem formulation, $VI(F, \mathcal{X})$.

This path integral, $f(x)$, satisfies all requirements for strictly monotone maps, however, it does not satisfy the greater than 0 requirement for *all* monotone maps. Consider the following monotone (but not strictly monotone) map: $F(x) = Ax$ where $A = -A^\top$ is skew-symmetric. Clearly, $x^* = 0$ solves $\text{VI}(F, \mathbb{R}^n)$ as $\langle F(x^*), x - x^* \rangle = 0 \geq 0 \ \forall x$. However, the path integral is equal to zero for any x:

$$\int_{z:x^* \to x} \langle F(z), dz \rangle = \int_0^1 \langle F(x^* + (x - x^*)t), (x - x^*) \rangle dt \tag{2.13}$$

$$= \int_0^1 \langle F(xt), x \rangle dt = \int_0^1 (Axt)^\top x dt \tag{2.14}$$

$$= x^\top A^\top x \int_0^1 t dt = \frac{1}{2} x^\top A^\top x \tag{2.15}$$

$$= \frac{1}{4} x^\top (A + A^\top) x = 0. \tag{2.16}$$

This is because this map with skew-symmetric A represents dynamics with pure concentric cycles, i.e., the vector field is always perpendicular to the path from the origin. In order to build an online framework applicable to all monotone maps, we need to modify the path integral. We will assume the map is bounded, $||F(x)|| \leq L$, and smooth, $||F(x) - F(y)|| \leq \beta ||x - y||$. In addition, the following discussion assumes $\mathcal{X}$ is $\mathbb{R}^n$, however, this is just for sake of exposition; proofs in the Appendix follow through for any convex $\mathcal{X}$. Consider the modified path integral which first integrates to a point $\hat{x} = x - \hat{\eta} F(x)$ before continuing to x:

$$f(x) = \int_{z:x^* \to \hat{x}} \langle F(z), dz \rangle + \int_{z:\hat{x} \to x} \langle F(z), dz \rangle \tag{2.17}$$

$$\geq \underbrace{\langle F(x^*), \hat{x} - x^* \rangle}_{\geq 0 \text{ by def of } x^*} + \langle F(\hat{x}), x - \hat{x} \rangle \tag{2.18}$$

$$\geq 0 + \hat{\eta} \underbrace{\langle F(\hat{x}), F(x) \rangle}_{\exists \hat{\eta} > 0 \text{ s.t. } \langle \cdot, \cdot \rangle > 0} \tag{2.19}$$

$$> 0 \text{ for some } \hat{\eta} > 0 \text{ assuming } x \neq x^*, \tag{2.20}$$

where the first bound follows from Lemma 1 and intuitively, if $\hat{\eta}$ is small enough, $F(x)$ and $F(\hat{x})$ will align. We give a formal proof in Appendix A.5 that considers more general proximal updates, i.e., $\hat{x} = \texttt{prox}(x)$. We omit a discussion of proximal operators here to avoid complicating the exposition.

Also, note that $\hat{x} = x^* - \hat{\eta}F(x^*) = x^*$ so both path integrals vanish when $x = x^*$. Finally, maps that can be written as the gradient of some function, i.e., $F = \nabla f$, are known as conservative maps. Conservative maps are path-independent meaning the value of the path integral is independent of the path. That is to say that any path integral we choose, as long as it starts at x^* and ends at x, will recover $f(x)$ by way of the fundamental theorem of calculus.

Therefore, this path integral satisfies all requirements for all monotone maps assuming $\hat{\eta}$ is chosen small enough (see Appendix A.5 for details).

2.3.1 Online Variational Inequality Problems

In this section, we formalize the problem of solving $\text{VI}(F, \mathcal{X})$ *online* where $F(x) = \sum_t F_t(x)$ and each F_t is a monotone map. Let x^* be the solution to $\text{VI}(F, \mathcal{X})$. Repeating the same path integral loss as before:

$$f(x) = \int_{z:x^* \to \hat{x}} \langle F(z), dz \rangle + \int_{z:\hat{x} \to x} \langle F(z), dz \rangle \tag{2.21}$$

$$= \int_{z:x^* \to \hat{x}} \langle \sum_t F_t(z), dz \rangle + \int_{z:\hat{x} \to x} \langle \sum_t F_t(z), dz \rangle \tag{2.22}$$

$$= \sum_t \Big[\int_{z:x^* \to \hat{x}} \langle F_t(z), dz \rangle + \int_{z:\hat{x} \to x} \langle F_t(z), dz \rangle \Big] \tag{2.23}$$

where $\hat{x} = x - \hat{\eta}F(x)$. Unfortunately, in an online / streaming setting, at time t, we will only have seen $F_{\tau \leq t}$. This means we cannot construct $\hat{x}$ until we see the end of the stream. Moreover, in most online settings, we assume that we only observe F_t evaluated at a finite number of points, i.e., $\hat{x}_t$ and x_t, after which we throw the map

away in order to avoid storing all maps in memory. For these reasons, we consider the modified path integral loss for sums of maps:

$$\hat{f}(x) = \sum_t \Big[\int_{z:x^* \to \hat{x}_t} \langle F_t(z), dz \rangle + \int_{z:\hat{x}_t \to x} \langle F_t(z), dz \rangle \Big] \tag{2.24}$$

$$= \sum_t f_t(x|x^*) \tag{2.25}$$

where $\hat{x}_t = x - \hat{\eta} F_t(x)$ and we introduce the abbreviation $f_t(x|x^*)$ to represent the term above in brackets—it is the path integral loss over the map F_t starting at x^*. This path integral loss is

- equal to 0 if x equals x^*,
- equivalent to $f(x)$ if $F = \nabla f$,
- **however**, it may be less than 0 for some x not equal to x^*.

Although this path integral loss does not satisfy all three conditions, it is still promising for a number of reasons. First, if $F = \nabla f$ or $x_t^* = x^* \ \forall t$, then the third condition **is** met. And second, in Appendix A.5.2, we derive the following lower bound for $\hat{f}(x)$:

$$\hat{f}(x) \geq \sum_{t=1}^{T} \Big(||F_t(x)||_p - ||F_t(x^*)||_q \Big) ||F_t(x)||_p \frac{\hat{\eta}}{m} - \beta_t \frac{\hat{\eta}^2}{m^2} L_t^2. \tag{2.26}$$

By leveraging this lower bound along with additional information, we are able to show later that minimizing the path integral, $\hat{f}(x)$, at a sufficient rate ensures that the average norm over t of $F_t(x)$ approaches the average norm of $F_t(x^*)$ as $T \to \infty$.

2.4 Online Monotone Equilibration

We are now ready to present a framework for Online Monotone Equilibration (Algorithm 2: OME) that will enable us to derive upper bounds for regret in online monotone equilibrium problems.

Framework 2 Online Monotone Equilibration (OME)

input: A convex set $\mathcal{X} \subseteq \mathbb{R}^n$
define: $\hat{x}_t = x_t - \hat{\eta} z_t$ where $z_t \in F_t(x_t)$
for all $t = 1, 2, \ldots$ **do**
  predict a vector $x_t \in \mathcal{X}$
  receive a vector, $\hat{z}_t$, from a monotone map, i.e., $\hat{z}_t \in F_t(\hat{x}_t)$
  suffer $f_t(x_t)$
end for

Note that the loss at each round, $f_t(x_t)$, assumes there is an oracle with knowledge of x^*. We will show later that despite this, knowledge of x^* is not required for learning in the OME framework.

We repeat the Online Convex Optimization framework (OCO) here for comparison.

Framework Online Convex Optimization (OCO)

input: A convex set $\mathcal{X} \subseteq \mathbb{R}^n$
for all $t = 1, 2, \ldots$ **do**
  predict a vector $x_t \in \mathcal{X}$
  receive a vector, z_t, from the subdifferential of a convex loss, i.e., $z_t \in \nabla f_t(x_t)$
  suffer loss $f_t(x_t)$
end for

Comparing OME to OCO, we see that the major difference is that we now receive vectors from a monotone map (at $\hat{x}_t$) whereas in OCO, we receive gradients of a convex loss (at x_t). In some cases, OME reduces to OCO, however, this is not always the case, so we cannot rely on OCO theory alone to bound f_t. In general, OME represents a strict superset of OCO (see Appendix A.2.1).

Theorem 1. *$OCO(f_t, \mathcal{X})$ is equivalent to $OME(\partial f_t, \mathcal{X})$ and $\exists F_t$ such that $OME(F_t, \mathcal{X}) \notin \{\forall f_t\ OCO(f_t, \mathcal{X})\}$ implying $OCO \subset OME$ in the strict sense.*

Figure 2.1 displays an example of a function resulting from a path integral over a monotone field $F : \mathbb{R}^2 \to \mathbb{R}^2$; the function is non-convex.

However, if F is affine, OME is equivalent to OCO (see Appendix A.3).

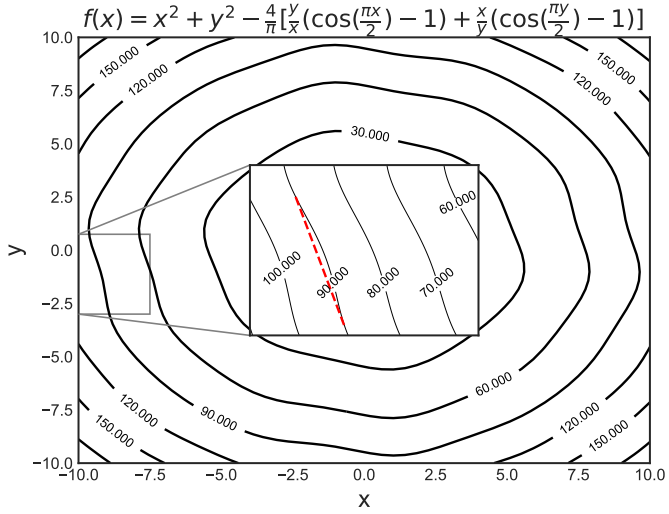


Figure 2.1: This figure shows the contour plot for the function representing the path integral over a 2-D monotone field, $F(x, y)$—observe the path integral function displayed in title. Notice in the inset that the function value in the interior of the line segment is greater than the function value at either endpoint. This implies that the function is not even quasi-convex, a weaker condition than convexity. The definition of the field and derivation of the path integral can be found in Appendix A.2.2.

Theorem 2. *If $F_t(x_t) = Ax_t + b$ and A is positive-definite, then there exists f_t such that OME$(F_t, \mathcal{X})$ is equivalent to OCO$(f_t, \mathcal{X})$.*

2.5 Upper Bound for Cumulative Path Integral Loss

Previously, we established lower bounds for the path integral to show it satisfied certain properties of a loss function. Now we will establish upper bounds that we can minimize efficiently with familiar algorithms.

The cumulative path integral loss can be upper bounded as follows:

$$\hat{f}(x) = \sum_{t=1}^{T} \Big[\int_{x:x^* \to \hat{x}_t} \langle F_t(x), dx \rangle + \int_{x:\hat{x}_t \to x} \langle F_t(x), dx \rangle \Big] \tag{2.27}$$

$$\leq \sum_{t=1}^{T} \Big[\langle F_t(\hat{x}_t), \hat{x}_t - x^* \rangle + \langle F_t(x), x - \hat{x}_t \rangle \Big] \tag{2.28}$$

$$= \sum_{t=1}^{T} \Big[\langle F_t(\hat{x}_t), x - \eta F_t(x) - x^* \rangle + \hat{\eta} \langle F_t(x), F_t(x) \rangle \Big] \tag{2.29}$$

$$= \sum_{t=1}^{T} \Big[\langle F_t(\hat{x}_t), x - x^* \rangle + \hat{\eta} \langle F_t(x), F_t(x) - F_t(\hat{x}_t) \rangle \Big] \tag{2.30}$$

$$= \sum_{t=1}^{T} \Big[\langle \hat{z}_t, x - x^* \rangle + \hat{\eta} \langle z_t, z_t - \hat{z}_t \rangle \Big] \tag{2.31}$$

where we have replaced the map evaluations with z's to emphasize that these vectors are potentially chosen by an adversary. Notice that we have upper bounded $\hat{f}(x)$ by a sum of functions that are linear in x. Therefore, the OME problem reduces to the Online Linear Optimization (OLO) problem and we can reuse techniques designed for OLO. This reduction is mirrored in the OCO framework as well. Formally,

$$regret_{\mathcal{A}}^{T}(\mathcal{X}) = \sum_{t=1}^{T} f_t(x_t|x^*) - f_t(x^*|x^*) = \sum_{t=1}^{T} f_t(x_t|x^*) \tag{2.32}$$

$$\leq \sum_{t=1}^{T} \langle \hat{z}_t, x_t - x^* \rangle + \hat{\eta} \langle z_t, z_t - \hat{z}_t \rangle \tag{2.33}$$

$$\leq \sum_{t=1}^{T} \langle \hat{z}_t, x_t - x^* \rangle + \frac{\beta_t L_t^2}{m^2} \hat{\eta}^2 \tag{2.34}$$

$$\leq \frac{1}{2\eta} ||x^*||^2 + (\eta + \frac{\beta_{\max}}{m^2} \hat{\eta}^2) \sum_{t=1}^{T} L_t^2 \tag{2.35}$$

$$\leq \frac{5}{4} BL\sqrt{2T}. \tag{2.36}$$

where $||x^*|| \leq B, ||F_t|| \leq L_t, L^2 \geq \frac{1}{T}\sum_t L_t^2, \beta_{\max} = \max_t \beta_t, \eta = \frac{B}{L\sqrt{2T}}$, and $\hat{\eta} = \sqrt{\frac{\eta}{2\beta_{\max}}}$ (see Appendix A.5 for details).

There is one other subtle problem with our path integral loss as we have defined it. Observe that as $\hat{\eta} \to 0$, $\hat{f}(x)$ approaches our original path integral loss, which we

argued is only applicable for strictly monotone maps. In order for this path integral loss to be meaningful, we need to show that our algorithms are actually minimizing this loss at a rate that is faster than $\hat{\eta}$'s rate of decay. At the very least, any surrogate loss that we propose must imply some sort of convergence in the offline case to be admissible in the online setting. In Appendix A.5.5, we leverage the fact that our lower bound derived in Equation (2.26) must, by definition, be *lower* than our upper bound derived in Equation (2.34). We use this to show that an *offline* algorithm that minimizes the above regret at the rate displayed in Equation (2.36) implies $||F_t(x)||$ approaches $||F_t(x^*)||$ on average, i.e.,

$$||F_t(x)|| \leq ||F_t(x^*)|| + T^{-1/8}\sqrt{C} \tag{2.37}$$

where $C = 8\frac{\sqrt{\beta_{\max} BL^3}}{2^{1/4}}$. Note that this bound does **not** necessarily imply that $||F_t(\mathbf{x_t})|| \leq ||F_t(x^*)|| + T^{-1/8}\sqrt{C}$ on average for some C. In future work, we will explore if this is a deficiency of our path integral loss or of the analysis. However, the main takeaway here is that the path integral loss, as defined, is meaningful in the offline case, which satisfies a natural baseline for admissibility in the online setting.

2.5.1 Derivation of No-Regret Algorithms for OME

Due to the work previously done in OLO, part of the derivation of no-regret algorithms for OME is trivial. We have shown that instantaneous regret for general monotone maps can be bounded above by considering the constant approximation of the map (see Equation (2.33)). Note that a constant map, $F_t(x_t)$, is always the subgradient of some linear function, $f_t(x) = \langle F_t(x_t), x \rangle$. This implies that the regret for general monotone maps is bounded above by considering the online linear optimization problem with $f_t(x)$. The implication is that the online gradient descent and even online mirror descent algorithms can be adapted from OLO with almost no effort to minimize regret in OME while enjoying similar $o(T)$ regret bounds (see

Algorithm 3 Online Extragradient Descent (OED)

input: Scalar learning rates $\eta > 0$ and $\hat{\eta} > 0$
$x_1 = 0$
for all $t = 1, 2, \ldots$ **do**
  $\hat{x}_t = x_t - \hat{\eta} z_t$ where $z_t \in F_t(x_t)$
  $x_{t+1} = x_t - \eta \hat{z}_t$ where $\hat{z}_t \in F_t(\hat{x}_t)$
end for

Algorithm 4 Online Mirror Prox (OMP)

input: Link function $g_\eta : \mathbb{R}^n \to \mathcal{X}$ and proximal operator $\texttt{prox}_{\hat{\eta}} : \mathbb{R}^n \to \mathcal{X}$
$\theta_1 = 0$
for all $t = 1, 2, \ldots$ **do**
  $x_t = g_\eta(\theta_t)$
  $\hat{x}_t = \texttt{prox}_{\hat{\eta}}(x_t) = \arg\min_{y \in \mathcal{X}} \left(\langle z_t, y \rangle + \frac{1}{\hat{\eta}} D(y, x_t) \right)$ where $z_t \in F_t(x_t)$
  $\theta_{t+1} = \theta_t - \hat{z}_t$ where $\hat{z}_t \in F_t(\hat{x}_t)$
end for

Equation (2.36)). This is somewhat surprising as Theorem 1 indicates that OME captures minimizing some non-convex functions. The no-regret algorithms for OME are given in Algorithms 3 and 4. Please see Appendix A.5 for a more thorough discussion of proximal operators as well as the book by Shalev-Shwartz [2011] for commonly used link functions.

These two algorithms may be recognized as online variants of the familiar Extragradient and Mirror Prox algorithms commonly seen in the VI [62] and (online) optimization and saddle point [53] literature. Our work generalizes their use beyond optimization and saddle point (2-D games) problems to more general monotone equilibrium problems (e.g., N-player games). In addition, our construction of the path integral loss gives a clear reason as to why Extragradient algorithms (as opposed to gradient descent) are necessary for solving monotone equilibrium problems. Note that the original path integral loss we considered leads to online gradient descent, but

as discussed earlier, this loss is not greater than 0 for all $x \neq x^*$ even in the offline setting.

It is also worth highlighting the difference between the Extragradient we present here and the variants seen in the literature. In particular, our variant uses a stepsize, $\hat{\eta}$, for computing $\hat{x}_t$, that grows relative to the stepsize, η, for computing x_t. Other variants in the literature use

- the same constant stepsize for both steps [25],
- average the iterates after training [39, 68] (also seen in socially-convex games [33]),
- average the maps during training [50],
- inertial proximal methods [4],
- square-summable, non-summable stepsizes [55, 56, 85, 120],
- modified EG algorithms [112],
- or complex stepsize schemes [27].

This difference in step size is crucial. Consider revisiting the problem $VI(F(x) = Ax, \mathbb{R}^n)$ where $A = -A^\top$ is skew-symmetric. If we use constant step sizes given a time horizon T and $\eta = \hat{\eta} = T^{-1/2}$, one can show that $\lim_{T\to\infty} ||x_T||^2 \to \frac{1}{e}||x_0||^2$, i.e., x_T does not converge to the equilibrium at $x^* = 0$ (see Appendix A.5.5). On the other hand, if $\eta = T^{-1/2}$ and $\hat{\eta} = T^{-1/4}$ as we have proposed, x_T does converge to x^*.

Most importantly, our online Extragradient method does not require storing or averaging either the iterates or the maps. This allows processing data streams using minimal memory and processor time.

2.6 Algorithmic Game Theory and Related Work

Related results in Algorithmic Game Theory (AGT) focus on maximizing welfare, W, which is the sum of player utilities (i.e., minus the sum of player losses). In

order to compare to results in AGT, we define a monotone game as one in which the map formed by concatenating the gradients of all players' losses is monotone, i.e. $F(x) = [\nabla f^{(1)}(x), \ldots, \nabla f^{(N)}(x)]$ is monotone.

OME is a framework that examines (external) regret in monotone games. Monotone games and socially-convex games are both subsets of convex games (see Theorems 6 and 7 in Appendix A.8). Convex games are games in which each agent's cost function, $f^{(i)}(x)$, is convex with respect to its own strategy, $x^{(i)}$. Gordon et al. [2008] studied internal and external regret for individual agents in convex games and related these to convergence towards correlated and coarse-correlated equilibria respectively. Note that in general, these results on equilibria do not imply results for welfare. Even-Dar et al. [2009] examined no-regret algorithms in *socially-concave games* (equivalently formulated as *socially-convex* games), and showed that each player's average strategy approaches that player's strategy at the Nash equilibrium; also, each player's average utility approaches that player's utility at the Nash equilibrium. Roughgarden [2009] developed the notion of *smooth* games not to be confused with the β-smooth maps defined earlier. Smoothness relates the convergence of strategies towards Nash equilibria to the *price of anarchy* (PoA), which defines the ratio of the worst-case sum cost of a Nash equilibrium, $\max_{x^* \in \mathcal{X}^*} -W(x^*)$, to the best-case sum cost of a player strategy set, $\min_{x \in \mathcal{X}} -W(x)$. In short, smoothness relates no-regret dynamics to the welfare of a game. The results above for convex games and socially-convex games apply for repeated play (the game is fixed), therefore, we do not consider these settings *online*; the smoothness results also apply to games that may change at each step (i.e., *online*). Table 2.1 outlines the intersections between the various game types.

Additional performance gains can be obtained if we can assume each player in the game is employing an algorithm from a given class. Syrgkanis et al. [2015] have accelerated convergence to Nash equilibria, to zero-regret, and to optimal welfare (assuming the game is smooth and fixed) under this scenario. An extension of this work

Type / Ex.	A.8.1	A.8.2	A.8.3	A.8.4	A.8.5	A.8.6	A.8.7	A.8.8	A.8.9
Smooth	√	√	√	√	√				
Convex		√	√	√	√	√	√	√	√
Monotone			√		√		√		√
Socially-Convex				√	√			√	√

Table 2.1: Games may share multiple properties at once. Definitions of properties and examples for each case (denoted by the column heading) are given in Appendix A.8.

also considers scenarios where the game is changing at each time step [71]. Specifically, Foster et al. [2016] showed that approximate optimality can be guaranteed if the game allows players to be replaced with probability p for small p. Moreover, this is true even when the players observe only bandit feedback (as opposed to expected costs) when comparing against a dynamic baseline. Critically, bounds on welfare are still derived from a smoothness constraint on the game.

In contrast, OME does not require Roughgarden's definition of smoothness and allows the game to change with probability 1 at each step (i.e., *online*), however, this framework bounds a weaker notion of regret, *auto*-welfare regret (see Appendix A.7), as opposed to welfare regret.

2.7 Applications

We illustrate applications of Online Monotone Equilibration on modeling several concave games, solving an online variational inequality problem (OVI), uncovering insights into a saddle-point based reinforcement learning algorithm, and fitting a generative model. Note that the majority of these equilibrium problems are actually strongly monotone, which allows us to use the simpler path integral loss in Equation (2.11) instead of the integral defined with an intermediate $\hat{x}_t$ for more general monotone fields. We will denote an application as monotone, strictly, or strongly monotone with the symbols (M), (sM), or (SM).

2.7.1 Concave Games

Even-Dar et al. [2009] developed the theory of socially-concave games and showed that minor simplifications of a number of concave games are socially-concave as well. In Appendix A.9, we show that variants of these games also satisfy monotonicity. Specifically, we prove F is monotone for the following games:

- Linear *Cournot* Competition (sM) [21]—Firms compete for consumers by adjusting quantities of goods produced. The price of goods is set by a linear function of the total quantity of goods in the market.

- Linear Resource Allocation (M) [52]—A network controller oversees the sharing of a communication channel, ensuring total communication does not exceed the network capacity. Users submit bids to the controller, which the controller uses when deciding how to allocate capacity. User value functions are linear.

- Congestion Control Protocols (M) [33]—We consider a *Tail Drop* policy where a router drops packets that exceed the network capacity.

In Appendix A.9.2, we analyze resource allocation and compare welfare with our path integral loss. In this case, welfare is maximized when all users submit the minimum bid amount. This result is independent of the parameters of the users' utility functions, which from a modeling viewpoint is unsatisfying. In contrast, the path integral loss is minimized by bids with an intuitive dependence on the utility function parameters: 1) as the penalty for large bids grows, the optimal bid amount decreases; 2) as the number of users increases, the optimal bid amount increases (due to increased competition), approaching an asymptote.

2.7.2 A Machine Learning Economy (SM)

Next, we consider a cloud-based machine learning network (MLN) adopted from the work of Nagurney and Wolf [2014]. Providers of machine learning data control

the quantity of data provided while network providers control the delivery price as well as service quality. Consumers influence the network through demand functions dictating the prices they are willing to pay for specific quantities and qualities of services rendered. See Appendix A.10 for a more thorough description.

We define each firm's utility function to be concave and quadratic in its strategy. This establishes the equivalence between the equilibrium state we are searching for and the variational inequality to be solved, $\mathrm{VI}(F, \mathbb{R}^+)$, where F_t returns a vector consisting of the negative gradients of the utility functions for each firm.

To cast this VI as an online learning problem, we let the parameters of the network change. This creates a more realistic model as a number of external factors can cause the network to change such as complex network congestion effects, network outages, etc. The goal then is to predict the equilibrium point of each new MLN in the face of these possibly adversarial forces. Specifically, our experiment considers ten different five-firm networks. At each time step, the adversary receives OED's prediction for the equilibrium point and returns the MLN whose equilibrium is farthest from the predicted one.

Figure 2.2 plots average regret with respect to the time step, demonstrating that average regret approaches zero in support of our derived sublinear bounds.

2.7.3 GTD Algorithms (SM)

Reinforcement Learning (RL) is a class of learning problems in which an agent attempts to maximize a long-term reward in an unfamiliar environment by reinforcing rewarding behaviors. Solving this problem typically requires first learning a *value function*, $V^\pi(s)$, which gives the long-term reward the agent is expected to receive if employing policy π and starting from state s. Often, we learn an approximate value function instead, $V_\theta^\pi(s)$, parameterized by $\theta \in \mathbb{R}^d$. Approximating $V_\theta^\pi(s)$ by observing an exploratory policy, π', is called *off-policy policy evaluation*. The gradient temporal

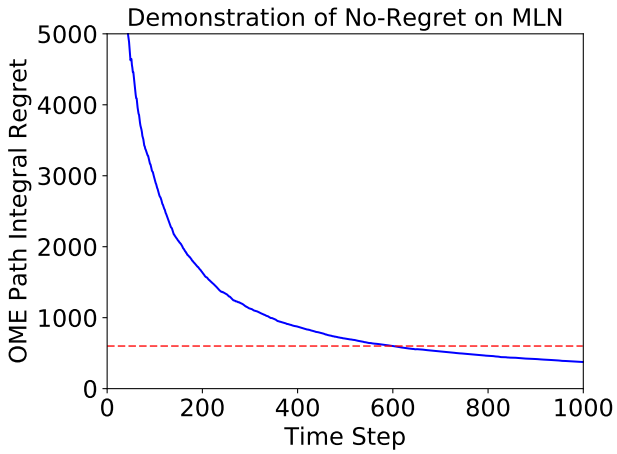


Figure 2.2: Demonstration of OMP on the described machine learning network. The dotted line denotes the upper bound derived for the regret of OMP.

difference (GTD) learning algorithms form a family of algorithms for accomplishing this task [106, 107].

Liu et al. [2015] showed that although the family of GTD algorithms are technically not gradient algorithms with respect to their original objectives, they are gradient algorithms with respect to the following saddle point objective, which is an example of a two-player game:

$$\min_{\theta} \max_{y} y^{\top}(b - A\theta) - \frac{1}{2}||y||_M^2 \tag{2.38}$$

where samples of A, b, and M are obtained from observing trajectories according to the behavioral policy and y is an auxiliary variable used by the GTD algorithm. We show that this game is strongly monotone (see Appendix A.11). The GTD update rules are given by

$$y_{t+1} = y_t + \eta_t(b - A\theta_t - My_t) \tag{2.39}$$

$$\theta_{t+1} = \theta_t + \eta_t(A^{\top}y_t) \tag{2.40}$$

where $M = \mathbb{1}_d$ or M is a covariance matrix. These updates are equivalent to running online gradient descent on an appropriate two-player game. Either way, M is symmetric positive definite and the corresponding map, F, is strongly monotone with parameter 1 for $M = \mathbb{1}_d$, or more generally, $\lambda_{\min}(M)$. Due to the strong-monotonicity, we do not need to consider the modified path integral that proceeds through the intermediate point $\hat{x}_t$. We can take the simpler path that goes directly to x.

So far, we have considered path integrals that start at x^*. In some cases (and in this case), it is more illuminating to consider a path integral that starts at a more general x_o and redefine $f_t(x_t) = \int_{z:x_o \to x_t} \langle F_t(z), dz \rangle - \int_{z:x_o \to x^*} \langle F_t(z), dz \rangle$. We discuss the technical details of this change in Appendix A.6. In this more general setting, the corresponding path integral loss that GTD bounds is

$$f([y_t; \theta_t]) = [y_0^\top (b - A\theta_t) - y_t^\top (b - A\theta_0)] + \frac{1}{2}||y_t||_M^2. \tag{2.41}$$

Now consider θ's task of minimizing $f([y; \theta])$ relative to $\theta_0 = \theta^*$ with y_0 fixed at y. In this case, $f([y; \theta])$ reduces to $y^\top (b - A\theta) + \frac{1}{2}||y||_M^2$. From the perspective of θ, this is equivalent to minimizing

$$y^\top (b - A\theta) - \frac{1}{2}||y||_M^2. \tag{2.42}$$

Similarly, consider y's task of minimizing $f([y; \theta])$ relative to $y_0 = y^*$ with θ_0 fixed at θ. In this case, $f([y; \theta])$ reduces to $-y^\top (b - A\theta) + \frac{1}{2}||y||_M^2$. From the perspective of θ, this is equivalent to maximizing

$$y^\top (b - A\theta) - \frac{1}{2}||y||_M^2. \tag{2.43}$$

Therefore, we have recovered the original saddle point problem by simply evaluating the path integral from each players' perspective! In this way, the path integral

has prescribed a constructive procedure for recovering a problem formulation that previously required a careful eye to conjure.

2.7.4 Constant-Linear GANs (M)

Generative adversarial networks formulate the training of a generative model as a game [40]. The original formulation is a minimax game between a generator, $G(z) : z \to x$, and a discriminator, $D(x) : x \to [0, 1]$,

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}\Big[\log(D(x))\Big] + \mathbb{E}_{z \sim p_z(z)}\Big[\log(1 - D(G(z)))\Big], \quad (2.44)$$

where $p_{data}(x)$ is the true data distribution and $p_z(z)$ is a simple (usually fixed) distribution that is easy to draw samples from, e.g., $\mathcal{N}(0, 1)$.

Unfortunately, this game is not monotone in general, however, we can derive a monotone version of the Wasserstein-type GAN [6] with online guarantees. The new minimax objective is

$$\min_G \max_d V(G, d) = \mathbb{E}_{x \sim p_{\text{data}(x)}}[d^\top x] - \mathbb{E}_{z \sim p_z(z)}[d^\top G z] \quad (2.45)$$

where $x \in R^n$, $z \in R^m$, $d \in R^n$, $G \in R^{n \times m}$. We derive the map, F, associated with this game in Appendix A.12, Equation (A.328). F is monotone for any $p_z(z)$ and $p_{\text{data}}(x)$. If G and d are regularized with $\frac{\alpha}{2}|| \cdot ||_2^2$, then F is strongly monotone with parameter α.

In Appendix A.13, we show that for any differentiable, strictly convex-concave minimax game, the corresponding path integral loss that Algorithms 3 and 4 bound is

$$f([G; d]) = V(G, d^*) - V(G^*, d). \quad (2.46)$$

which is a familiar Lyapunov function for the game. In the case that the minimax game is only convex-concave, not strictly, the path integral is

$$f([G;d]) = V(\hat{G}, d^*) - V(G^*, \hat{d}) + V(G, \hat{d}) - V(\hat{G}, d) \tag{2.47}$$

which, for small $\hat{\eta}$, is close to the previous path integral.

2.8 Conclusion

We proposed a new framework for online learning, namely Online Monotone Equilibration, which enables the study of regret for online monotone equilibrium problems. This framework generalizes the popular Online Convex Optimization framework in a way that allows it to model regret for equilibrium problems while still retaining the simplicity of standard no-regret algorithms from previous work. We support the broad applicability of our new framework with connections to network congestion protocols, empirical results from a VI, analysis of an existing RL algorithm, and design of a generative model.

In terms of technical contributions, we illuminated the boundary between OCO and OME (e.g., affine maps), showed that OME can successfully frame some online non-convex problems, defined a new, more general notion of regret, and derived efficient, realtime algorithms with sublinear regret.

2.8.1 Up Next

In this chapter, we showed that the path integral loss can be upper bounded by the expression below. Assuming our intermediate step to $\hat{x}$ uses a small step size, $\hat{\eta}$, we would expect $z_t \in F(x_t)$ and $\hat{z}_t \in F(\hat{x}_t)$ to be relatively well aligned, i.e., $\hat{z}_t \approx \rho z_t$. This allows the following approximation (let $\hat{\rho} = \hat{\eta}(1-\rho)$):

$$f_t(x_t) \leq \langle \hat{z}_t, x_t - x^* \rangle + \hat{\eta} \langle z_t, z_t - \hat{z}_t \rangle \tag{2.48}$$

$$\approx \langle \hat{z}_t, x_t - x^* \rangle + \hat{\rho} ||z_t||^2, \tag{2.49}$$

which supports the derivation of a new algorithm in the next chapter.

CHAPTER 3

LINEAR QUADRATIC GANS AND CROSSING-THE-CURL

3.1 Purpose of Research

Equilibrium problems introduce additional challenges over optimization problems. Despite these challenges, initial research on GANs began with extremely complex, neural network models and little accompanying theory. Linear-Quadratic GANs (LQ-GANs) were recently introduced as a testbed for better understanding GAN training and equilibria [80]. LQ-GANs can be used to fit a multivariate Gaussian distribution to data, a fundamental task in generative modeling / density estimation. They replace the standard neural-network generator and discriminator with a linear generator and a quadratic discriminator. This makes analysis more tractable and insights gleaned in this setting will hopefully lead to better algorithms in the more powerful neural-network setting. Here, we aim to better understand the LQ-GAN setting and propose a new algorithm with provable convergence guarantees in this setting.

Spoiler: We present an intuitive derivation of *Crossing-the-Curl* (also known as SGA [11]), a recently proposed algorithm for solving GANs, and prove that it can be used to fit normal distributions to data with convergence guarantees. This algorithm does not have a large impact on the training of neural-network based GANs, however, our analysis reveals an additional property that does transfer to the more complex setting. Specifically, we solve the LQ-GAN by successively increasing the complexity of the generator and discriminator throughout training. This curriculum is mirrored in state-of-the-art GANs [57] as well as our own work in the next chapter.

3.2 Introduction

When minimizing $f(x)$ over $x \in \mathcal{X}$, it is known that f decreases fastest if x moves in the direction $-\nabla f(x)$. In addition, any direction orthogonal to $-\nabla f(x)$ will leave $f(x)$ unchanged. In this chapter, we show that these orthogonal directions that are ignored by gradient descent can be critical in equilibrium problems, which are central to game theory. If each player i in a game updates with $x^{(i)} \leftarrow x^{(i)} - \rho \nabla_{x^{(i)}} f^{(i)}(x)$, $x = [x^{(1)}; x^{(2)}; \ldots]^\top$ can follow a cyclical trajectory, similar to a person riding a merry-go-round (see Figure 3.1). This toy scenario perfectly reflects an aspect of training for a particular machine learning model mentioned below, and is depicted more technically later on in Figure 3.2. To arrive at the equilibrium point, a person riding the merry-go-round should walk perpendicularly to their direction of travel, taking them directly to the center.

Equilibrium problems have drawn heightened attention in machine learning due to the emergence of the Generative Adversarial Network (GAN) [40]. GANs have served a variety of applications including generating novel images [57], simulating particle physics [26], and imitating expert policies in reinforcement learning [45]. Despite this plethora of successes, GAN training remains heuristic.

Deep learning has benefited from an understanding of simpler, more fundamental techniques. For example, multinomial logistic regression formulates learning a multiclass classifier as minimizing the cross-entropy of a log-linear model where class probabilities are recovered via a `softmax`. The minimization problem is convex and is solved efficiently with guarantees using stochastic gradient descent (SGD). Unsurprisingly, the majority of deep classifiers incorporate a `softmax` at the final layer, minimize a cross-entropy loss, and train with a variant of SGD. This progression from logistic regression to classification with deep neural nets is not mirrored in GANs. In contrast, from their inception, GANs were architected with deep nets. Only recently

has the Linear-Quadratic GAN (LQ-GAN) [36, 80] been proposed as a minimal model for understanding GANs.

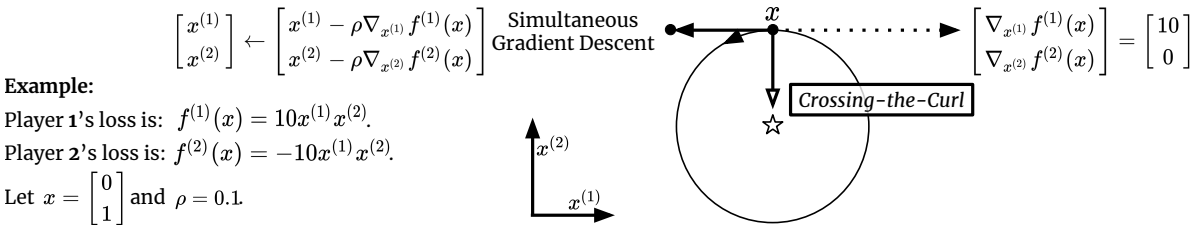


Figure 3.1: The goal is to find the equilibrium point (denoted by the star) of the merry-go-round. If someone follows simultaneous gradient descent, she will ride along in circles forever. However, if she travels perpendicularly to this direction, a.k.a. *Crosses-the-Curl*, she will arrive at the equilibrium.

In this chapter, we analyze the convergence of several GAN training algorithms in the LQ-GAN setting. We survey several candidate theories for understanding convergence in GANs, naturally leading us to select Variational Inequalities, an intuitive generalization of the widely relied-upon theories from Convex Optimization. According to our analyses, none of the current GAN training algorithms is globally convergent in this setting. We propose a new technique, *Crossing-the-Curl*, for training GANs with guaranteed convergence in the N-dimensional (N-d) LQ-GAN setting.

This work makes the following contributions (proofs can be found in Appendix B):

- The first global convergence analysis of several GAN training methods for the N-d LQ-GAN,
- *Crossing-the-Curl*, the first technique with $\mathcal{O}(N/k)$ stochastic convergence for the N-d LQ-GAN,
- An empirical demonstration of *Crossing-the-Curl* in the multivariate LQ-GAN setting as well as some common neural network driven settings in Appendix B.16.

3.3 Generative Adversarial Networks

The Generative Adversarial Network (GAN) [40] formulates learning a generative model of data as finding a Nash equilibrium of a minimax game. The generator (min player) aims to synthesize realistic data samples by transforming vectors drawn from a fixed source distribution, e.g., $\mathcal{N}(\mathbf{0}, I_d)$. The discriminator (max player) attempts to learn a scoring function that assigns low scores to synthetic data and high scores to samples drawn from the true dataset. The generator's transformation function, G, and discriminator's scoring function, D, are typically chosen to be neural networks parameterized by weights θ and ϕ respectively. The minimax objective of the original GAN [40] is

$$\min_{\theta} \max_{\phi} \Big\{ V(\theta, \phi) = \mathbb{E}_{y \sim p(y)}[g(D_\phi(y))] + \mathbb{E}_{z \sim p(z)}[g(-D_\phi(G_\theta(z))] \Big\}, \tag{3.1}$$

where $p(z)$ is the source distribution, $p(y)$ is the true data distribution, and $g(x) = -\log(1 + e^{-x})$.

In practice, finding the solution to Equation (3.1) consists of local updates, e.g., SGD, to θ and ϕ. This continues until 1) V has stabilized, 2) the generated data is judged qualitatively accurate, or 3) training has de-stabilized and appears irrecoverable, at which point, training is restarted. The difficulty of training GANs has spurred research that includes reformulating the minimax objective [6, 73, 78, 79, 87, 114, 123], devising training heuristics [42, 57, 98, 96], proving the existence of equilibria [8], and conducting local stability analyses [39, 74, 75, 80].

We acknowledge here that our algorithm, *Crossing-the-Curl*, was independently proposed in [11] as *Symplectic Gradient Adjustment* (SGA). In contrast to that work, this chapter specifies a non-trivial application of this algorithm to LQ-GAN which obtains guaranteed global convergence.

Recent work has studied a simplified setting, the Wasserstein LQ-GAN, where G is a linear function, D is a quadratic function, $g(x) = x$, and $p(z)$ is Gaussian [36, 80].

Follow-up research has shown that, in this setting, the optimal generator distribution is a rank-k Gaussian containing the top-k principal components of the data [36]. Furthermore, it is shown that if the dimensionality of $p(z)$ matches that of $p(y)$, LQ-GAN is equivalent to maximum likelihood estimation of the generator's resulting Gaussian distribution. To our knowledge, no GAN training algorithm with guaranteed convergence is currently known for this setting. We revisit the LQ-GAN in more detail in Section 3.5.

3.4 Convergence of Equilibrium Dynamics

In this section, we briefly review Variational Inequalities (VIs) and compare it to the ODE Method leveraged in recent work [80]. See B.1.2 and B.1.1 for a discussion of two additional theories.

3.4.1 Variational Inequalities

Variational Inequalities (VIs) are used to study equilibrium problems in a number of domains including mechanics, traffic networks, economics, and game theory [23, 34, 43, 81]. The Variational Inequality problem, $\text{VI}(F, \mathcal{X})$, is to find an x^* such that for all x in the feasible set $\mathcal{X}$, $\langle F(x^*), x - x^* \rangle \geq 0$. Under mild conditions (see Appendix B.2), x^* constitutes a Nash equilibrium point. For readers familiar with convex optimization, note the consistent similarity throughout this subsection for when $F = \nabla f$. In game theory, F often maps to the set of player gradients. For example, the map corresponding to the minimax game in Equation (3.1) is $F : \mathbb{R}^{|\theta|+|\phi|} \to [\nabla V_\theta; -\nabla V_\phi] \in \mathbb{R}^{|\theta|+|\phi|}$.

A map, F, is monotone [9] if $\langle F(x) - F(x'), x - x' \rangle \geq 0$ for all $x \in \mathcal{X}$ and $x' \in \mathcal{X}$. Alternatively, if the (possibly asymmetric) Jacobian matrix of F, $J(F)$, is positive semidefinite (PSD), then F is monotone [81, 100] where

Table 3.1: Existing convergence rates for VI algorithms in different settings.

	Strongly-Monotone	(Smooth/Sharp+)Monotone	Pseudomonotone
Deterministic	$\mathcal{O}(e^{-k})$ [19]	($\mathcal{O}(1/k)$ [18, 84]) $\mathcal{O}(1/\sqrt{k})$ [53]	$\mathcal{O}(1/\sqrt{k})$ [25]
Stochastic	$\mathcal{O}(1/k)$ [55]	($\mathcal{O}(1/k)$ [55, 120]) $\mathcal{O}(1/\sqrt{k})$ [53]	$\mathcal{O}(1/\sqrt{k})$ [50]

$$J(F) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \cdots & \frac{\partial F_n}{\partial x_n} \end{bmatrix}. \tag{3.2}$$

A matrix, J, is PSD if for all $x \in \mathbb{R}^n$, $x^\top Jx \geq 0$, or equivalently, J is PSD if $(J+J^\top) \succeq 0$.

As in convex optimization, a hierarchy of monotonicity exists. F is

$$\text{monotone iff} \quad \forall x \in \mathcal{X}, \forall x' \in \mathcal{X}, \langle F(x) - F(x'), x - x' \rangle \geq 0, \tag{3.3}$$

$$\text{pseudomonotone iff} \quad \forall x \in \mathcal{X}, \forall x' \in \mathcal{X}, \langle F(x'), x - x' \rangle \geq 0 \implies \langle F(x), x - x' \rangle \geq 0,$$

$$\text{and quasimonotone iff} \quad \forall x \in \mathcal{X}, \forall x' \in \mathcal{X}, \langle F(x'), x - x' \rangle > 0 \implies \langle F(x), x - x' \rangle \geq 0. \tag{3.4}$$

If, in Equation (3.3), "$\geq$" is replaced by "$>$", then F is strictly-monotone; if "$\geq$" is replaced by "$s||x - x'||^2$", then F is s-strongly-monotone. If F is a gradient, then replace monotone with convex.

Table 3.1 cites various Extragradient-type algorithms with convergence rates for several settings. Whereas gradient descent achieves optimal convergence rates for various convex optimization settings, Extragradient [61] achieves optimal rates for VIs. If we can prove that a map, $\tilde{F}$, associated with the game satisfies a known monotonicity property while maintaining the same fixed point as the original game, we need only look up the appropriate algorithm in this table to be able to solve for the equilibrium point of the game.

3.4.2 The ODE Method and Hurwitz Jacobians

Nagarajan and Kolter [2017] performed a *local* stability analysis of the gradient dynamics of Equation (3.1), proving that the Jacobian of F evaluated at x^* is Hurwitz[1] [15, 16, 58], i.e., the real parts of its eigenvalues are strictly positive. Assuming the dynamics are Lipschitz continuous, their finding means that if simultaneous gradient descent using a "square-summable, not summable" step sequence enters an ϵ-ball with a low enough step size, it will converge to the equilibrium. This guarantee applies only in the deterministic setting because stochastic gradients can cause the iterates to exit this ball and diverge. Note that while the real parts of eigenvalues reveal exponential growth or decay of trajectories, the imaginary parts reflect any rotation in the system[2].

The Hurwitz and monotonicity properties are complementary (see B.8). To summarize, Hurwitz encompasses dynamics with exponentially stable trajectories and with arbitrary rotation, while monotonicity includes cycles (Jacobians with purely imaginary eigenvalues) and is similar to convex optimization. Also note we are interested in these as global properties. This means that if a subset of $\mathcal{X}$ is not Hurwitz (or monotone), then the map is not Hurwitz (or monotone) globally and we cannot naturally guarantee convergence globally.

Given the preceding discussion, we believe VIs and monotone operator theory will serve as a strong foundation for deriving fundamental convergence results for GANs; this theory is

1. Similar to convexity suggesting its adoption by the GAN community should be smooth,

---

[1]Our definition of Hurwitz is equivalent to the more standard definition: $-J$ is Hurwitz if $\max_i[\mathrm{Re}(\lambda_i(-J))] < 0$.

[2]Linearized Dynamical System: $x(t) = \sum_i c_i v_i e^{\lambda_i t}$; Euler's formula: $e^{(a+ib)t} = e^{at}(\cos(bt) + i\sin(bt))$.

2. Mature with natural mechanisms for handling constraints, subdifferentials, and online scenarios,

3. Rich with algorithms with finite sample convergence for a hierarchy of monotone operators.

Finally, we suggest [102] for a lucid comparison of convex optimization, game theory, and VIs.

3.5 The Linear Quadratic GAN

In the Linear-Quadratic GAN, $g(x) = x$, and the generator and discriminator are restricted to be linear and quadratic respectively: $G(z) = Az + b$ and $D(y) = y^\top W_2 y + w_1^\top y$. Equation (3.1) becomes

$$\min_{A,b} \max_{W_2,w_1} \Big\{ V(W_2, w_1, A, b) = \mathbb{E}_{y\sim p(y)}[D(y)] - \mathbb{E}_{z\sim p(z)}[D(G(z)] \Big\}. \tag{3.5}$$

Let $\mathbb{E}[y] = \mu$, $\mathbb{E}[(y-\mu)^\top (y-\mu)] = \Sigma$, $\mathbb{E}[z] = 0$, and $\mathbb{E}[z^2] = I$. If A is constrained to be lower triangular with positive diagonal, i.e., of Cholesky form, then $(W_2^*, w_1^*, A^*, b^*) = (\mathbf{0}, \mathbf{0}, \Sigma^{1/2}, \mu)$ is the unique minimax solution (see Proposition 9). The majority of this chapter focuses on the case where $p(y)$ and $p(z)$ are 1-d distributions. Equation (3.5) simplifies to

$$\min_{a>0,b} \max_{w_2,w_1} \Big\{ V(w_2, w_1, a, b) = w_2(\sigma^2 + \mu^2 - a^2 - b^2) + w_1(\mu - b) \Big\}. \tag{3.6}$$

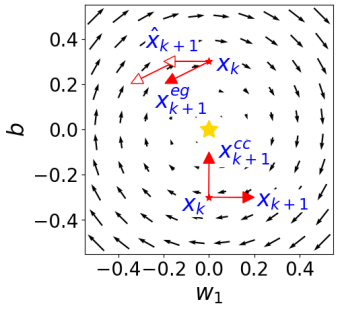
The map F naturally associated with this zero-sum game is constructed by concatenating the gradients of the two players' losses ($f_G = V, f_D = -V$):

$$F = \left[\tfrac{\partial f_D}{\partial w_2}, \tfrac{\partial f_D}{\partial w_1}, \tfrac{\partial f_G}{\partial a}, \tfrac{\partial f_G}{\partial b} \right]^\top = \left[a^2 + b^2 - \sigma^2 - \mu^2, \quad b - \mu, \quad -2w_2 a, \quad -2w_2 b - w_1 \right]^\top.$$

We say naturally because the unique fixed point of this system, $F(x^*) = 0$, occurs when both generator and discriminator gradients are zero—occurring at (W_2^*, w_1^*, A^*, b^*).

3.6 Crossing-the-Curl

In this section, we will derive our proposed technique, *Crossing-the-Curl*, motivated by an examination of the (w_1, b)-subsystem of LQ-GAN, i.e., (w_2, a) fixed at $(0, a_0)$ for any a_0. The results discussed here hold for the N-dimensional case as well. The map associated with this subsystem is plotted in Figure 3.2 and formally stated in Equation (3.7).



$$F^{w_1,b} = [b - \mu, -w_1]^\top \tag{3.7}$$

$$J^{w_1,b} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$x_k = [w_{1,k}, b_k]^\top$$

$$x_{k+1} = x_k - \rho_k F^{w_1,b}(x_k) \tag{3.8}$$

Figure 3.2: Vector field plot of $F^{w_1,b}$ for $\mu = 0$ with Extragradient, x^{eg}_{k+1} (see updates (3.9) and (3.10)), simultaneous gradient descent, x_{k+1}, and *Crossing-the-Curl*, x^{cc}_{k+1}, updates overlayed on top.

The Jacobian of $F^{w_1,b}$ is not Hurwitz, and simultaneous gradient descent, defined in Equation (3.8), will diverge for this problem (see B.5). However, $F^{w_1,b}$ is monotone $(J + J^\top = \mathbf{0})$ and $1-$Lipschitz in the sense that $||F^{w_1,b}(x) - F^{w_1,b}(x')||^2 \leq 1||x - x'||^2$. Table 3.1 offers an Extragradient method (see Figure 3.2) with an $\mathcal{O}(1/k)$ convergence rate, which is optimal for worst case monotone maps.

Nevertheless, an algorithm that travels perpendicularly to the vector field will proceed directly to the equilibrium. In this example, the intuition is to travel in the direction that is perpendicular to both F and the axis of rotation. For a 2-D system, the axis of rotation can be obtained by taking the curl of the vector field. To derive

a direction perpendicular to both F and the axis of rotation, we can take their cross product:

$$F_{cc} = -\frac{1}{2}(\overbrace{\nabla \times F}^{\text{curl}}) \times F = -\frac{1}{2}\{\nabla_F(v \cdot F) - (v \cdot \nabla)F\}\Big|_{v=F} = -\Big(\frac{J - J^\top}{2}\Big)F = \begin{bmatrix} w_1 \\ b - \mu \end{bmatrix}$$

where ∇_F is Feynman notation for the gradient with respect to F only and $|_{v=F}$ means evaluate the expression at $v = F$. The $^-1/_2$ factor ensures the algorithm moves toward regions of "tighter cycles" and simplifies notation. It may be sensible to perform some linear combination of simultaneous gradient descent and *Crossing-the-Curl*, so we will refer to $(I - \eta(J - J^\top))F$ as $F_{\eta cc}$.

Note that the fixed point of F_{cc} remains the same as the original field F. Furthermore, the reader may recognize F_{cc} as the gradient of the function $\frac{1}{2}(w_1^2 + (b - \mu)^2)$, which is strongly convex, allowing an $\mathcal{O}(e^{-k})$ convergence rate in the deterministic setting. F_{cc} is derived from intuition in 2-D, however, we discuss reasons in the next subsection for why this approach generalizes to higher dimensions.

3.6.1 Discussion and Relation to Other Methods

For the (w_1, b)-subsystem, *Crossing-the-Curl* is equivalent to two other methods: the consensus algorithm [74] and a Taylor series approximation to Extragradient [61]. Note that if we differentiate the path integral loss highlighted in Section 2.8.1 of the previous chapter, the first term in the path integral loss recovers Extragradient while the second term recovers the consensus algorithm.

These equivalences occur because the Jacobian is skew-symmetric ($J^\top = -J$) for the (w_1, b)-subsystem. In the more general case, where J is not necessarily skew-symmetric, *Crossing-the-Curl* represents a combination of the two techniques. Extragradient (EG) is key to solving VIs and the consensus algorithm has delivered

$$\hat{x}_{k+1} = x_k - \hat{\eta}F(x_k) \tag{3.9}$$

$$x_{k+1} = x_k - \eta F(\hat{x}_{k+1}) \tag{3.10}$$

$$= x_k - \eta \underbrace{(I - \hat{\eta}J(x_k))F(x_k)}_{F_{eg}} \tag{3.11}$$

$$+ \mathcal{O}(\eta\hat{\eta}^2)$$

$$x_{k+1} = x_k - \eta(F(x_k) + \hat{\eta}\nabla||F||^2)$$

$$= x_k - \eta \underbrace{(I + \hat{\eta}J^\top(x_k))F(x_k)}_{F_{con}} \tag{3.12}$$

Figure 3.3: A Taylor series expansion of Extragradient (3.11) and the consensus algorithm (3.12).

impressive results for GANs, so this is promising for F_{cc}. To our knowledge, F_{eg} is novel and has not appeared in the Variational Inequality literature.

Crossing-the-Curl stands out in many ways though. Observe that in higher dimensions, the subspace orthogonal to F is $(n-1)$ dimensional, which means $(J^\top - J)F$ is no longer the unique direction orthogonal to F. However, every matrix can be decomposed into a symmetric part with real eigenvalues, $1/2(J + J^\top)$, and a skew-symmetric part with purely imaginary eigenvalues, $1/2(J - J^\top)$. Notice that for an optimization problem, $J - J^\top = H - H^\top = 0$ where H is the Hessian.[3] It is the imaginary eigenvalues, i.e., rotation, that set equilibrium problems apart from optimization and necessitate the development of new algorithms like Extragradient. It is reassuring that this matrix appears explicitly in F_{cc}. In addition, F_{cc} reduces to gradient descent when applied to an optimization problem making the map agnostic to the type of problem at hand: optimization or equilibration.

The curl also shares close relation to the gradient. The gradient is ∇ applied to a scalar function and the curl is ∇ crossed with a vector function. Furthermore, under mild conditions, every vector field, $F : \mathbb{R}^3 \to \mathbb{R}^3$, admits a Helmholdtz decomposition: $F = -\nabla f + \nabla \times G$ where f is a scalar function and G is a vector function suggesting the gradient and curl are both fundamental components.

[3]Assuming the objective function has continuous second partial derivatives—see Schwarz's theorem.

Consider the perspective of F_{cc} as preconditioning F by a skew-symmetric matrix. Preconditioning with a positive definite matrix dates back to Newton's method and has reappeared in machine learning with natural gradient [5]. Dafermos [1983] considered asymmetric positive definite preconditioning matrices for VIs. Thomas [2014] extended the analysis of natural gradient to PSD matrices. We are not aware of any work using skew-symmetric matrices for preconditioning. The scalar $x^\top Ax \equiv 0$ for any skew-symmetric matrix A, so calling $(J^\top - J)$ a PSD matrix is not adequately descriptive.

Note that *Crossing-the-Curl* does not always improve convergence; this technique can transform a strongly-monotone field into a saddle and an unstable fixed point (non-monotone) into a strongly-monotone field (see B.9 for examples), so this technique should generally be used with caution.

Lastly, *Crossing-the-Curl* is inexpensive to compute. The Jacobian-vector product, JF, can be approximated accurately and efficiently with finite differences. Likewise, $J^\top F$ can be computed efficiently with double backprop [30] by taking the gradient of $1/2||F||^2$. In total, three backprops are required, one for $F(x_k)$, one for $F(\hat{x}_{k+1})$, and one for $1/2||F(x_k)||^2$.

In our analysis, we also consider the gradient regularization proposed in [80], F_{reg}, the Unrolled GAN proposed in [76], F_{unr}, alternating gradient descent, F_{alt}, as well as any linear combination of F, JF, and $J^\top F$, deemed F_{lin}, which forms a family of maps that includes F_{eg}, F_{con}, and F_{cc}:

$$F_{reg} = \begin{bmatrix} F_D; & F_G + \eta \nabla_G ||F_D||^2 \end{bmatrix}^\top, \qquad\qquad F_{lin} = (\rho I + \beta J^\top - \gamma J)F.$$

Keep in mind that we are proposing F_{lin} as a generalization of *Crossing-the-Curl*. We state our main results here for the (w_1, b)-subsystem.

Proposition 1. *For any α, $F_{lin}^{w_1,b}$ with at least one of β and γ positive and both non-negative is strongly monotone. Also, its Jacobian is Hurwitz. See Proposition 13.*

Corollary 1. *$F_{cc}^{w_1,b}$, $F_{\eta cc}^{w_1,b}$, $F_{eg}^{w_1,b}$, and $F_{con}^{w_1,b}$ with $\eta > 0$ are strongly-monotone with Hurwitz Jacobians. See Proposition 1.*

Proposition 2. *$F_{alt}^{w_1,b}$, $F_{unr}^{w_1,b}$, $F^{w_1,b}$, and $F_{reg}^{w_1,b}$ with any η are monotone, but not strictly monotone. Of these maps, only $F_{reg}^{w_1,b}$'s Jacobian is Hurwitz. See Propositions 12 and 13.*

3.7 Analysis of the Full System

Here, we analyze the maps for each of the algorithms discussed above, testing for quasimonotonicity (the weakest monotone property) and whether the Jacobian is Hurwitz for the full LQ-GAN system.

Proving quasiconvexity of 4th degree polynomials has been proven strongly NP-Hard [3]. This implies that proving monotonicity of 3rd degree maps is strongly NP-Hard. The original F contains quadratic terms suggesting it may welcome a quasimonotone analysis, however, the remaining maps all contain 3rd degree terms. Unsurprisingly, analyzing quasimonotonicity for F_{lin} represents the most involved of our proofs given in Appendix B.11.

The definition stated in (3.4) suggests checking the truth of an expression depending on four separate variables: x, x', y, y'. While we used this definition for certain cases, the following alternate requirements proposed in the work of Crouzeix and Ferland [1996] made the complete analysis of the system tractable. We restate simplified versions of the requirements we leveraged for convenience.

Consider the following conditions:

(A) For all $x \in \mathcal{X}$ and $v \in \mathbb{R}^n$ such that $v^\top F(x) = 0$ we have $v^\top J(x) v \geq 0$.

(B) For all $x \in \mathcal{X}$ and $x^* \in \mathcal{X}$ such that $F(x^*) = 0$, we have that $F(x)^\top (x - x^*) \geq 0$.

Theorem 3 ([22], Theorem 3)**.** *Let $F : \mathcal{X} \to \mathbb{R}^n$ be differentiable on the open convex set $\mathcal{X} \subset \mathbb{R}^n$.*

1. *F is quasimonotone on $\mathcal{X}$ only if (A) holds, i.e. (A) is necessary but not sufficient.*

2. *F is pseudomonotone on $\mathcal{X}$ if (A) and (B) hold, i.e. (A) and (B) are sufficient but not necessary.*

Condition (A) says that for a map to be quasimonotone, the map must be monotone along directions orthogonal to the vector field. In addition to this, condition (B) says that for a map to be pseudomonotone, the dynamics, $-F$, must not be leading away from the equilibrium anywhere.

Equipped with these definitions, we can conclude the following:

Proposition 3. *None of the maps, including F_{lin} with any setting of coefficients, is quasimonotone for the full LQ-GAN. See Corollary 5 and Propositions 15 through 17.*

Proposition 4. *None of the maps, including F_{lin} with any setting of coefficients, has a Hurwitz Jacobian for the full LQ-GAN. See Propositions 27 and 15 through 17.*

3.7.1 Learning the Variance: The (w_2, a)-Subsystem

Results from the previous section suggest that we cannot solve the full LQ-GAN, but given that we can solve the (w_1, b)-subsystem, we shift focus to the (w_2, a)-subsystem assuming the mean has already been learned exactly, i.e., $b = \mu$. We will revisit this assumption later.

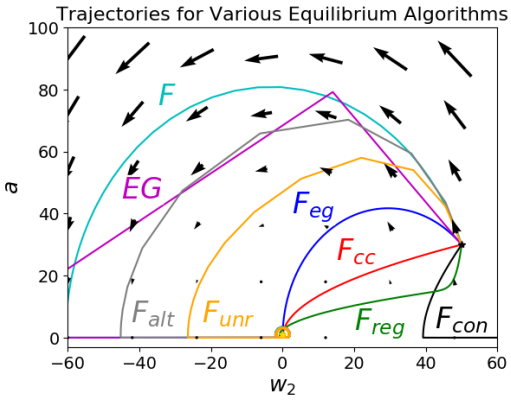
We can conclude the following for the (w_2, a)-subsystem:

Proposition 5. *$F^{w_2,a}$, $F^{w_2,a}_{reg}$, $F^{w_2,a}_{unr}$, $F^{w_2,a}_{alt}$, and $F^{w_2,a}_{con}$ are not quasimonotone. Also, their Jacobians are not Hurwitz. See Propositions 14 through 19.*

Proposition 6. *$F^{w_2,a}_{eg}$ and $F^{w_2,a}_{cc}$ are pseudomonotone which implies an $\mathcal{O}(1/\sqrt{k})$ stochastic convergence rate. See Propositions 21 and 24. Their Jacobians are not Hurwitz. See Proposition 27.*

Proposition 7. *No monotone $F_{lin}^{w_2,a}$ exists. See Proposition 26.*

These results are not purely theoretical. Figure 3.4 displays trajectories resulting from each of the maps.



$$F_{eg}^{w_2,a} = \begin{bmatrix} 4w_2a^2 \\ 2a(a^2-\sigma^2) - 4w_2^2a \end{bmatrix} \tag{3.13}$$

$$\downarrow *^{1}/_{4a^2}$$

$$F_{eg'}^{w_2,a} = \begin{bmatrix} w_2 \\ \frac{a^2-\sigma^2-2w_2^2}{2a} \end{bmatrix}; \tag{3.14}$$

$$F_{cc}^{w_2,a} = \begin{bmatrix} 4w_2a^2 \\ 2a(a^2-\sigma^2) \end{bmatrix} \tag{3.15}$$

$$\downarrow *^{1}/_{4a^2}$$

$$F_{cc'}^{w_2,a} = \begin{bmatrix} w_2 \\ \frac{a^2-\sigma^2}{2a} \end{bmatrix} \tag{3.16}$$

Figure 3.4: (Left) Comparison of trajectories on the (w_2, a)-subsystem.[4] The vector field plotted is for the original system, $\dot{x} = -F^{w_2,a}(x)$. Observe how $F_{cc}^{w_2,a}$ takes a more direct route to the equilibrium. (Right) Maps derived after rescaling $F_{cc}^{w_2,a}$ and $F_{eg}^{w_2,a}$.

We can further improve upon $F_{eg}^{w_2,a}$ and $F_{cc}^{w_2,a}$ by rescaling with $^{1}/_{4a^2}$: (3.13)→(3.14) and (3.15)→(3.16) respectively. This results in strongly-monotone and strongly-convex systems respectively, improving the stochastic convergence rate to $\mathcal{O}(1/k)$. In deriving these results, we assumed the mean was given. We can relax this assumption and analyze the (w_2, a)-subsystem under the assumption that the mean is "close enough". Using a Hoeffding bound, we find that $k > \left(\frac{y_{hi}-y_{low}}{-|\mu|+\sqrt{\mu^2+d\sigma^2}}\right)^2 \log[\frac{\sqrt{2}}{\delta^{1/2}}]$ iterations of $F_{cc}^{w_1,b}$ are required to achieve a $1-\delta$ probability of the mean being accurate enough to ensure the (w_2, a)-subsystem is strongly-monotone. Note that this approach of first learning the mean, then the variance retains the overall $\mathcal{O}(1/k)$ stochastic rate. We summarize the main points here.

[4]ODEs were simulated using Heun-Euler with Phase Space Error Control [44].

Claim 1. *A nonlinear scaling of $F_{eg}^{w_2,a}$ and $F_{cc}^{w_2,a}$ results in strictly monotone and $1/2$-strongly monotone subsystems respectively. See Proposition 29.*

Claim 2. *If the mean is first well approximated, i.e., $b^2 \leq \mu^2+\sigma^2$, then $F_{cc'}^{w_2,a}$ remains 1) $1/2$-strongly-monotone if the (w_1, b)-subsystem is "shut off" or 2) strictly-monotone if the (w_1, b)-subsystem is re-weighted with a high coefficient. See Propositions 30 and 31.*

Proposition 8. *$F_{eg}^{W_2,A}$ and $F_{cc}^{W_2,A}$ are not quasimonotone for the 2-D LQ-GAN system (with and without $(AA^\top)^{-1}$ scaling). See Proposition 32.*

Several takeaways emerge. One is that the stability of the system is highly dependent on the mean first being learned. In other words, *batch norm* is required for the monotonicity of LQ-GAN, so it is not surprising that GANs typically fail without these specialized layers.

Second is that stability is achieved by first learning a simple subsystem, (w_1, b), then learning the more complex, (w_2, a)-subsystem. This theoretically confirms the intuition behind progressive training of GANs [57], which have generated the highest quality images to date. Note that the work by Karras et al. [2017] was inspired by ideas described in Chapter 4 of this thesis.

Thirdly, because $J_{w_2,a}^{cc'}$ is symmetric (and $\succ 0$), we can integrate $F_{w_2,a}^{cc'}$ to discover the convex function it is implicitly descending via gradient descent: $f_{w_2,a}^{cc'} = 1/2[(a^2 - \sigma^2) - \sigma^2 \log(a^2/\sigma^2)]$. Compare this to KL-divergence: $KL(\sigma||a) = 1/2[(\sigma^2/a^2) + \log(a^2/\sigma^2) - 1]$. In contrast to KL, $f_{w_2,a}^{cc'}$ is convex in a and may be a desirable alternative due to less extreme gradients near $a = 0$.

3.7.2 Learning the Covariance: The (W_2, A)-Off-Diagonal Subsystem

After learning both the mean and variance of each dimension, the covariance of separate dimensions can be learned. Proposition B.14 in the Appendix states that the subsystem relevant to learning each row of A is strictly monotone when

Table 3.2: For convenience, we summarize many of our theoretical results in this table. Legend: M=Monotone, C=Convex, H=Hurwitz, S=Strongly, s=Strictly, P=Pseudo, Q=Quasi, /=Not.

Subsystem	F	F_{alt}	F_{unr}	F_{reg}	F_{con}	F_{eg}	F_{cc}	$F_{eg'}$	$F_{cc'}$
(w_1, b)	M,~~H~~	M,~~H~~	M,~~H~~	M,H	SC,H	SC,H	SC,H	NA	NA
(w_2, a)	~~QM~~,~~H~~	~~QM~~,~~H~~	~~QM~~,~~H~~	~~QM~~,~~H~~	~~QM~~,~~H~~	PM,~~H~~	PM,~~H~~	sM,H	SC,H

all other rows are held fixed. In fact, the maps for these subsystems are affine and skew-symmetric just like the (w_1, b)-subsystem. This implies that *Crossing-the-Curl* applied successively to each row of A can solve for A^*; pseudocode is presented in Algorithm 5. Note that this procedure is reminiscent of the Cholesky-Banachiewicz algorithm which computes A row by row, beginning with the first row. The resulting algorithm is $\mathcal{O}(N/k)$.

Algorithm 5 *Crossing-the-Curl* for LQ-GAN

Input: Sampling distribution $p(y)$, max iterations K, batch size B, lower bound on variance $\sigma_{\min}$

(1) Learn Mean

$\mu_0 = [0, \dots, 0]^\top$

for all $k = 1, 2, \dots, K$ **do**

$\hat{\mu} = \frac{1}{B}\sum_{s=1}^{B}(y_s \sim p(y))$

$\mu_k = \frac{k}{k+1}\mu_{k-1} + \frac{1}{k+1}\hat{\mu}$, i.e., $\mu_k = \mu_{k-1} - \rho_k F^b_{cc}$ with step size $\rho_k = \frac{1}{k+1}$

end for

(2) Learn Variance

$\sigma_0 = [1, \dots, 1]^\top$

for all $k = 1, 2, \dots, K$ **do**

$\hat{\sigma}^2 = \frac{1}{B}\sum_{s=1}^{B}[(y_s \sim p(y)) - \mu_K]^2$

$F^a_{cc'} = (\sigma_k^2 - \hat{\sigma}^2)/(2\sigma_k)$

$\sigma_k = \texttt{clip}(\sigma_{k-1} - \frac{1}{k+1}F^a_{cc'}, \sigma_{\min}, \infty)$

end for

(3) Learn Covariance

$A_0 = LT(I_N)$, i.e., lower triangular part of identity matrix

$A_{0,11} = \sigma_{K,1}$

for all $d = 2, \dots, N$ **do**

for all $k = 1, 2, \dots, K$ **do**

$y_s \sim p(y),\ s = 1, \dots, B$

$\hat{\Sigma} = \frac{1}{B}\sum_{s=1}^{B}(y_s - \mu_K)^\top(y_s - \mu_K)$

$F_{W_{i<d}} = 2\big(\sum_{j\le i} A_{k-1,ij}A_{k-1,dj} - \hat{\Sigma}_{id}\big)$

$F^A_{cc} = A^\top_{k-1,:d-1}F_{W_{i<d}}$ where $A_{k-1,:d-1}$ refers to the top left $d-1 \times d-1$ block of A_{k-1}

$\hat{A}_{k,d:} = A_{k-1,d:} - \frac{1}{k+1}F^A_{cc}$ where $A_{k-1,d:}$ refers to the dth row of A_k excluding the diagonal

if $\sum_j \hat{A}^2_{k,dj} > \sigma^2_{K,d} - \sigma^2_{\min}$ **then**

$\hat{A}_{k,dj} = \hat{A}_{k,dj} \cdot \sigma_{K,d}/\sqrt{\sum_j \hat{A}^2_{k,dj} + \sigma^2_{\min}}$

end if

$F_{W_{i<d}} = 2\big(\sum_{j\le i} A_{k-1,ij}\hat{A}_{k,dj} - \hat{\Sigma}_{id}\big)$

$F^A_{cc} = A^\top_{k-1,:d-1}F_{W_{i<d}}$ where $A_{k-1,:d-1}$ refers to the top left $d-1 \times d-1$ block of A_{k-1}

$A_{k,d:} = A_{k-1,d:} - \frac{1}{k+1}F^A_{cc}$ where $A_{k-1,d:}$ refers to the dth row of A_k excluding the diagonal

if $\sum_j A^2_{k,dj} > \sigma^2_{K,d} - \sigma^2_{\min}$ **then**

$A_{k,dj} = A_{k,dj} \cdot \sigma_{K,d}/\sqrt{\sum_j A^2_{k,dj} + \sigma^2_{\min}}$

end if

end for

$A_{K,dd} = \sqrt{\sigma^2_{K,d} - \sum_j A^2_{K,dj}}$

end for

3.8 Experiments

Our theoretical analysis proves convergence of the stagewise procedure using *Crossing-the-Curl* for the N-d LQGAN. Experiments solving the (w_2, a)-subsystem alone for randomly generated $\mathbb{E}[(y-\mu)^2] = \sigma^2$ support the analysis of Subsection 3.7.1—see the first row of Table 3.3. Not listed in the first row of the table are $F_{cc'}$ and $F_{eg'}$ which converge in 32 and 33 steps on average respectively with a constant step size of 0.1. Our novel maps, F_{cc} and F_{eg}, converge in a quarter of the iterations of the next best method (F_{reg}), and $F_{cc'}$ and $F_{eg'}$ in nearly a quarter of their parent counterparts. These experiments used analytical results of the expectations, i.e., the systems are deterministic.

Table 3.3: Each entry in the table reports two quanities. First is the average number of steps, k, required for each dynamical system, e.g., $\dot{x} = -F(x)$, to reduce $||x_k - x^*||/||x_0 - x^*||$ to 0.001 for the (W_2, A)-subsystem. The second, in parentheses, reports the fraction of trials that the algorithm met this threshold in under 100,000 iterations. Dim denotes the dimensionality of $x \sim p(x)$ for the LQ-GAN being trained (with $|\theta| + |\phi|$ in parentheses). For each problem, x_0 is randomly initialized 10 times for each of ten randomly initialized Σ's, i.e., 100 trials per cell. Extragradient (EG) is run with a fixed step size. All other ODEs are solved via Heun-Euler with Phase Space Error Control [44].

Dim	F	EG	F_{con}	F_{reg}	F_{eg}	F_{cc}
1 (2)	10^5 (0)	83315 (0.4)	6354 (0.94)	395 (1)	116 (1)	**110** (1)
2 (6)	10^5 (0)	98244 (0.05)	33583 (0.68)	2595 (1)	**1321** (1)	1441 (1)
4 (10)	10^5 (0)	99499 (0.01)	77589 (0.23)	**33505** (0.7)	34929 (0.67)	34888 (0.68)

The second and third rows of the table reveal that convergence slows considerably for higher dimensions. However, the stagewise procedure discussed in Subsection 3.7.2 is guaranteed to converge. This procedure solves the 4-d *deterministic* LQ-GAN in **20549** iterations with a **0.88** success rate. For the 4-d *stochastic* LQ-GAN using single-sample minibatch estimates, this procedure achieves $||x_k - x^*||/||x_0 - x^*|| < 0.1$ in 100,000 iterations with a 0.75 success rate.

3.9 Conclusion

In this chapter, we performed the first global convergence analysis for a variety of GAN training algorithms. According to Variational Inequality theory, none of the current GAN training algorithms is globally convergent for the LQ-GAN. We proposed an intuitive technique, *Crossing-the-Curl*, with the first global convergence guarantees for any generative adversarial network. As a by-product of our analysis, we extract high-level explanations for why the use of *batch norm* and progressive training schedules for GANs are critical to training. In experiments with the multivariate LQ-GAN, *Crossing-the-Curl* achieves performance superior to any existing GAN training algorithm.

3.9.1 Up Next

In this chapter, we showed that by applying *Crossing-the-Curl* first to the problem of learning the mean of a distribution, then to learning the covariance in successively higher dimensional equilibrium problems, we were able to ensure global convergence. This concept of training the GAN using discriminators of varying and increasing levels of complexity supports our discussion in the next chapter.

CHAPTER 4

GENERATIVE MULTI-ADVERSARIAL NETWORKS

4.1 Purpose of Research

GANs are theoretically formulated as a search for the minimax optimal generator, the generator that achieves the minimal loss with respect to the best discriminator. In practice, GAN training typically consists of optimizing a single generator and a single discriminator simultaneously. This means that the discriminator is nearly always suboptimal. Furthermore, once training has reached an equilibrium, we can only trust that the generator achieves the minimal loss with respect to the best discriminator in a local neighborhood. Unfortunately, the most obvious alternative, training the discriminator to convergence before each generator update, results in a discriminator that provides very little training signal to the generator. We would like a tractable technique for obtaining a generator that is closer to minimax optimal.

In search of this goal, we pit the single generator against several discriminators. Intuitively, if the generator "fools" a diverse set of discriminators, we can be more confident that the generator is minimax optimal. We explore various ways of presenting the discriminator training signal to the generator and find that simply averaging the discriminator training signals leads to many performance benefits.

Spoiler: We present a simple extension to GANs that incorporates multiple discriminators. We show that introducing more discriminators into the standard GAN framework reduces variance of the minimax objective, improves the quality of the resulting samples that are generated, and accelerates convergence of the GAN to a steady-state minimax loss.

4.2 Introduction

In this chapter, we theoretically and empirically justify generalizing the GAN framework to multiple discriminators. We review GANs and summarize our extension in Section 4.3. In Section 4.4, we present our N-discriminator extension to the GAN framework (*Generative Multi-Adversarial Networks*). Section 4.4.2 explains how this extension makes training with the untampered minimax objective tractable. In Section 4.5, we define an intuitive metric (GMAM) to quantify GMAN performance and evaluate our framework on a variety of image generation tasks. Section 4.6 concludes with a summary of our contributions and directions for future research.

Contributions—To summarize, our main contributions are: **i**) a multi-discriminator GAN framework, GMAN, that allows training with the original, untampered minimax objective; **ii**) a generative multi-adversarial metric (GMAM) to perform pairwise evaluation of separately trained frameworks; **iii**) a particular instance of GMAN, GMAN*, that allows the generator to automatically regulate training and reach higher performance (as measured by GMAM) in a fraction of the training time required for the standard GAN model.

4.3 Generative Adversarial Networks to GMAN

The original formulation of a GAN is a minimax game between a generator, $G_\theta(z) : z \to x$, and a discriminator, $D_\omega(x) : x \to [0, 1]$,

$$\min_G \max_{D \in \mathcal{D}} V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} \Big[\log(D(x)) \Big] + \mathbb{E}_{z \sim p_z(z)} \Big[\log(1 - D(G(z))) \Big], \quad (4.1)$$

where $p_{data}(x)$ is the true data distribution and $p_z(z)$ is a simple (usually fixed) distribution that is easy to draw samples from (e.g., $\mathcal{N}(0, 1)$). We differentiate between the function space of discriminators, $\mathcal{D}$, and elements of this space, D. Let $p_G(x)$ be the distribution induced by the generator, $G_\theta(z)$. We assume D, G to be deep neural networks as is typically the case.

In their original work, Goodfellow et al. [2014] proved that given sufficient network capacities and an oracle providing the optimal discriminator, $D^* = \arg\max_{\mathcal{D}} V(D, G)$, gradient descent on $p_G(x)$ will recover the desired globally optimal solution, $p_G(x) = p_{data}(x)$, so that the generator distribution exactly matches the data distribution. In practice, they replaced the second term, $\log(1 - D(G(z)))$, with $-\log(D(G(z)))$ to enhance gradient signals at the start of the game; note this is no longer a zero-sum game. Part of their convergence and optimality proof involves using the oracle, D^*, to reduce the minimax game to a minimization over G only:

$$\min_G V(D^*, G) = \min_G \Big\{ C(G) = -\log(4) + 2 \cdot JSD(p_{data} || p_G) \Big\} \tag{4.2}$$

where JSD denotes Jensen-Shannon divergence. Minimizing $C(G)$ necessarily minimizes JSD, however, we rarely know D^* and so we instead minimize $V(D, G)$, which is only a lower bound.

4.3.1 GMAN: A Multi-adversarial Extension

We propose introducing multiple discriminators, which brings with it a number of design possibilities. We explore approaches ranging between two extremes: 1) a more discriminating D (better approximating $\max_{\mathcal{D}} V(D, G)$) and 2) a D better matched to the generator's capabilities. Approach 1 failed to produce good results—it has been relegated to the appendix. We describe approach 2 below. Mathematically, we reformulate G's objective as $\min_G \max F(V(D_1, G), \ldots, V(D_N, G))$ for different choices of F (see Figure 4.1). Each D_i is still expected to independently maximize its own $V(D_i, G)$, i.e. there is no explicit cooperation. We sometimes abbreviate $V(D_i, G)$ with V_i and $F(V_1, \ldots, V_N)$ with $F_G(V_i)$.

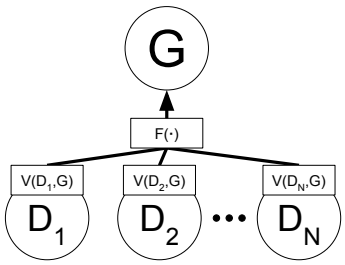


Figure 4.1: (GMAN) The generator trains using feedback aggregated over multiple discriminators. If $F \equiv \max$, G trains against the best discriminator. If $F \equiv \texttt{mean}$, G trains against an ensemble. We explore other alternatives to F in Subsections 4.4.1 and 4.4.3 that improve on both these options.

4.4 A Forgiving Teacher

This section focuses on the perspective that asks the question, "Is $\max_{\mathcal{D}} V(D, G)$ too harsh a critic?"

4.4.1 *Soft*-Discriminator

In practice, training against a far superior discriminator can impede the generator's learning. This is because the generator is unlikely to generate any samples considered "realistic" by the discriminator's standards, and so the generator will receive uniformly negative feedback. This is problematic because the information contained in the gradient derived from negative feedback only dictates where to drive down $p_G(x)$, not specifically where to increase $p_G(x)$. Furthermore, driving down $p_G(x)$ necessarily increases $p_G(x)$ in other regions of $\mathcal{X}$ (to maintain $\int_{\mathcal{X}} p_G(x) = 1$) which may or may not contain samples from the true dataset (*whack-a-mole* dilemma). In contrast, a generator is more likely to see positive feedback against a more lenient discriminator, which may better guide a generator towards amassing $p_G(x)$ in approximately correct regions of $\mathcal{X}$.

For this reason, we explore a variety of functions that allow us to *soften* the max operator. We choose to focus on soft versions of the three classical Pythagorean means parameterized by λ where $\lambda = 0$ corresponds to the mean and the max is recovered

as $\lambda \to \infty$:

$$\mathrm{AM}_{soft}(V, \lambda) = \sum_{i}^{N} w_i V_i \tag{4.3}$$

$$\mathrm{GM}_{soft}(V, \lambda) = -\exp\Big(\sum_{i}^{N} w_i \log(-V_i)\Big) \tag{4.4}$$

$$\mathrm{HM}_{soft}(V, \lambda) = \Big(\sum_{i}^{N} w_i V_i^{-1}\Big)^{-1} \tag{4.5}$$

where $w_i = e^{\lambda V_i}/\Sigma_j e^{\lambda V_j}$ with $\lambda \geq 0, V_i < 0$. Using a *softmax* also has the well known advantage of being differentiable (as opposed to subdifferentiable for max). Note that we only require continuity to guarantee that computing the *softmax* is actually equivalent to computing $V(\tilde{D}, G)$ where $\tilde{D}$ is some convex combination of D_i (see Appendix C.2).

4.4.2 Using the Original Minimax Objective

To illustrate the effect the *softmax* has on training, observe that the component of $AM_{soft}(V, 0)$ relevant to generator training can be rewritten as

$$\frac{1}{N}\sum_{i}^{N} \mathbb{E}_{x\sim p_G(x)}\Big[\log(1 - D_i(x))\Big] = \frac{1}{N}\mathbb{E}_{x\sim p_G(x)}\Big[\log(z)\Big]. \tag{4.6}$$

where $z = \prod_i^N (1 - D_i(x))$. Note that the generator gradient, $|\frac{\partial \log(z)}{\partial z}|$, is minimized at $z = 1$ over $z \in (0, 1]$[1]. From this form, it is clear that $z = 1$ if and only if $D_i = 0 \,\forall i$, so G only receives a vanishing gradient if all D_i agree that the sample is fake; this is especially unlikely for large N. In other words, G only needs to fool a single D_i to receive constructive feedback. This result allows the generator to successfully minimize the original generator objective, $\log(1 - D)$. This is in contrast to the more popular $-\log(D)$ introduced to artificially enhance gradients at the start of training.

[1] $\nabla_G V = -\sum_i \frac{D_i}{z}\frac{\partial D_i}{\partial G}\prod_{j\neq i}(1 - D_j) = -\frac{1}{z}\frac{\partial D_k}{\partial G}$ for $D_k = 1, D_{\neq k} = 0$. Our argument ignores $\frac{\partial D_k}{\partial G}$.

At the beginning of training, when $\max_{D_i} V(D_i, G)$ is likely too harsh a critic for the generator, we can set λ closer to zero to use the mean, increasing the odds of providing constructive feedback to the generator. In addition, the discriminators have the added benefit of functioning as an ensemble, reducing the variance of the feedback presented to the generator, which is especially important when the discriminators are far from optimal and are still learning a reasonable decision boundary. As training progresses and the discriminators improve, we can increase λ to become more critical of the generator for more refined training.

4.4.3 Automating Regulation

The problem of keeping the discriminator and generator in balance has been widely recognized in previous work with GANs. Issues with unstable dynamics, oscillatory behavior, and generator collapse are not uncommon. In addition, the discriminator is often times able to achieve a high degree of classification accuracy (producing a single scalar) before the generator has made sufficient progress on the arguably more difficult generative task (producing a high dimensional sample). Salimans et al. [2016] suggested label smoothing to reduce the vulnerability of the generator to a relatively superior discriminator. Here, we explore an approach that enables the generator to automatically temper the performance of the discriminator when necessary, but still encourages the generator to challenge itself against more accurate adversaries. Specifically, we augment the generator objective:

$$\min_{G, \lambda \in (0, \lambda_{\max})} F_G(V_i) - f(\lambda) \tag{4.7}$$

where $f(\lambda)$ is monotonically increasing in λ which appears in the *softmax* equations, (4.3)—(4.5). In experiments, we simply set $f(\lambda) = c\lambda$ with c a constant (e.g., 0.001). The generator is incentivized to increase λ to reduce its objective at the expense of competing against the best available adversary D^* (see Appendix C.3).

4.5 Evaluation

Evaluating GANs is still an open problem. In their original work, Goodfellow et al. [2014] report log likelihood estimates from Gaussian Parzen windows, which they admit, has high variance and is known not to perform well in high dimensions. Theis et al. [2016] recommend avoiding Parzen windows and argue that generative models should be evaluated with respect to their intended application. Salimans et al. [2016] suggest an *Inception score*, however, it assumes labels exist for the dataset. Recently, Im et al. [2016] introduced the Generative Adversarial Metric (GAM) for making pairwise comparisons between independently trained GAN models. The core idea behind their approach is given two generator, discriminator pairs (G_1, D_1) and (G_2, D_2), we should be able to learn their relative performance by judging each generator under the opponent's discriminator.

4.5.1 Metric

In GMAN, the opponent may have multiple discriminators, which makes it unclear how to perform the swaps needed for GAM. We introduce a variant of GAM, the generative multi-adversarial metric (GMAM), that is amenable to training with multiple discriminators,

$$\text{GMAM} = \log\Big(\frac{F^a_{G_b}(V^a_i)}{F^a_{G_a}(V^a_i)} \Big/ \frac{F^b_{G_a}(V^b_i)}{F^b_{G_b}(V^b_i)}\Big). \tag{4.8}$$

where a and b refer to the two GMAN variants (see Section 4.3.1 for notation $F_G(V_i)$). The idea here is similar. If G_2 performs better than G_1 with respect to both D_1 and D_2, then GMAM>0 (remember $V \leq 0$ always). If G_1 performs better in both cases, GMAM<0, otherwise, the result is indeterminate.

4.5.2 Experiments

We evaluate the aforementioned variations of GMAN on a variety of image generation tasks: MNIST [64], CIFAR-10 [63] and CelebA [69]. We focus on rates of convergence to steady state along with quality of the steady state generator according to the GMAM metric. To summarize, loosely in order of increasing discriminator leniency, we compare

- F-boost: A single *AdaBoost.OL*-boosted discriminator (see Appendix C.4).
- P-boost: D_i is trained according to *AdaBoost.OL*. A max over the weak learner losses is presented to the generator instead of the boosted prediction (see Appendix C.4).
- GMAN-max: $\max\{V_i\}$ is presented to the generator.
- GAN: Standard GAN with a single discriminator (see Appendix C.0.2).
- mod-GAN: GAN with modified objective (generator minimizes $-\log(D(G(z)))$.
- GMAN-λ: GMAN with $F \equiv$arithmetic *softmax* with parameter λ.
- GMAN*: The arithmetic *softmax* is controlled by the generator through λ.

All generator and discriminator models are deep (de)convolutional networks [92], and aside from the boosted variants, all are trained with Adam [59] and batch normalization [48]. Discriminators convert the real-valued outputs of their networks to probabilities with *squashed*-sigmoids to prevent saturating logarithms in the minimax objective ($\epsilon + \frac{1-2\epsilon}{1+e^{-z}}$). See Appendix C.5 for further details. We test GMAN systems with $N = \{2, 5\}$ discriminators. We maintain discriminator diversity by varying dropout probability and network depth.

Figure 4.2 reveals that increasing the number of discriminators reduces the number of iterations to steady-state by 2x on MNIST; increasing N (the size of the

discriminator ensemble) also has the added benefit of reducing the variance the minimax objective over runs. Figure 4.3 displays the variance of the same objective over a sliding time window, reaffirming GMAN's acceleration to steady-state. Figure 4.4 cor-

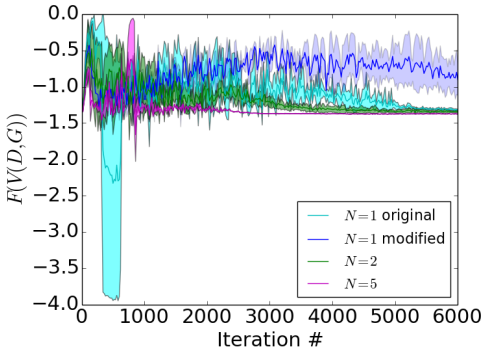


Figure 4.2: Generator objective, F, averaged over 5 training runs on MNIST. Increasing the number of discriminators accelerates convergence of F to steady state (solid line) and reduces its variance, σ^2 (filled shadow $\pm 1\sigma$). Figure 4.3 provides alternative evidence of GMAN*'s accelerated convergence.

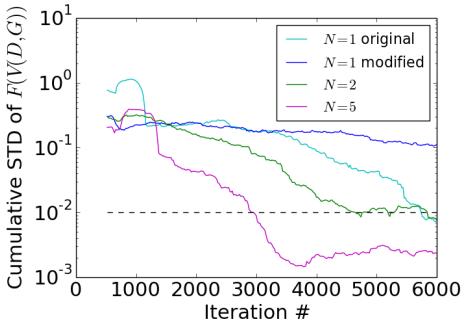


Figure 4.3: *Stdev*, σ, of the generator objective over a sliding window of 500 iterations. Lower values indicate a more steady-state. GMAN* with $N = 5$ achieves steady-state at $\approx$2x speed of GAN ($N = 1$). Note Figure 4.2's filled shadows reveal *stdev* of F over runs, while this plot shows *stdev* over iterations.

roborates this conclusion with recognizable digits appearing approximately an epoch before the single discriminator run; digits at steady-state appear slightly sharper as well.

Our GMAM metric (see Table 4.1) agrees with the relative quality of images in Figure 4.4 with GMAN* achieving the best overall performance. Figure 4.5 reveals GMAN*'s attempt to regulate the difficulty of the game to accelerate learning. Figure 4.6 displays the GMAM scores comparing fixed λ's to the variable λ controlled by GMAN*.

We see similar accelerated convergence behavior for the CelebA dataset in Figure 4.7.

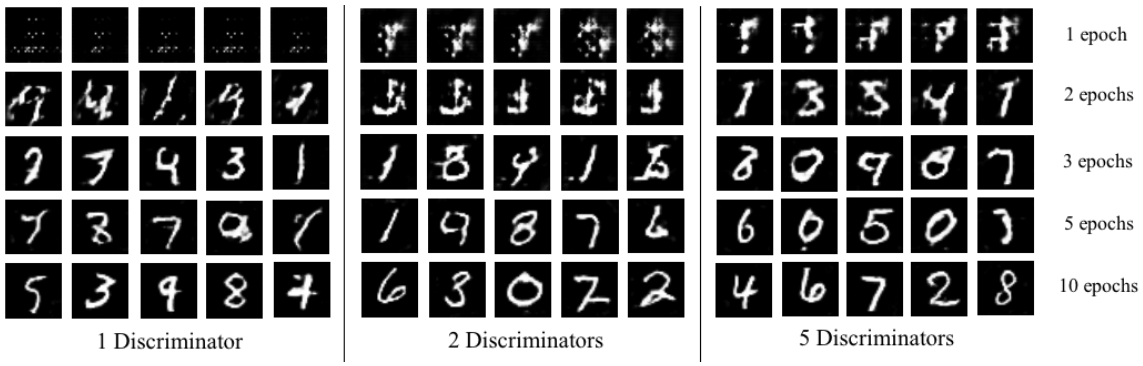


Figure 4.4: Comparison of image quality across epochs for $N = \{1, 2, 5\}$ using GMAN-0 on MNIST.

	Score	Variant	GMAN*	GMAN-0	GMAN-max	mod-GAN
Better→	**0.127**	GMAN*	-	-0.020 ± 0.009	-0.028 ± 0.019	-0.089 ± 0.036
	0.007	GMAN-0	0.020 ± 0.009	-	-0.013 ± 0.015	-0.018 ± 0.027
	-0.034	GMAN-max	0.028 ± 0.019	0.013 ± 0.015	-	-0.011 ± 0.024
	-0.122	mod-GAN	0.089 ± 0.036	0.018 ± 0.027	0.011 ± 0.024	-

Table 4.1: Pairwise GMAM metric means with *stdev* for select models on MNIST. For each column, a positive GMAM indicates better performance relative to the row opponent; negative implies worse. Scores are obtained by summing each variant's column.

Figure 4.8 displays images generated by GMAN-0 on CIFAR-10. See Appendix C.0.3 for more results.

We also found that GMAN is robust to *mode collapse*. We believe this is because the generator must appease a diverse set of discriminators in each minibatch. Emitting a single sample will score well for one discriminator at the expense of the rest of the discriminators. Current solutions (e.g., minibatch discrimination) are quadratic in batch size. GMAN, however, is linear in batch size.

4.6 Conclusion and Future Work

We introduced multiple discriminators into the GAN framework and explored discriminator roles ranging from a formidable adversary to a forgiving teacher. Allowing the generator to automatically tune its learning schedule (GMAN*) outperformed

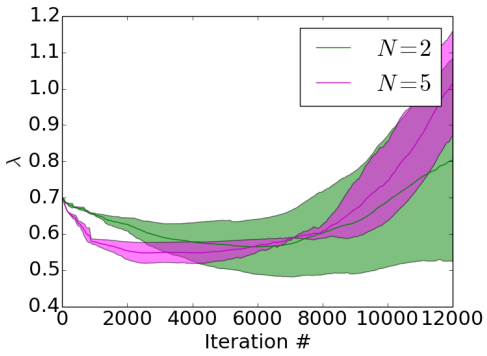


Figure 4.5: GMAN* regulates difficulty of the game by adjusting λ. Initially, G reduces λ to ease learning and then gradually increases λ for a more challenging learning environment.

	Score	λ $(N=5)$	λ^*	$\lambda=1$	$\lambda=0$
Better→	**0.028**	λ^*	-	$\frac{-0.008}{\pm 0.009}$	$\frac{-0.019}{\pm 0.010}$
	0.001	$\lambda=1$	$\frac{0.008}{\pm 0.009}$	-	$\frac{-0.008}{\pm 0.010}$
	-0.025	$\lambda=0$	$\frac{0.019}{\pm 0.010}$	$\frac{0.008}{\pm 0.010}$	-

Figure 4.6: Pairwise $\frac{\text{GMAM}}{stdev(\text{GMAM})}$ for GMAN-λ and GMAN* (λ^*) over 5 runs on MNIST.

GANs with a single discriminator on MNIST. In general, GMAN variants achieved faster convergence to a higher quality steady state on a variety of tasks as measured by a GAM-type metric (GMAM). In addition, GMAN makes using the original GAN objective possible by increasing the odds of the generator receiving constructive feedback. Follow up research motivated by the curriculum training schedule just presented achieved some of the highest quality images generated by a GAN to date [57].

4.6.1 Up Next

One of the benefits observed of GMAN is accelerated convergence of the minimax loss to steady-state. Often, convergence of the loss along with visually satisfactory samples indicates to practitioners that training is complete. However, do a steady loss and steady sample quality imply that the weights of the game have converged? The next chapter borrows techniques from dynamical system theory to answer this question.

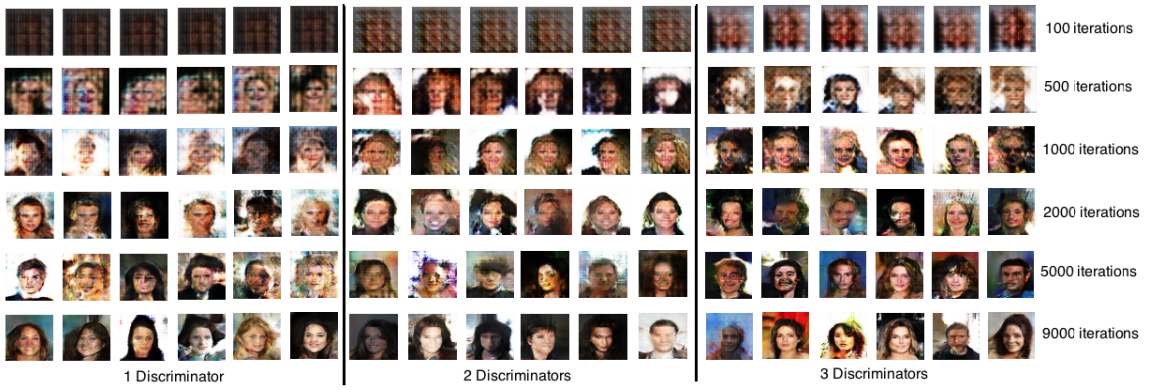


Figure 4.7: Image quality improvement across number of generators at same number of iterations for GMAN-0 on CelebA.

Figure 4.8: Images generated by GMAN-0 on the CIFAR-10 dataset.

CHAPTER 5

ANALYZING NON-MONOTONE GAMES

5.1 Purpose of Research

Many of the equilibrium problems of interest in ML, for example GANs, are not monotone. That being said, researchers have been able to employ heuristics to achieve promising empirical results which demonstrates that training these models is tractable to some degree. Similar challenges were encountered in deep learning. It was thought that deep networks would be intractable to train due to their inherent non-convexity, yet the repeated successes of researchers suggested otherwise. It was only recently discovered through random matrix (Hessian) theory and other approaches that the primary obstacles to successful optimization of deep networks are saddle points and also that most local minima are only marginally suboptimal. This finding has been followed by a surge of research into methods for "escaping saddle points".

Given that monotone operator theory can not explain the success of GANs trained with deep networks, we would like some set of tools for analyzing these more complex models. In contrast to optimization theory where one can analyze a local neighborhood by examining the spectrum of the Hessian, there exist structures in dynamical systems that can only be recognized at a macro-scale. For example, we cannot necessarily recognize a limit-cycle of large radius by examining the cycle's center. Here, we extend tools from dynamical systems theory, namely Lyapunov Exponent calculation, for characterizing the dynamics of complex systems. These tools reveal qualitative characteristics of the equilibrium dynamics including stable fixed points, limit cycles, and strange attractors. By gaining a better understanding of complex equilibrium

problems we may develop better algorithms and better understand behavior away from the equilibrium.

Spoiler: By computing the Lypapunov exponents, we are able to show that successfully trained GANs are not always converging to equilibria or even local neighborhoods of equilibria. Instead they are sometimes converging to limit-cycles or strange attractors. Our contribution focuses on identifying these challenges, and we leave overcoming these challenges to future work.

5.2 Introduction

While the necessary progress in (non)-monotone operator theory / VIs may not emerge for some time, Lyapunov exponent and machine learning techniques can provide useful empirical tools for analyzing game dynamics. In Section 5.3 we explain how VI's connection to projected dynamical systems allows us to apply a Monte-Carlo sampling tool for analyzing complex VI problems; we then enhance this tool in Section 5.4 so it scales to large games (i.e., many player variables). In Section 5.5 we discuss an interesting application in modeling the cloud services market economy. We then we explore our proposed model with a hypothetical case study and demonstrate the proposed machine learning pipeline on our new cloud services model. In Section 5.7, we compute the Lyapunov exponents of GANs applied to a variety of datasets and show that "successful" GAN training sometimes converges to strange attractors.

5.3 Identifying Boundaries of Attraction

VI theory provides no general guarantees on the uniqueness of Nash equilibria when losses are non-convex. This motivates an *algorithmic* approach to identifying the number of equilibria, their locations, and possibly other phenomena. In particular, we

will leverage theory and algorithms from dynamical systems - we refer the interested reader to the book by Strogatz [2014] for a gentle introduction.

Nagurney and Zhang [1996] established an equivalence between VIs and projected dynamical systems that makes available new theory and algorithms, providing a foundation for the necessary analysis.

Definition 1. *Assuming that the feasible set $\mathcal{X}$ is a convex polytope, the projected dynamical system, PDS($F, \mathcal{X}$), corresponding to VI($F, \mathcal{X}$) is $\dot{x} = \Pi_{\mathcal{X}}(x, -F(x))$ with $x(0) = x_0$ and $\Pi_{\mathcal{X}}(x, -F(x)) = \lim_{\delta \to 0} \frac{P_{\mathcal{X}}(x - \delta F(x)) - x}{\delta}$.*

In terms of attractors, strongly monotone VIs admit only stable fixed points accompanied by a relatively small range of attractor dynamics including stable spirals and nodes. As expected, less can be said of VIs arising from non-convex loss functions. Other, qualitatively distinct attractors include limit cycles, tori, and strange attractors (see Figure 5.1). It's important to be aware of these other possible attrac-

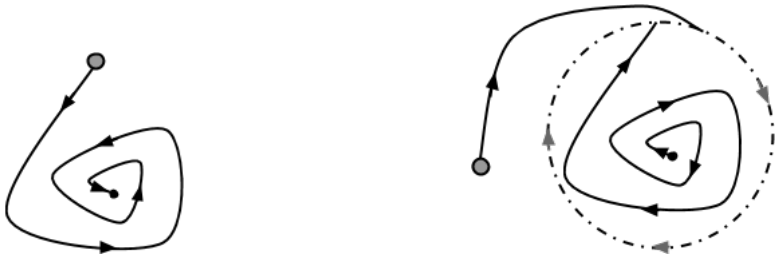


Figure 5.1: Stable spiral (left) and limit cycle (right, dashed).

tors when analyzing a more complex system. For example, consider a stock exchange and assume the market closed with prices at a stable equilibrium. A stock opening the next morning in one range of prices may cause the group of stocks as a whole to simply *readjust* to a new stable NE. On the other hand, opening the stock in another range of prices may result in the group tending towards a limit cycle where prices continuously oscillate. It's then obvious that the ability to predict which ranges result in which behaviors helps determine where it's best to open the stock. Thus, we

would like to identify the endpoints of these ranges, or more generally, the *boundaries of attraction* (BoAs).

There are several existing techniques for identifying BoAs. The theory of Lyapunov functions has long motivated a large group of these, however, they can only be applied to restricted types of nonlinear systems and are not capable of identifying the entire BoA [66]. Others attempt to approximate Lypapunov functions using a set of scalar functions [88]. Still other, non-Lypunov based approaches have been proposed that work backwards from the attractor. These methods tend to be lightweight, but less reliable. Recently, Armiyoon and Wu [2014] developed a method for identifying BoAs that relies on Lyapunov exponent (LE) theory. Convergence of LEs can be slow, but they enjoy the advantage of being independent of initial conditions and can be applied to general nonlinear systems. The authors proposed the use of Monte-Carlo sampling to alleviate the computational load of calculating LEs. Their approach can give us an idea of the number and types of attractors we can encounter in a bounded space, but first, to understand their algorithm, we need an understanding of LEs.

LEs measure the long-term deformation of a sphere along a trajectory in the dynamical system and are invariant within a single BoA. It's this invariance property that allows us to use the LE as a signature for the basin of attraction in spite of varying initial conditions.[1] Furthermore, LEs reveal the type of attractor. For instance, if all values in the LE are negative, the attractor is a stable fixed point; if instead, one of the values is zero, the attractor is a limit cycle (see Table 5.1).

Consider the following linear approximation to an n-dimensional dynamical system, $\dot{x} = F(x)$: $\dot{\psi} = J\psi$ where we have replaced x with a matrix ψ whose columns are meant to approximate the eigenvectors of the system. Assume the eigenvalues of the Jacobian are distinct (implies its eigenvectors are linearly independent). In the

[1] Two basins may have the same LE though.

Type	Sorted LE Spectrum
Stable Fixed Point	$(-, \ldots, -)$
Limit Cycle (1-torus)	$(0, -, \ldots, -)$
n-Torus	$(\underbrace{0, \ldots, 0}_{n \text{ leading } 0's}, -, \ldots, -)$
Chaos (*repeller*)	$(+, \ldots)$

Table 5.1: LE spectrum for continuous-time attractors.

following example, we will focus on the first column of ψ and assume $\psi_1(0) = u_1$, the first unit-norm eigenvector. The following process provides intuition for the LE computation process:

$$\dot{\psi}_1 = J\psi \tag{5.1}$$

$$\psi_1(t) = c_1 e^{\lambda_1 t} u_1 + \ldots + c_n e^{\lambda_n t} u_n \tag{5.2}$$

$$\psi_1(0) = u_1 = c_1 u_1 + \ldots + c_n u_n \implies c_1 = 1, c_2 = \ldots = c_n = 0 \tag{5.3}$$

$$\psi_1(\Delta t) = u_1 e^{\lambda_1 \Delta t} \tag{5.4}$$

$$\log ||\psi_1(\Delta t)|| = \log ||u_1 e^{\lambda_1 \Delta t}|| = \log(e^{\lambda_1 \Delta t}) + \log ||u_1|| \tag{5.5}$$

$$= \lambda_1 \Delta t. \tag{5.6}$$

Therefore, by evolving the system $\dot{\psi} = J\psi$ and tracking the change in norm of the columns of ψ, we can attempt to recover the "eigenvalues" of the system.

The general idea of Armiyoon and Wu's algorithm is to sample grid points with high probability of being near a BoA, compute the LEs of the sampled grid point as well as a few of its neighbors, and then compare LEs between all pairs of tested points. If a pair of LEs do not match, then they are located on either side of a BoA and the pair can be added to a training set for a classifier (e.g. SVM). In addition, the probabilities of the neighbors can be increased since they are most likely near the boundary as well. In the case where the LEs are the same (within some tolerance), the probabilities can be reduced.

In their paper, they consider domains in $\mathbb{R}^2$ to $\mathbb{R}^4$. Low dimensionality allows them to apply standard LE calculation techniques coupled with more basic ODE solvers (e.g. constant step size) without compromising runtime. We are more interested in the high dimensional domains that often occur in VIs with many players, each of which controls multiple variables. Given a constant number of grid points per dimension, the total number of grid points scales exponentially with the number of dimensions and quickly makes this Monte-Carlo sampling approach impractical. Moreover, basic ODE solvers may incorrectly track the trajectories of systems that contain multiple time scales.

5.4 Improving the BoA Identification Algorithm

As stated, we would like to alter the BoA algorithm so it scales more gracefully with dimensionality. The first step is to adjust the LE computation to be able to accompany an ODE solver ($\mathbb{S}$) with an adaptive step size scheme ($\mathbb{T}$). While the fix is somewhat trivial, it was very difficult to come across explicit LE computation instructions for constant step sizes [117, 99] and we never found any such instructions for adaptive step sizes. We include the necessary pseudocode in Algorithm 6.

Algorithm 6 LE for use with Adaptive Step Sizes

INPUT: $F, x^0, \Delta t^0, \mathbb{S}, \mathbb{T}$

$\Lambda = (0, \ldots, 0), \psi^0 = \mathbb{I}, k = 0, T = 0$
J $\leftarrow$ Jacobian(F$(x)) \cdot \psi$
GS $\leftarrow$ GramSchmidt **without** normalization
$|\cdot|_c \leftarrow$ column-wise norm
repeat
  $x^{k+1} = \mathbb{S}(x^k, \Delta t^k, F)$ *evolve trajectory
  $\hat{\psi}^{k+1} = \mathbb{S}(\psi^k, \Delta t^k, J)$ *evolve ellipsoid
  $\hat{\psi}^{k+1} = \text{GS}(\hat{\psi}^{k+1})$ *orthogonalize ellipsoid
  $\lambda \Delta t = \log(|\hat{\psi}^{k+1}|_c)$ *measure growth
  $\Lambda = (\Lambda \cdot T + \lambda \Delta t)/(T + \Delta t^k)$ *update mean
  $T = T + \Delta t^k$
  $\psi^{k+1} = \hat{\psi}^{k+1}/|\hat{\psi}^{k+1}|_c$ *reset to sphere
  $\Delta t^{k+1} = \mathbb{T}(x^k, x^{k+1}, \psi^k, \psi^{k+1}, \Delta t^k)$
until Convergence of Λ

Next, we point out that computing an LE involves following the trajectory from an initial point x^0 until convergence. The runtime for this computation alone can be extensive for high dimensional systems. Since the LE is a global property and hence, in theory, a property shared by all points along the trajectory, ignoring the computed LE's association with all points along the trajectory seems particularly wasteful. Instead of throwing out this information, we can include it by recognizing that all subsequent points after the initial point along the trajectory are ideally progressing away from the boundary (assuming integer dimensional BoA's). Moreover, the LE gives us an idea of the exponential rate of *divergence* away from the boundary, and so we can use the LE to decay the probability of grid points along the trajectory. Algorithm 7 describes the steps used to adjust probabilities using this heuristic and an example is displayed in Figure 5.2. This approach allows us to update the probabilities of many more grid points per LE computation, helping to combat the issues of dimensionality.

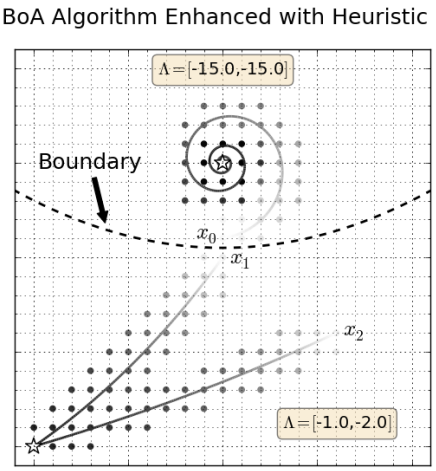


Figure 5.2: The probabilities of points farther along the trajectory (white to black) should be reduced as they are most likely far away from any boundary. These adjustments can be shared with the surrounding grid points.

5.5 A New Market Model

We demonstrate the potential of the proposed algorithm on a model of the prominent, commercial cloud market that has arisen over the past decade. Several companies, or clouds, offer compute services to the public at different prices and qualities of service. In general, the quality of a service degrades as the price is lowered. Each cloud i advertises the same price-degradation pair, (p_i, d_i), to every client j. As suggested by Wang et al. [2015], client j's demand for cloud i, Q_{ij}, is monotonically decreasing in p_i and d_i with a nonzero *zero-utility* cutoff. Note that while we will continue to discuss this model in the context of cloud services, our model can likely be applied to any industry where firms set prices for quality of service at a cost to themselves.

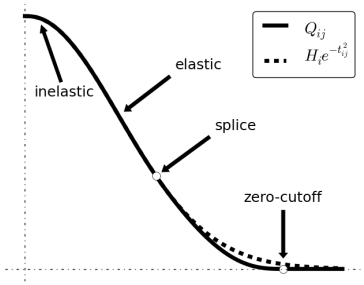
Our demand function, Q_{ij}, consists of a squared-exponential spliced with a 5th degree polynomial (coefficients β are in Appendix D.2). The function is twice differentiable, contains both elastic and inelastic regions, and drops to zero-demand at finite t_{ij} (see Figure 5.3). We've also included factors $p_r = \frac{p_i}{\bar{p}}, d_r = \frac{d_i}{\bar{d}}$ where $\bar{p}$ and $\bar{d}$ are cloud price and degradation averages so that clients are also attracted to low

Algorithm 7 Update Grid Probability Along Trajectory $\boldsymbol{x}$

INPUT: LE, $\boldsymbol{x}, \Delta t^k, d_{max}$

Initialize hashes N, D
$t = 0, T = \sum \Delta t^k, \lambda = \max(|\text{LE}|)$
for x^k in $\boldsymbol{x}$ **do**
  $\boldsymbol{g}, \boldsymbol{d} = \text{gridNeighborsDistances}(x^k)$
  for each (g, d) in $(\boldsymbol{g}, \boldsymbol{d})$ **do**
    $N[g] \overset{+}{=} e^{-\lambda \cdot t/T} \cdot \Delta t^k$
    $D[g] \overset{+}{=} \Delta t^k$
  end for
  $t \overset{+}{=} \Delta t^k$
end for
for each g in N, D **do**
  $P(g) \overset{*}{=} N[g]/D[g]$
end for

prices/degradation in a relative sense. Client-cloud loyalty is simulated through client j's elasticity coefficient, α_{ij}, while purchasing power is given by H_{ij} (see equations 5.7 and 5.8).



$$t_{ij} = \alpha_{ij} p_i d_i p_r d_r \tag{5.7}$$

$$Q_{ij} = \begin{cases} H_{ij} e^{-t_{ij}^2} & , t_{ij} \in [0, t^c] \\ \sum_{k=0}^{5} \beta_k t_{ij}^i & , t_{ij} \in (t^c, t^c + 1) \\ 0 & , t_{ij} \in [t^c + 1, \infty) \end{cases} \tag{5.8}$$

$$\pi_i = \sum_j \underbrace{p_i Q_{ij}(p_i, d_i)}_{revenue} - \underbrace{\frac{c_i}{d_i^2} Q_{ij}(p_i, d_i)}_{cost} \tag{5.9}$$

Figure 5.3: Proposed demand function $Q_{ij}(t_{ij})$ with $t^c = 1$.

Cloud profit[2], π_i, is defined as revenue minus cost where cost scales as the square of quality $(1/d_i)$ with coefficient c_i.

Let $x_i = (p_i, d_i) \in [\epsilon, \infty)^2$, $i \in 1, \ldots, n$, and $L_i(x_i, x_{-i}) = -\pi_i$, then we would like to analyze the model given by VI$(F, \mathcal{X})$ where $F = (\nabla_{x_1} L_1, \ldots, \nabla_{x_n} L_n)$ and $\mathcal{X} = [\epsilon, \infty)^{2n}$. Note that $\mathcal{X}$ is unbounded (not compact), so we are not guaranteed a solution to the VI exists.

We stated in the introduction, an equivalence between the VI with pseudo-convex losses and the NE problem. The cloud profit functions, as defined, are, in general, non-concave. Although we no longer have a guarantee that solutions to the VI are necessarily Nash equilibria, we still have an equivalence between VI$(F, \mathcal{X})$ and PDS$(F, \mathcal{X})$. This means we can perform the same BoA analysis, but we'll need to check stable fixed points to see if they satisfy the Nash definition, which amounts to solving n non-convex, 2-D, constrained optimization problems. In our solution, we use Scikit-learn's *L-BFGS-B* for this task [90]; runtime is negligible relative to the BoA algorithm.

5.6 Cloud Services Experiment

To demonstrate the promise of the described pipeline, we focus on identifying the BoA's (as well as Nash equilibria) of our proposed cloud services market economy model. Here we investigate a hypothetical scenario in which four cloud companies compete for the opportunity to provide service to five clients looking to transfer their in-house computation to the cloud. The first three cloud companies are large providers with highly optimized servicing capabilities (lower c_i), while the last two are newcomers to the market, trying to fill a niche with higher cost green-tech (higher c_i). Client 1 is a big buyer loyal to clouds with the 3 lowest cost functions (e.g. big

[2] π_i is nondifferentiable at $d_i = 0$, however, zero price and infinite quality are nonsensical, so our market is constrained to $[\epsilon, \infty)$

name providers). Client 2 is a medium buyer with slight preference towards green-tech. Client 3 is a small buyer who prefers green-tech, but is not opposed to a large corporation. Client 4 is a big buyer loyal to cloud 1, but otherwise prefers green-tech. To compute LEs, we're using a projected version of Heun-Euler, a 2nd order, explicit ODE solver with an adaptive step size.

Running the BoA algorithm[3] over a 10 dimensional grid (6 points/dimension) with the enhancements described in section 5.4 returns a set of of positive-negative samples for each reference LE. After running an SVM on each LE sample set, we define the boundaries as the critical points at which the SVM with the highest margin prediction is dethroned by an SVM with a higher margin prediction.

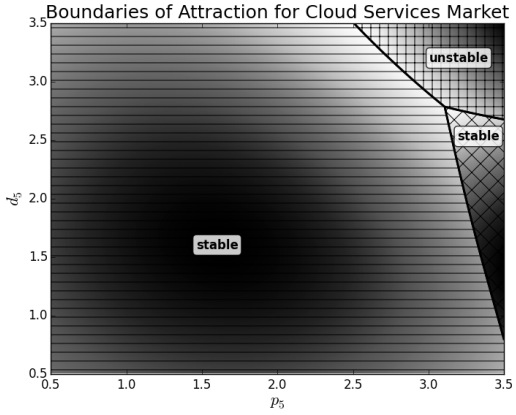


Figure 5.4: Basins of attraction are marked stable or unstable and differentiated by pattern, each with a gradient that runs from most likely belonging to the region (dark) to least likely (light). Boundaries are marked by black lines.

In Figure 5.4, we consider a scenario where green-tech newcomer, cloud 5, enters the pre-established cloud services market described above. Opening with (p_5, d_5) in either of the stable regions sets the market on a path toward the same NE; the two regions are mislabeled as distinct due to noise in their LE calculations. On the other hand, launching their business in the unstable region results in chaos and

[3]All code at https://github.com/all-umass/VI-Solver

should be avoided. Although we can't visualize both green-tech newcomers entering the market (>3-D), we can quickly evaluate our SVM classifiers to determine the corresponding basin of attraction and associated characteristic LE for any given set of price-degradation pairs. Obviously, there are factors that our model does not take into account. In spite of this, knowledge of BoAs combined with market monitoring can also be used to suggest when a discussion of external intervention might be prudent (e.g. government regulation) or when external intervention might transition the market into a more desirable basin of attraction.

5.7 Lyapunov GANs

Several papers have conducted a local stability analysis of common GAN training algorithms about the global equilibrium. In the remainder of this chapter, we explore whether or not these analyses are relevant to current GAN training protocols. More concretely, does successful GAN training imply convergence to a locally stable fixed point or are weights possibly converging to other dynamics such as limit cycles and strange attractors?

GANs produce the sharpest and most perceptually pleasing image samples to date. They are also useful for other domains. Improving their performance and being able to trust their training can make their widespread adoption into commercial applications a reality. Understanding the dynamics at the end of GAN training will provide useful information for developing better algorithms that converge to local equilibria. For example, if GANs are converging to the local equilibrium, then we can use algorithms and analysis that focuses on that. However, if GANs are converging to a limit cycle, we can use algorithms designed to break through the cycle and converge towards the center. And if GANs are converging to a strange attractor, we need to research ways of finding the fixed point of these systems. Also, are their other implications of converging to a strange attractor. Is that a desirable property? Here, we focus

on identifying the dynamics near the end of GAN training and leave overcoming the identified challenges to future research.

Adam [59] is an algorithm commonly used to successfully train GANs, however, the Adam update scheme is iteration dependent. This is to say that the dynamics for Adam cannot be written down simply as an autonomous ODE, $\dot{x} = F(x)$. To compute the Lyapunov exponents for the GAN, we require this property. Therefore, we shift focus to RMSProp [111], another popular algorithm used to successfully train GANs. RMSProp can be written down as an autonomous ODE:

$$a_t = \gamma a_{t-1} + (1-\gamma) g_t^2 \tag{5.10}$$

$$x_t = x_{t-1} - \frac{\eta g_t}{\sqrt{a_t + \epsilon}} \tag{5.11}$$

where $g_t = \nabla_x f(x_t)$. We can rewrite the RMSProp update as follows:

$$a_t = a_{t-1} - (1-\gamma)(a_{t-1} - g_t^2) \tag{5.12}$$

$$x_t = x_{t-1} - \frac{\eta g_t}{\sqrt{\gamma a_{t-1} + (1-\gamma) g_t^2 + \epsilon}}. \tag{5.13}$$

After rewriting in this form, its ODE formulation is apparent:

$$\dot{a} = -(1-\gamma)(a - g^2) \tag{5.14}$$

$$\dot{x} = -\frac{\eta g}{\sqrt{\gamma a + (1-\gamma) g^2 + \epsilon}}. \tag{5.15}$$

It was shown that RMSProp as well as other related algorithms like Adam are not always locally convergent [94]. For this reason, after training the GANs with RMSProp, we switch to SGD in order to determine local convergence near the end of training.

If the dimensionality of the dynamical system is very large ($n \gg 10$), then it is more efficient to compute only the top-k LEs. In this case ψ^0 is constructed as the

first k columns of $\mathbb{I}$. Recall that the top LEs reveal the qualitative dynamics of the system. Moreover, representing the Jacobian of a large system in memory can be prohibitively expensive. Line 2 of Algorithm 6 only requires the action of $J(F(x))$ on the ellipsoid ψ whose result is an $n \times k$ matrix; it does not require the $n \times n$ matrix $J(F(x))$ on its own. This action can be approximated with finite differences:

$$[J(F(x)) \cdot \psi]_i \approx \frac{F(x + \epsilon\psi_i) - F(x)}{\epsilon} \tag{5.16}$$

where $\epsilon \ll 1$ and the subscript i denotes the i^{th} column of the matrix.

5.8 GAN Experiments

We compute LEs for GANs in several domains:

- Constant-Linear (CL-GAN) and Linear-Quadratic GAN (LQ-GAN): We examine simultaneous gradient descent and the consensus algorithm applied to learning the mean and variance of a 1-d distribution. We use this setting to illustrate how Lyapunov exponents recover known properties of these systems.

- Mixture of 8 (MO8G) and 25 Gaussians (MO25G): Fitting mixtures of Gaussians is a common benchmark for GAN models. We show that successfully trained neural-network based GANs can exhibit positive LEs in this setting. By examining the change in the norm and angle of the weights throughout training with RMSProp, we establish that the weights naturally stay within a compact set. This fact combined with positive LEs suggest the weights are caught in a strange attractor.

- MNIST and CIFAR-10: We discover similar results when performing the same calculations for these popular image dataset benchmarks. GAN training with RMSProp gets caught in a strange attractor.

Note that assuming simultaneous gradient descent, the Wasserstein GAN was shown to be cyclic locally while the original objective is locally stable [80]. However, the consensus algorithm [74] is proven to converge to a local equilibrium if the step size is small enough and the iterates are near enough to the equilibrium, $||x_k - x^*|| < \epsilon$. In the following experiments, we train GANs using neural network distance [8] (similar to Wasserstein distance) and have found the consensus algorithm to perform quite well in practice.

Standard GAN training uses stochastic optimization methods which estimate expectations using minibatches of samples. To remove the possibility of stochasticity introducing chaotic behavior and conflating our understanding of training dynamics, we sometimes use one large, single minibatch to compute expectations over both $p(z)$ and $p(x)$ throughout the entire training process. We will distinguish between this setting and the traditional stochastic setting by writing [Det] or [Sto] at the beginning of the Figure caption.

5.8.1 CL and LQ-GAN

The analytically computed Lyapunov exponents for simultaneous gradient descent applied to the constant-linear GAN (CL-GAN) are $\Lambda_{1,2} = \log(\sqrt{1+\alpha^2})/\alpha = 0.005$ where $\alpha = 0.01$ is the step size.

Proof. For the CL-GAN, $x_{k+1} = x_k - \alpha A x_k = (I - \alpha A)x_k$ where $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = -A^\top$. Let $||x_k|| = 1$.

$$||x_{k+1}||^2 = ||(I - \alpha A)x_k||^2 \tag{5.17}$$

$$= x_k^\top (I - \alpha A)(I - \alpha A)x_k \tag{5.18}$$

$$= x_k^\top (I + \alpha^2 A^\top A)x_k \tag{5.19}$$

$$= (1 + \alpha^2)||x_k||^2 = 1 + \alpha^2 \tag{5.20}$$

$$\implies \Lambda_{1,2} = \frac{\log ||x_{k+1}||}{\alpha} = \frac{\log \sqrt{1 + \alpha^2}}{\alpha}. \tag{5.21}$$

□

The consensus algorithm applied to the same problem gives $\Lambda_{1,2} = -1.005$.

Proof. For the CL-GAN, $x_{k+1} = x_k - \alpha \frac{A^\top - A}{2} A x_k = (1 - \alpha)x_k$.

$$||x_{k+1}||^2 = ||(1 - \alpha)x_k||^2 \tag{5.22}$$

$$\implies \Lambda_{1,2} = \frac{\log ||x_{k+1}||}{\alpha} = \frac{\log(1 - \alpha)}{\alpha} = -1.005. \tag{5.23}$$

□

Note these agree with the values empirically computed using finite differences (see values reported in the titles of Figure 5.5).

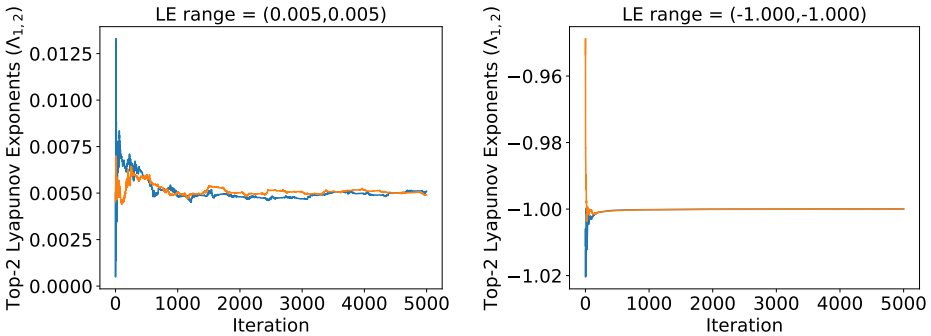


Figure 5.5: [Det] Top two Lyapunov exponents vs iterations for CL-GAN trained with simultaneous gradient descent (left) and the consensus algorithm (right).

We plot the exponents computed using stochastic optimization in Figure 5.6. Note the exponent calculation remains accurate although convergence to the analytical values is mildly delayed.

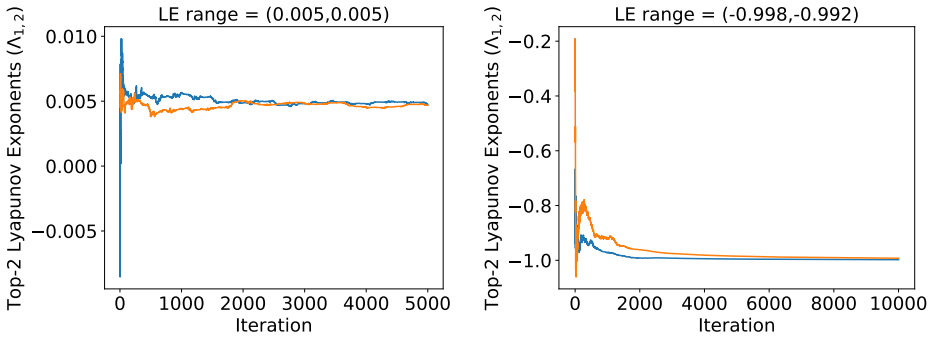


Figure 5.6: [Sto] Top two Lyapunov exponents vs iterations for CL-GAN trained with simultaneous gradient descent (left) and the consensus algorithm (right).

The consensus algorithm applied to the LQ-GAN results in $\Lambda_{1,2} \approx 0.03, -0.15$ which supports the earlier analysis (see Sections 3.7.1 and 3.8) that the consensus algorithm is not convergent on this domain.

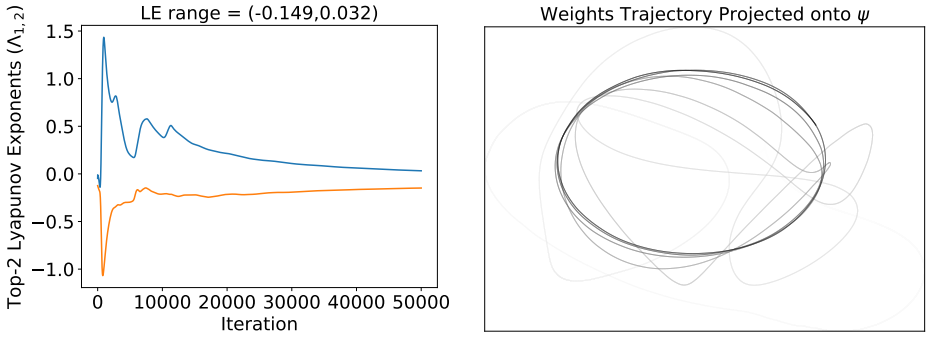


Figure 5.7: [Det] Top two Lyapunov exponents vs iterations for LQ-GAN trained with the consensus algorithm (left) and weights projected onto the first two columns of ψ (right). The trajectory of $\Lambda_{1,2}$ over iterations reveals that the system is initially chaotic (positive leading exponent) and then converges toward a limit cycle (near zero leading exponent). The trajectory of the weights projected onto ψ supports this conclusion: initial portions of the trajectory (light gray) exhibit chaos while later portions (black) reveal cyclic behavior.

5.8.2 Mixture of Gaussians

The Lyapunov exponents for MO8G with RMSProp+consensus are 942 and 895 (see Figure 5.8). For MO25G, they are 7296 and 7089 (see Figure 5.9). Losses for both systems have converged to steady-state and sample distributions for both systems accurately reflect ground truth, yet the LEs computed after switching to SGD are near zero for MO8G indicating a limit cycle and positive for MO25G indicating chaos.

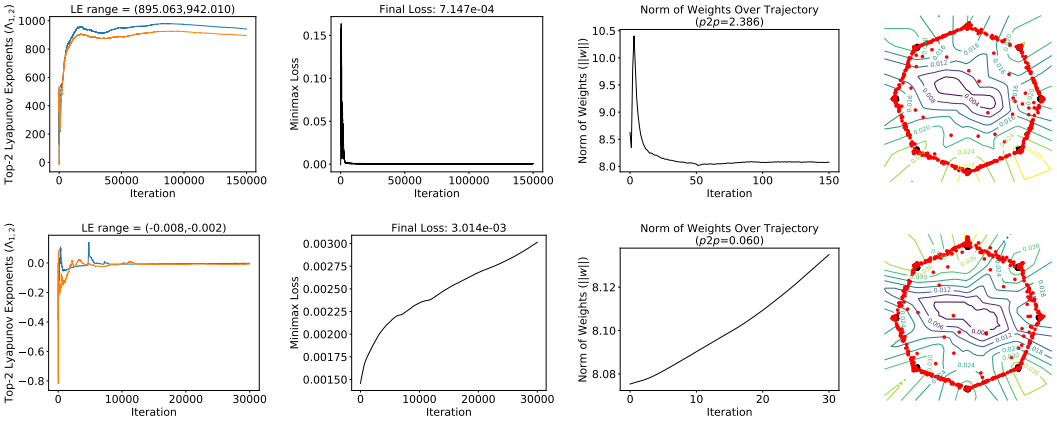


Figure 5.8: [Det] Top two Lyapunov exponents (left), minimax loss (2nd column), Euclidean norm of the weights (3rd column), and final samples (right) vs iterations for a GAN trained with RMSProp+consensus on a mixture of 8 Gaussians (top row). Training is continued without RMSProp in the bottom row. We also tried rescaling the gradients by the final exponentially averaged norms obtained by RMSProp, but have not presented them here because this approach immediately diverged (NaNs).

5.8.3 MNIST

The Lyapunov exponents for RMSProp+consensus (stochastic) are 5041 and 3789 (see Figure 5.10). Notice in Figure 5.10 (see insets) that we observe highest sample quality when the loss is stable. This coincides with a steady norm for the weights. However, the LEs over this period are increasing, which suggests the system is becoming more chaotic. The constant norm suggests the weights are remaining within some compact ball, yet the positive exponents suggest the system is divergent. These

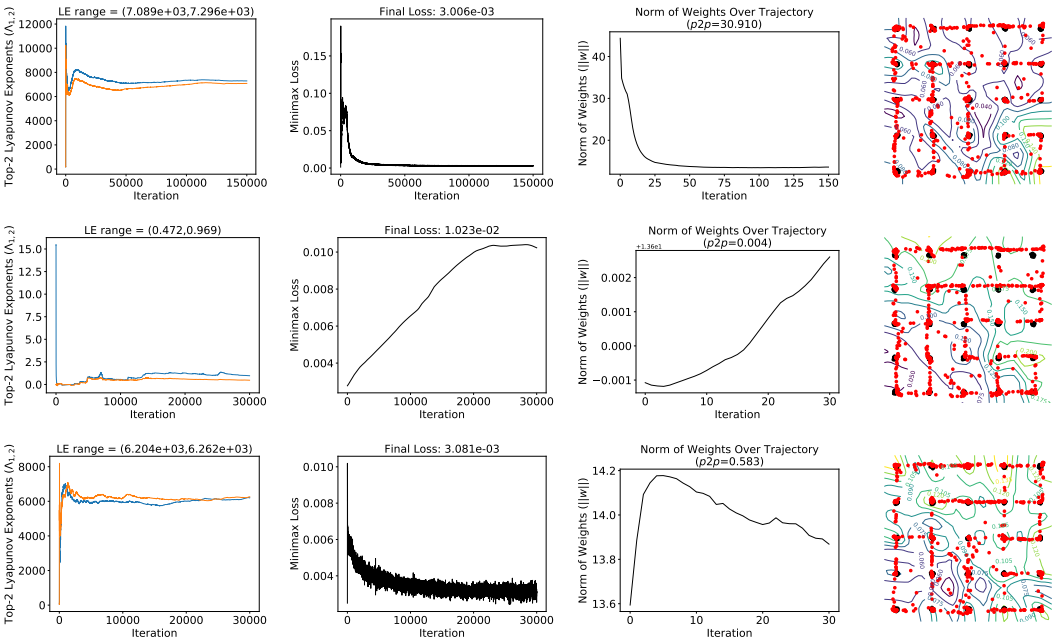


Figure 5.9: [Det] Top two Lyapunov exponents (left), minimax loss (2nd column), Euclidean norm of the weights (3rd column), and final samples (right) vs iterations for a GAN trained with RMSProp+consensus on a mixture of 25 Gaussians (top row). Training is continued without RMSProp in the middle row. We also tried rescaling the gradients by the final exponentially averaged norms obtained by RMSProp (bottom row).

two together suggest a strange attractor that is becoming increasingly chaotic. The system finally "breaks" around 300 thousand iterations at which point the norm of the weights increases until the loss stabilizes again and sample quality returns to its previously high level.

5.8.4 CIFAR-10

The Lyapunov exponents for RMSProp+consensus (stochastic) are 19884 and 15931 (see Figure 5.11). The loss in Figure 5.11 remains relatively stable and the norms of the weights appear to be approaching an asymptote. We would need to train for many more iterations to confirm that the system is caught in a strange attractor, but based on results from the domains examined above, it is likely.

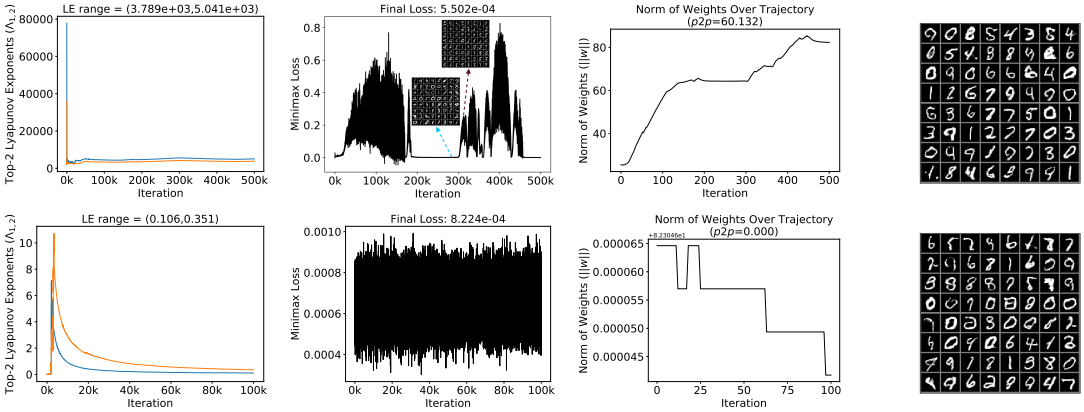


Figure 5.10: [Sto] Top two Lyapunov exponents (left), minimax loss (2nd column), Euclidean norm of the weights (3rd column), and final samples (right) vs iterations for a GAN trained on MNIST with RMSProp+consensus (top) and then just consensus (bottom).

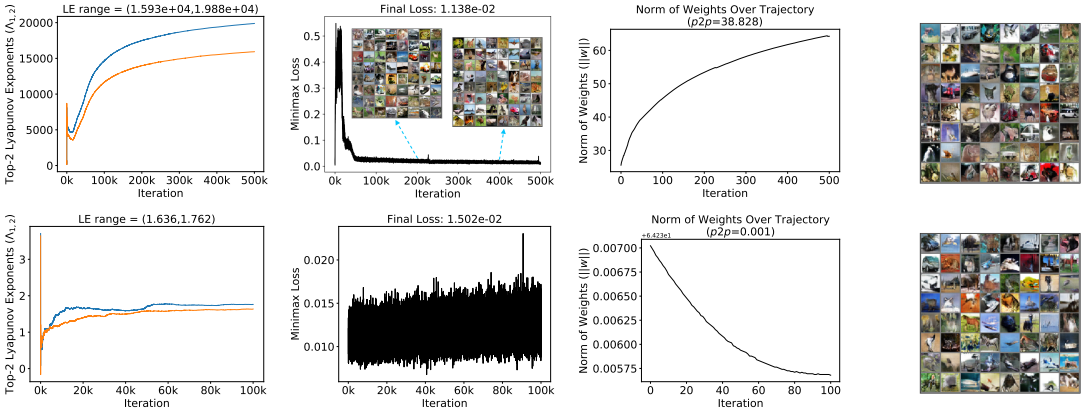


Figure 5.11: [Sto] Top two Lyapunov exponents (left), minimax loss (2nd column), Euclidean norm of the weights (3rd column), and final samples (right) vs iterations for a GAN trained on CIFAR-10 with RMSProp+consensus (top) and then just consensus (bottom).

Note that the GAN trained on CIFAR-10 consistently generated high quality samples (see inset of loss in Figure 5.11) while the one trained on MNIST exhibited intermittent periods of divergence that prevented successful training. Below in Figure 5.12, we plot the PCA-projected trajectories for both domains—the trajectory for the GAN successfully trained on CIFAR-10 matches the trajectories of successful trained models reported in [70].

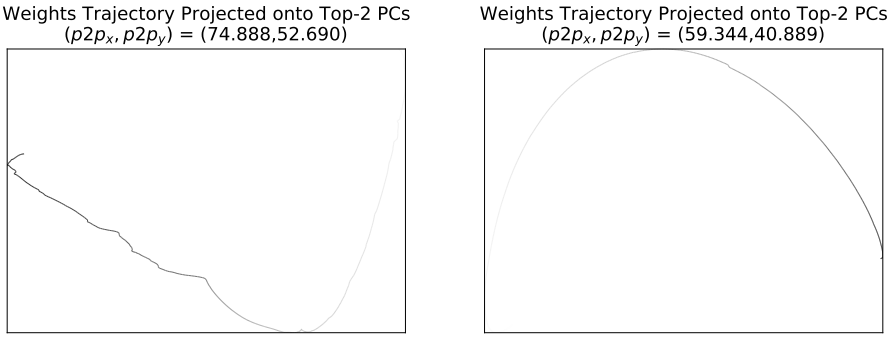


Figure 5.12: Projection of the generator and discriminator weights onto the top two principal components vs iterations for a GAN trained on MINST (left) and CIFAR-10 (right) using RMSProp.

5.9 Conclusion and Future Work

In this chapter, we presented an improved Lyapunov Exponent calculation and Boundary of Attraction Identification algorithm. We demonstrated this algorithm on an economic game model of the cloud services economy. We also computed the top-k Lyapunov exponents of GANs using finite differences to approximate the Jacobian-vector products that are required.

By computing the Lypapunov exponents, we were able to show that successfully trained GANs are not always converging to equilibria or even local neighborhoods of equilibria. Given that adaptive stochastic gradient methods like RMSProp and Adam are the training methods of choice for GANs, it appears that successful training often means trapping the GAN in a limit cycle or strange attractor. Figuring out how RMSProp remains "trapped" in these attractors will require more research.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this dissertation we have made several contributions.

We began by introducing a framework for solving monotone equilibrium problems in an online or streaming setting, namely Online Monotone Equilibration (OME). This framework was constructed using a notion of regret that is defined as the path integral over the vector field associated with the equilibrium problem. By leveraging the properties of monotonicity, we can ensure that an *Extragradient* type algorithm achieves vanishing average regret as the number of samples in the stream approaches infinity. The *no-regret* algorithm derived from this framework is novel in the sense that the first step of the Extragradient update uses a stepsize that is growing at a rate $\mathcal{O}(T^{1/4})$ with respect to the stepsize of the second step. Unlike some other algorithms in the literature, ours does not require storing and averaging the iterates or the maps. We presented applications of OME which included equilibrating models of market economies, providing guarantees for network packet protocols, learning agent policies from non-stationary behavioral policies, and training a GAN online. With regards to variational inequality problems and their applications (market economies, traffic networks, supply chains, etc.) specifically, OME supports monotonicity as an important property to a dynamic, healthy ecosystem in which the goal is to have all interested parties safely track the equilibrium. It also suggests foresight (as Extragradient uses gradients from the "future") is critical to reaching an equilibrium.

The path integral loss used to construct OME supports a more sophisticated algorithm suited for the offline setting. We called this new algorithm *Crossing-the-*

Curl and proved that it is guaranteed to solve a certain GAN variant: the Linear-Quadratic GAN (LQ-GAN). Solving the LQ-GAN is equivalent to fitting a normal distribution to data, and so it represents a fundamental problem in density estimation or generative modeling. In addition to proving that *Crossing-the-Curl* solves the LQ-GAN with finite sample convergence rate guarantees, we showed the negative result that at the time of this thesis, none of the current GAN training algorithms provably solve the LQ-GAN according to both Variational Inequality (VI) and Dynamical Systems (Hurwitz) convergence theory.

Our approach to solving the LQ-GAN required applying *Crossing-the-Curl* in stages, increasing the complexity of the discriminator and generator at each stage. This insight motivates a more tailored training regimen for GANs in which discriminators of varying complexity are pitted against the generator. We called this setting Generative Multi-Adversarial Networks (GMAN). In this setting, the generator can be given control over which discriminators to focus on, deemed GMAN*, and experiments revealed an intuitive pattern. The generator chooses to compete against a weak discriminator initially, but competes against successively more complex discriminators later in training. In general, we found that introducing a variety of discriminators in to the training regimen resulted in 1) reduced variance of the minimax objective, 2) improved quality of generated samples, and 3) accelerated convergence of the minimax objective to steady-state.

The third property of GMAN prompted a closer inspection of the dynamics at the end of training. Does convergence of the minimax objective imply convergence of the generator and discriminator weights? More generally, what dynamics do the weights exhibit during training? To answer these questions, we computed the top-2 Lyapunov exponents of the system throughout training. The exponents we computed revealed that convergence of the objective does **not** necessarily imply convergence of the weights. The exponents also confirmed that a popular GAN training algorithm,

RMSProp, is not convergent [94]. Despite this fact, RMSProp (as opposed to non-adaptive gradient methods) is critical to successful GAN training. In some cases, switching to the consensus algorithm, a better understood yet more primitive training algorithm, revealed that the weights were in fact near a local equilibrium. In other cases, the Lyapunov exponents remained near zero suggesting the weights were caught in a limit-cycle or strange attractor.

The work in this thesis aims at making fundamental steps toward better understanding equilibrium problems as they pertain to machine learning and learning theory more generally. We hope that this work will provide a useful foundation for artificial intelligence and machine learning researchers to extend and study relevant equilibrium problems.

6.1 Future Work

With regards to our theoretical contributions, we focused on controlled settings where we could make progress. We intend to build on this progress by relaxing our assumptions and tackling other types of equilibrium problems. We also intend to explore more tangential applications of the ideas presented here. For example,

1. Is the β-smoothness constraint crucial to obtaining regret bounds? Can we relax this constraint? What is a simple example of a non-smooth vector field? What about an example that is not the gradient of any function?

2. In the OME framework, we assumed the addition of a strongly-convex regularizer to the path integral loss to make the adversarial setting learnable. In an equilibrium problem, it may make more sense to bias learning with a strongly-monotone field. In future work, we can look into replacing $R(x)$ with the path integral over a strongly-monotone field.

3. The modified path integral loss used to formulate OME suggested including an extra term beyond what we explored for *Crossing-the-Curl*: $-\langle z_t, \hat{z}_t \rangle$. Does an analysis including this modification improve convergence for the LQ-GAN?

4. We studied the LQ-GAN with only one possible parameterization. Does a different parameterization, for example, one where $D(y) = w_2(y - w_1)^2$, lead to better training dynamics? If the dynamics exhibit a symmetric Jacobian, what is the derived convex divergence between distributions?

5. We were able to use the fundamental theorem of calculus for path integrals to construct a general loss function for equilibrium problems. Machine learning models often consist of loss functions paired with function approximators. Can we use the path integral to define new function approximators as well? What advantages does this parameterization allow?

6. In this thesis we encountered several obstacles to equilibration that do not appear in optimization. In some cases, for example when deriving an algorithm to solve the LQ-GAN, we were able to turn the equilibrium problem into an optimization problem. Should this always be the goal—to somehow remove the rotation and transform the problem into a simpler one? When is it possible[1]? Or is there a reason to desire the more general dynamics possible in equilibrium problems?

7. In Chapter 4, we introduced the GMAM metric for comparing the performance of different GAN models. Recently, Balduzzi et al. [2018] introduced a theoretically sound framework for evaluating agent-vs-agent play. We may be able to apply this framework to improve the evaluation of GANs.

[1] We might start with the Hairy Ball theorem [51].

8. Stochastic gradient descent was recently proven to converge to limit cycles [20] suggesting that equilibrium algorithms that converge in the presence of cycles, e.g., *Crossing-the-Curl*, may be helpful here. In future work, we will explore this possibility experimentally.

9. A complete convergence analysis of the solution to an equilibrium problem requires analyzing the selected algorithm paired with the problem. For example, a complete convergence analysis of a GAN requires analyzing the selected algorithm, e.g., *Crossing-the-Curl*, paired with the chosen divergence, e.g., Jensen-Shannon, and function approximators, e.g., deep networks. In future work, we will examine more of these combinations to better understand the best marriages for each problem.

As suggested by prior work in computational neuroscience, interesting behavior reminiscent of transient cognitive dynamics emerges just outside the boundary of monotone systems (i.e., just one eigenvalue of the Jacobian is negative) [91]. Can a strong understanding of monotone equilibrium problems better equip us to explore the space beyond their boundary? Other work argues that *integrated information* possibly achieved via sensitivity to initial inputs is crucial (i.e., diverging dynamics are required) to an emerging consciousness [113]. More generally, the brain's intelligence is highly parallel and distributed which motivates research to move away from the monolithic learning formulations given by optimization and toward the multi-agent systems present in game theory and equilibration. We hope the tools presented here can help with the transition.

APPENDIX A

ONLINE MONOTONE EQUILIBRATION

This appendix serves as a supplement primarily to Chapter 2, however, we include additional proofs (e.g., the following section) and materials that may be of interest to the reader looking for more insight. In some cases, we consider a more general path integral loss starting at a vector o deemed a *reference vector* rather than the standard x^*:

$$f_o(x) = f_o(o) + \int_{o \to x} \langle F(z), dz \rangle. \tag{A.1}$$

A.1 Pseudo-monotonicity in Integral Form

Definition 2 (Pseudo-monotone)**.** *F is pseudo-monotone if the following one-way implication holds for all $x, y \in \mathcal{X}$: $\langle F(x), y - x \rangle \geq 0 \implies \langle F(y), y - x \rangle \geq 0$.*

Lemma 2. *If F is pseudo-monotone, F also obeys the following one-way implication*

$$\langle F(x), y - x \rangle \geq 0 \implies \int_{z: x \to y} \langle F(z), dz \rangle \geq 0. \tag{A.2}$$

Proof. Assume $\langle F(x), y - x \rangle \geq 0$ and let $\Delta z = \frac{y-x}{n}$ for $n \in \mathbb{Z}^+$. Then

$$\langle F(x), y - x \rangle = \langle F(x), \frac{y - x}{n} \rangle \cdot n \tag{A.3}$$

$$= \langle F(x), \Delta z \rangle \cdot n \tag{A.4}$$

$$\geq 0 \tag{A.5}$$

$$\implies \langle F(x), i\Delta z \rangle \geq 0 \ \forall \ i \geq 0 \tag{A.6}$$

Let $i\Delta z = (x + i\Delta z) - x = \hat{y}_i - x$. Then

$$\langle F(x), i\Delta z\rangle \geq 0 \;\forall\; i \geq 0 \tag{A.7}$$

$$\implies \langle F(x), \hat{y}_i - x\rangle \geq 0 \;\forall\; i \geq 0 \tag{A.8}$$

$$\implies \langle F(\hat{y}_i), \hat{y}_i - x\rangle \geq 0 \;\forall\; i \geq 0 \tag{A.9}$$

$$\implies \langle F(\hat{y}_i), i\Delta z\rangle \geq 0 \;\forall\; i \geq 0 \tag{A.10}$$

$$\implies \langle F(\hat{y}_i), \Delta z\rangle \geq 0 \;\forall\; i \geq 0 \tag{A.11}$$

$$\implies \sum_{i=0}^{n} \langle F(\hat{y}_i), \Delta z\rangle \geq 0 \tag{A.12}$$

$$\text{where } \Delta z = \Delta z(n) \tag{A.13}$$

$$\implies \lim_{n\to\infty} \sum_{i=0}^{n} \langle F(\hat{y}_i), \Delta z\rangle \geq 0 \tag{A.14}$$

$$= \int_{z:x\to y} \langle F(z), dz\rangle \geq 0. \tag{A.15}$$

□

A.2 Theorem 1: OCO ⊂ OMO

Let the feasible set, $\mathcal{X}$, and field, $F(x)$, be defined as follows:

$$x = [r, c] \in \mathcal{X} \equiv [0, 1]^2, \tag{A.16}$$

$$F(x) = \begin{pmatrix} r^2+2rc+c^2 \\ -2r^2+2rc+c^2 \end{pmatrix} \tag{A.17}$$

with equilibrium point $x^* = [0, 0]$.

A.2.1 F is monotone over $\mathcal{X} = [0, 1]^2$

The symmetric part of the Jacobian of F is positive semi-definite:

$$J(F) = \left(\begin{smallmatrix} 2r+2c & 2r+2c \\ -4r+2c & 2r+2c \end{smallmatrix}\right), \tag{A.18}$$

$$J_s(F) = \frac{1}{2}(J + J^\top) = \left(\begin{smallmatrix} 2r+2c & 2c-r \\ 2c-r & 2r+2c \end{smallmatrix}\right) \tag{A.19}$$

with

$$det(J_s) = 3r^2 + 12rc \geq 0 \ \forall \ [r, c] \in \mathcal{X}, \tag{A.20}$$

$$\tau(J_s) = 4r + 4c \geq 0 \ \forall \ [r, c] \in \mathcal{X}, \tag{A.21}$$

$$\implies J_s(F) \succeq 0. \tag{A.22}$$

The trace and determinant of the (2×2 matrix) symmetrized J_s are both non-negative, which imply the eigenvalues of J_s are non-negative. Therefore, F is monotone. □

A.2.2 f is non-convex over $\mathcal{X} = [0, 1]^2$

The path integral over the field F starting at x^* is

$$f(x) = \cancel{f_o} + \int_{z:o\to x} \langle F, dz\rangle \tag{A.23}$$

$$= \int_0^1 \langle F(o + \tau(x - o)), (x - o)d\tau\rangle \tag{A.24}$$

$$= \int_0^1 \langle F(\tau x), x\rangle d\tau \tag{A.25}$$

$$= \frac{1}{3}(r^3 + 3rc^2 + c^3) \tag{A.26}$$

with Hessian

$$H(f) = \left(\begin{smallmatrix} 2r & 2c \\ 2c & 2r+2c \end{smallmatrix}\right), \tag{A.27}$$

$$det(H) = 4(r^2 + rc - c^2) < 0 \forall \ \{[r, c] \,|\, [r, c] \in \mathcal{X}, c > \frac{\sqrt{5}+1}{2}r\} \tag{A.28}$$

$$\implies H \not\succeq 0. \tag{A.29}$$

This means f forms a saddle surface over a compact subset of $\mathcal{X}$, therefore, it is non-convex. In fact, f is not even quasi-convex. For example, let $x_0 = [0, 0.8], x_f = [0.5, 0.45]$ and consider their midpoint, then

$$f(\frac{x_0 + x_f}{2}) \nleq \max\{f(x_0), f(x_f)\}. \tag{A.30}$$

The following example provides a field whose path integral is non-convex over an unconstrained domain. The field F and its Jacobian are shown below:

$$F = [2x + \frac{2}{\pi}\sin(\frac{\pi}{2}y), 2y + \frac{2}{\pi}\sin(\frac{\pi}{2}x)], \tag{A.31}$$

$$J = \begin{bmatrix} 2 & \cos(\frac{\pi}{2}y) \\ \cos(\frac{\pi}{2}x) & 2 \end{bmatrix}, \tag{A.32}$$

$$J_{sym} = \begin{bmatrix} 2 & \frac{1}{2}(\cos(\frac{\pi}{2}x) + \cos(\frac{\pi}{2}y)) \\ \frac{1}{2}(\cos(\frac{\pi}{2}x) + \cos(\frac{\pi}{2}y)) & 2 \end{bmatrix} \succeq \mathbb{1} \tag{A.33}$$

with $x^* = [0, 0]$. The path integral over this field and its indefinite Hessian are

$$f = x^2 + y^2 - \frac{4}{\pi^2}(\frac{y}{x}(\cos(\frac{\pi}{2}x) - 1) + \frac{x}{y}(\cos(\frac{\pi}{2}y) - 1)), \tag{A.34}$$

$$H = \begin{bmatrix} 2 - \frac{y}{x}\left(\frac{8(\cos(\frac{\pi x}{2})-1)}{(\pi x)^2} + \frac{4\sin(\frac{\pi x}{2})}{\pi x} - \cos(\frac{\pi x}{2})\right) & 4\left(\frac{\cos(\frac{\pi x}{2})-1}{(\pi x)^2} + \frac{\sin(\frac{\pi x}{2})}{2\pi x} + \frac{\cos(\frac{\pi y}{2})-1}{(\pi y)^2} + \frac{\sin(\frac{\pi y}{2})}{2\pi y}\right) \\ 4\left(\frac{\cos(\frac{\pi x}{2})-1}{(\pi x)^2} + \frac{\sin(\frac{\pi x}{2})}{2\pi x} + \frac{\cos(\frac{\pi y}{2})-1}{(\pi y)^2} + \frac{\sin(\frac{\pi y}{2})}{2\pi y}\right) & 2 - \frac{x}{y}\left(\frac{8(\cos(\frac{\pi y}{2})-1)}{(\pi y)^2} + \frac{4\sin(\frac{\pi y}{2})}{\pi y} - \cos(\frac{\pi y}{2})\right) \end{bmatrix}, \tag{A.35}$$

$$H|_{x=1,y=10} = \begin{bmatrix} 2 - \frac{40}{\pi}(1 - \frac{2}{\pi}) & \frac{2(25\pi-51)}{25\pi^2} \\ \frac{2(25\pi-51)}{25\pi^2} & 2 - \frac{1}{10} + \frac{2}{125\pi^2} \end{bmatrix} = \begin{bmatrix} -2.626 & 0.223 \\ 0.223 & 1.902 \end{bmatrix} \not\succeq 0. \tag{A.36}$$

A.3 Theorem 2: OME ≡ OCO for Positive definite Affine Maps

This concerns such problems as linear complementarity problems (LCPs):

$$F_t(x_t) = Ax_t + b \tag{A.37}$$

where $x_t, b, o_t \in \mathbb{R}^n,\ A \succ 0 \in \mathbb{R}^{n\times n}$. The path integral over the field F_t starting at o_t (e.g., $o_t = x_t^*$) is a quadratic function,

$$\begin{aligned}
&f_t(x_t) - f_{o_t} = && \text{(A.38)}\\
&= \int_{x:o_t\to x_t} \langle F_t, dx\rangle && \text{(A.39)}\\
&= \int_0^1 \langle F_t(o_t + \tau(x_t - o_t)), (x_t - o_t)d\tau\rangle && \text{(A.40)}\\
&= \int_0^1 \langle A(o_t + \tau(x_t - o_t)) + b, (x_t - o_t)d\tau\rangle && \text{(A.41)}\\
&= \int_0^1 \langle Ao_t + \tau A(x_t - o_t) + b, (x_t - o_t)d\tau\rangle && \text{(A.42)}\\
&= \int_0^1 \langle Ao_t + b, (x_t - o_t)d\tau\rangle \\
&\qquad + \tau\langle A(x_t - o_t), (x_t - o_t)d\tau\rangle && \text{(A.43)}\\
&= \langle Ao_t + b, (x_t - o_t)\rangle \\
&\qquad + \frac{1}{2}\langle A(x_t - o_t)), (x_t - o_t)\rangle && \text{(A.44)}\\
&= o_t^\top A^\top x_t - o_t^\top A^\top o_t + b^\top(x_t - o_t) \\
&\qquad + \frac{1}{2}(x_t - o_t)^\top A^\top (x_t - o_t) && \text{(A.45)}\\
&= o_t^\top A^\top x_t - o_t^\top A^\top o_t + b^\top(x_t - o_t) \\
&\quad \frac{1}{2}[x_t^\top A^\top x_t - o_t^\top A^\top x_t - x_t^\top A^\top o_t + o_t^\top A^\top o_t] && \text{(A.46)}\\
&= \frac{1}{2}[x_t^\top A^\top x_t + x_t^\top (A - A^\top)o_t - o_t^\top A^\top o_t] + b^\top(x_t - o_t) && \text{(A.47)}\\
&= \frac{1}{2}[x_t^\top \Big(\frac{A + A^\top}{2}\Big) x_t + x_t^\top (A - A^\top)o_t - o_t^\top A^\top o_t] + b^\top(x_t - o_t), && \text{(A.48)}
\end{aligned}$$

with positive definite Hessian

$$Hessian(f_t) = \frac{1}{2}[A + A^\top] \succ 0 \implies f_t \text{ is convex.} \tag{A.49}$$

This also implies that every multivariate function with nonzero Hessian can be represented by an infinite number of fields (other than the gradient), specifically any field whose symmetric component equals $\frac{A+A^\top}{2}$.

A.4 Monotone Equilibration with $\mathbf{o} = \mathbf{x}^*$

Here, we consider the case where the reference point is the solution to the corresponding variational inequality problem, $o = x^* = VI(F, \mathcal{X})$. Remember, this means x^* is an equilibrium point of the field, F, and has the property

$$\langle F(x^*), z - x^* \rangle \geq 0 \; \forall z \in \mathcal{X}. \tag{A.50}$$

Theorem 4. *If o is a solution to $VI(F, \mathcal{X})$ where $F : \mathcal{X} \to \mathbb{R}^n$ is a monotone (or at least pseudo-monotone) map and $\mathcal{X}$ is a convex set, then o is a global minimizer of the monotone optimization problem with map F, reference vector o, and any reference scalar f_{ot}.*

Proof. Without loss of generality, let $f(o) = 0$. Then

$$\nabla_x \Big\{ \int_{z:o\to x} \langle F(z), dz \rangle \Big\} \Big|_{x=o} = \tag{A.51}$$

$$= \nabla_x \Big\{ \int_{t:0\to 1} \langle F(o + t(x-o)), x - o \rangle dt \Big\} \Big|_{x=o} \tag{A.52}$$

$$= \int_{t:0\to 1} \Big\{ F(o + t(x-o)) + \tag{A.53}$$

$$J(o + t(x-o))^\top (x-o) t dt \Big\} \Big|_{x=o} \tag{A.54}$$

$$= \int_{t:0\to 1} \Big\{ F(o) \Big\} \tag{A.55}$$

$$= F(o). \tag{A.56}$$

A necessary first order condition for optimality is

$$\langle F(o), z - o \rangle \geq 0 \; \forall z \in \mathcal{X}, \tag{A.57}$$

which by the definition of the variational inequality problem is solved by $o = x^*$. Reversing this result

$$f(o) = 0 \leq \langle F(o), z - o \rangle \leq \int_{x:o \to z} \langle F(x), dx \rangle = f(x) \tag{A.58}$$

reveals that $o = x^*$ is also a global minimum. □

This directly implies that the Projection Method (online gradient descent with $R = \frac{1}{2\eta}||x^2||$) converges exponentially fast to a minimum of the path integral loss for strongly monotone fields.

Note that the optimality proof also carries through for pseudo-monotone fields (see A.1):

$$f(o) = 0 \leq \langle F(o), z - o \rangle \tag{A.59}$$

$$\implies 0 \leq \int_{x:o \to z} \langle F(x), dx \rangle = f(x) \tag{A.60}$$

which reveals that $o = x^*$ is also a global minimum of the pseudo-monotone loss.

A.5 Upper and Lower Bounds for Path Integral Loss

In this section, we derive linear lower and upper bounds for the path integral loss assuming $\hat{x} = \texttt{prox}(x)$. First, we review proximal maps. We assume F_t has bounded norm, i.e., $||F_t(y)||_q \leq L_t \; \forall y$, and is β_t-smooth, i.e., $||F_t(x) - F_t(y)||_q \leq \beta_t ||x - y||_p \; \forall x, y$ for some $p \in [1, 2]$ and some q such that $1/p + 1/q = 1$.

A.5.1 Proximal Maps

We assume the following form for the proximal map:

$$\texttt{prox}(x) = \arg\min_{z \in \mathbb{R}^n} \Big(\langle F(x), z \rangle + \frac{1}{\hat{\eta}} D(z, x) + \iota_{\mathcal{X}}(z) \Big) \tag{A.61}$$

$$= \arg\min_{z \in \mathcal{X}} \Big(\langle F(x), z \rangle + \frac{1}{\hat{\eta}} D(z, x) \Big) \tag{A.62}$$

$$= \arg\min_{z \in \mathcal{X}} g(z, x), \tag{A.63}$$

$$D(z, x) = \psi(z) - \psi(x) - \langle \psi'(x), z - x \rangle, \tag{A.64}$$

where $\iota_{\mathcal{X}}$ is the indicator function,

$$\iota_{\mathcal{X}}(x) = \begin{cases} 0, & \text{if } x \in \mathcal{X} \\ \infty, & \text{otherwise,} \end{cases} \tag{A.65}$$

$D(z, x)$ is a Bregman divergence, and $\psi : \mathbb{R}^n \to R$ is m-strongly-convex w.r.t. the p-norm which implies

$$D(z, x) \geq D(x, x) + \langle D'(x, x), z - x \rangle + \frac{m}{2} ||z - x||_p^2 \tag{A.66}$$

$$\geq \frac{m}{2} ||z - x||_p^2 \tag{A.67}$$

$$= \frac{m}{2} ||z - x||_2^2 = \tilde{D}(z, x), \tag{A.68}$$

where the last step follows from the fact that $||y||_{d+a} \leq ||y||_d$ for all $a \geq 0$ and $d > 0$ along with our constraint that $p \in [1, 2]$.

Let $\texttt{prox}$ defined with $\tilde{D}$ be $\texttt{prox}^w$, i.e., a prox operator with weakened divergence. The proximal operator can be viewed as minimizing $\langle \nabla F(x), z \rangle$ with a penalty given

by $\frac{1}{\hat{\eta}}D(z,x)$ for deviating too far from x. Therefore, $\texttt{prox}^w(x)$ will be no closer to x than $\texttt{prox}(x)$. This implies

$$||x_t - \texttt{prox}(x_t)||_q \leq ||x_t - \texttt{prox}^w(x_t)||_q \tag{A.69}$$

$$= ||x_t - x_t + \frac{\hat{\eta}}{m}F_t(x_t)||_q \tag{A.70}$$

$$\leq \frac{\hat{\eta}}{m}L_t. \tag{A.71}$$

A.5.2 Lower Bounds

We use the following Lemma in building a lower bound for the path integral loss.

Lemma 3 (`Prox` Segment Lower Bound)**.** *The lower bound for the path integral $\int_{\hat{x}_t \to x_t}\langle F(z), dz\rangle$ is further lower bounded as follows:*

$$\langle F_t(\hat{x}_t), x_t - \hat{x}_t\rangle \geq \Big(\frac{m}{\hat{\eta}} - \beta_t\Big)||\hat{x}_t - x_t||_p^2, \tag{A.72}$$

where m is the strong-convexity parameter of the Bregman divergence used to form the proximal operator and β_t is the smoothness coefficient for the map F_t.

Proof. Let $G = g(x_t, x_t) - g(\hat{x}_t, x_t) = \langle F(x_t), x_t\rangle - \langle F(x_t), \hat{x}_t\rangle - \frac{1}{\hat{\eta}}D(\hat{x}_t, x_t) \geq 0$ where $\hat{x}_t = \texttt{prox}(x_t)$. Note that the minimization problem associated with $\texttt{prox}(x)$, $\min_{z\in\mathcal{X}} g(z,x)$, is a strongly-convex optimization problem over a convex set. Solutions, $\hat{x}_t$, to this problem enjoy the property that the derivative of g at $\hat{x}_t$ is in the normal cone, C, at $\hat{x}_t$:

$$-\frac{\partial g(z, x_t)}{\partial z}\Big|_{\hat{x}_t} \in C(\hat{x}_t), \tag{A.73}$$

$$C(\hat{x}_t) = \{y \in \mathbb{R}^n | \langle y, x - \hat{x}_t\rangle \leq 0,\ \forall x \in \mathcal{X}\}, \tag{A.74}$$

which, by definition of the normal cone, directly implies

$$\langle -\frac{\partial g(z, x_t)}{\partial z}\Big|_{\hat{x}_t}, x - \hat{x}_t \rangle \leq 0 \tag{A.75}$$

$$\langle F(x_t) + \frac{1}{\hat{\eta}} \frac{\partial D(z, x_t)}{\partial z}\Big|_{\hat{x}_t}, x - \hat{x}_t \rangle \geq 0 \tag{A.76}$$

$$\implies \langle F(x_t), x - \hat{x}_t \rangle \geq -\frac{1}{\hat{\eta}} \langle D'(\hat{x}_t, x_t), x - \hat{x}_t \rangle \tag{A.77}$$

where we have switched to a shorthand representation of the derivative in the last step for the sake of exposition. This allows us to lower bound the gap G by

$$G = \langle F(x_t), x_t - \hat{x}_t \rangle - \frac{1}{\hat{\eta}} D(\hat{x}_t, x_t) \tag{A.78}$$

$$= -\frac{1}{\hat{\eta}} \Big[\langle D'(\hat{x}_t, x_t), x - \hat{x}_t \rangle + D(\hat{x}_t, x_t) \Big]. \tag{A.79}$$

Revisiting the strong-convexity of D and swapping z and x gives

$$D(x, x) \geq D(z, x) + \langle D'(z, x), x - z \rangle + \frac{m}{2} ||z - x||_p^2 \tag{A.80}$$

$$0 \geq D(z, x) + \langle D'(z, x), x - z \rangle + \frac{m}{2} ||z - x||_p^2 \tag{A.81}$$

$$\implies D(\hat{x}_t, x) + \langle D'(\hat{x}_t, x), x - \hat{x}_t \rangle \leq -\frac{m}{2} ||\hat{x}_t - x||_p^2. \tag{A.82}$$

Plugging this back into the lower bound for G gives

$$G \geq \frac{m}{2\hat{\eta}} ||\hat{x}_t - x||_p^2. \tag{A.83}$$

Rearranging gives $\langle F(x_t), x_t - \hat{x}_t \rangle = G + \frac{1}{\hat{\eta}} D(\hat{x}_t, x_t)$. Therefore, we have

$$\beta_t ||\hat{x}_t - x_t||_p^2 \geq ||F_t(\hat{x}_t) - F_t(x_t)||_q ||\hat{x}_t - x_t||_p \tag{A.84}$$

$$\geq \langle F_t(\hat{x}_t) - F_t(x_t), \hat{x}_t - x_t \rangle \tag{A.85}$$

$$\implies \langle F_t(\hat{x}_t), x_t - \hat{x}_t \rangle \geq \langle F_t(x_t), x_t - \hat{x}_t \rangle - \beta_t ||\hat{x}_t - x_t||^2 \tag{A.86}$$

$$\geq G + \frac{1}{\hat{\eta}} D(\hat{x}_t, x_t) - \beta_t ||\hat{x}_t - x_t||^2 \tag{A.87}$$

where the first lines follow from the Cauchy-Schwarz inequality for the dual norm and the β_t-smoothness property of F_t. Using the strong-convexity of D and then the lower bound for G we have,

$$\langle F_t(\hat{x}_t), x_t - \hat{x}_t \rangle \geq G + \Big(\frac{m}{2\hat{\eta}} - \beta_t\Big)||\hat{x}_t - x_t||_p^2 \tag{A.88}$$

$$\geq \Big(\frac{m}{\hat{\eta}} - \beta_t\Big)||\hat{x}_t - x_t||_p^2. \tag{A.89}$$

□

Now let $F_t^{eff}(x) = (x - \hat{x}_t)\frac{m}{\hat{\eta}}$. Notice that in the unconstrained setting, $F_t^{eff} = F_t$ while in the constrained setting F_t^{eff} represents the component of F_t projected onto the feasible set. We define this map because $F^{eff}(x) = 0$ implies x is a fixed point of the map F_t. In other work, F_t^{eff} is referred to as the *residue vector* [25, 85, 86]. We are now ready to derive a lower bound for the path integral loss:

$$\hat{f}_{x^*}(x) = \frac{1}{T}\sum_{t=1}^{T}\Big[\int_{z:x^*\to\hat{x}_t}\langle F_t(z), dz\rangle + \int_{z:\hat{x}_t\to x}\langle F_t(z), dz\rangle\Big] \tag{A.90}$$

$$\geq \frac{1}{T}\sum_{t=1}^{T}\Big[\langle F_t(x^*), \hat{x}_t - x^*\rangle + \langle F_t(\hat{x}_t), x - \hat{x}_t\rangle\Big] \tag{A.91}$$

$$= \frac{1}{T}\sum_{t=1}^{T}\Big[\langle F_t(x^*), x - x^*\rangle + \langle F_t(x^*), \hat{x}_t - x\rangle + \langle F_t(\hat{x}_t), x - \hat{x}_t\rangle\Big] \tag{A.92}$$

$$\geq \underbrace{\langle F(x^*), x - x^*\rangle}_{\geq 0} + \frac{1}{T}\sum_{t=1}^{T}\Big[\langle F_t(x^*), \hat{x}_t - x\rangle + \Big(\frac{m}{\hat{\eta}} - \beta_t\Big)||\hat{x}_t - x||_p^2\Big] \tag{A.93}$$

$$\geq \frac{1}{T}\sum_{t=1}^{T}\Big[-||F_t(x^*)||_q||\hat{x}_t - x||_p + \Big(\frac{m}{\hat{\eta}} - \beta_t\Big)||\hat{x}_t - x||_p^2\Big] \tag{A.94}$$

$$\geq \frac{1}{T}\sum_{t=1}^{T}\Big[-\frac{||F_t(x^*)||_q||F_t^{eff}(x)||_p}{m}\hat{\eta} + \Big(\frac{m}{\hat{\eta}} - \beta_t\Big)||\hat{x}_t - x||_2^2\Big] \tag{A.95}$$

$$\geq \frac{1}{T}\sum_{t=1}^{T}\Big[\Big(||F_t^{eff}(x)||_p - ||F_t(x^*)||_q\Big)||F_t^{eff}(x)||_p\frac{\hat{\eta}}{m} - \beta_t\frac{\hat{\eta}^2}{m^2}L_t^2\Big] \tag{A.96}$$

$$\geq \frac{1}{T}\sum_{t=1}^{T}\Big[\Big(||F_t^{eff}(x)||_p - ||F_t(x^*)||_q\Big)||F_t^{eff}(x)||_p\frac{\hat{\eta}}{m} - \beta_t\frac{\hat{\eta}^2}{m^2}L_t^2\Big]. \tag{A.97}$$

A.5.3 Upper Bounds

We are also able to obtain the following upper bound for the path integral loss:

$$\hat{f}_{x^*}(x) = \frac{1}{T}\sum_{t=1}^{T}\Big[\int_{z:x^*\to\hat{x}_t}\langle F_t(z), dz\rangle + \int_{z:\hat{x}_t\to x}\langle F_t(z), dz\rangle\Big] \tag{A.98}$$

$$\leq \frac{1}{T}\sum_{t=1}^{T}\Big[\langle F_t(\hat{x}_t), \hat{x}_t - x^*\rangle + \langle F_t(x), x - \hat{x}_t\rangle\Big] \tag{A.99}$$

$$= \frac{1}{T}\sum_{t=1}^{T}\Big[\langle F_t(\hat{x}_t), x - x^*\rangle + \langle F_t(\hat{x}_t), \hat{x}_t - x\rangle + \langle F_t(x), x - \hat{x}_t\rangle\Big] \tag{A.100}$$

$$= \frac{1}{T}\sum_{t=1}^{T}\Big[\langle F_t(\hat{x}_t), x - x^*\rangle + \langle F_t(x) - F_t(\hat{x}_t), x - \hat{x}_t\rangle\Big] \tag{A.101}$$

$$\leq \frac{1}{T}\sum_{t=1}^{T}\Big[\langle F_t(\hat{x}_t), x - x^*\rangle + ||F_t(x) - F_t(\hat{x}_t)||_q||x - \hat{x}_t||_p\Big] \tag{A.102}$$

$$\leq \frac{1}{T}\sum_{t=1}^{T}\Big[\langle F_t(\hat{x}_t), x - x^*\rangle + \beta_t||x - \hat{x}_t||_p^2\Big] \tag{A.103}$$

$$\leq \frac{1}{T}\sum_{t=1}^{T}\Big[\langle F_t(\hat{x}_t), x - x^*\rangle + \frac{\beta_t L_t^2}{m^2}\hat{\eta}^2\Big], \tag{A.104}$$

where we leveraged the monotone path integral bounds in the first step, rearranged terms, and then used Cauchy-Schwarz and the β_t-smoothness of F_t.

A.5.4 OED and OMP Regret Bounds

We repeat the bounds adopted from the work of Shalev-Shwartz [2011] for convenience.

Theorem 5. *Let R be a $(1/\eta)$-strongly-convex function over $\mathcal{X}$ with respect to a norm $||\cdot||$. Assume that $\mathcal{A} :=$OMP is run on the sequence of monotone maps, F_t, with the link function*

$$g(\theta) = \arg\max_{x\in\mathcal{X}}(\langle x, \theta\rangle - R(x)). \tag{A.105}$$

Then, for all $x^* \in \mathcal{X}$*,*

$$\textit{regret}_{\mathcal{A}}(\mathcal{X}) \leq R(x^*) - \min_{v \in \mathcal{X}} R(v) + (\eta + \frac{\beta_{\max}}{m^2}\hat{\eta}^2) \sum_{t=1}^{T} L_t^2 \tag{A.106}$$

$$\leq \frac{3}{2} BL\sqrt{2T} \textit{ for } R(x) = \frac{1}{2\eta}||x||_2^2, \tag{A.107}$$

where $||x^*||_2 \leq B, ||F_t||_q \leq L_t, L^2 \geq \frac{1}{T}\sum_t L_t^2, \beta_{\max} = \max_t \beta_t, \eta = \frac{B}{L\sqrt{2T}}$*, and* $\hat{\eta} = m\sqrt{\frac{\eta}{\beta_{\max}}}$*.*

Proof. As we have shown previously,

$$\textit{regret}_{\mathcal{A}_{(t,T)}}(\mathcal{X}) \leq \langle F_t(\hat{x}_t), x_t - x^* \rangle + \frac{\beta_t L_t^2}{m^2}\hat{\eta}^2 \tag{A.108}$$

$$\tag{A.109}$$

The OMP algorithm is equivalent to running Follow the Regularized Leader (FTRL) on the sequence of linear functions $\langle F(\hat{x}_t), x_t \rangle$ with the regularization $R(x)$. The theorem now follows directly from Theorem 2.11 and Lemma 2.6 in the work of Shalev-Shwartz [2011]. □

A.5.5 Combining Upper and Lower Bounds

Unfortunately, obtaining meaningful regret bounds is not as clean as in the simpler setting of online convex optimization. This is because our path integral loss is a function of our step size $\hat{\eta}$. If we set $\hat{\eta}$ to zero, then we can arbitrarily decrease our loss without obtaining any real performance gains. We will demonstrate this with an example later. To reiterate, our regret bound is only meaningful if minimizing regret implies improved performance with respect to some other, $\hat{\eta}$-independent, performance measure.

Lemma 4. *Minimizing the path integral loss for monotone maps at a rate* $\propto T^{-1/2}$ *implies*

$$||F_t^{eff}(x)||_p \leq ||F_t(x^*)||_q + CT^{-1/8} \tag{A.110}$$

on average where $C = (2^{11/8})(\beta_{\max} BL^3)^{1/4} < 2.6(\beta_{\max} BL^3)^{1/4}$.

Proof. Rewriting the lower and upper bounds together, we see

$$\frac{1}{T}\sum_{t=1}^{T}\Big[\Big(||F_t^{eff}(x)||_p - ||F_t(x^*)||_q\Big)||F_t^{eff}(x)||_p\frac{\hat{\eta}}{m} - \beta_t\frac{\hat{\eta}^2}{m^2}L_t^2\Big)\Big] \tag{A.111}$$

$$\leq \hat{f}_{x^*}(x) \tag{A.112}$$

$$\leq \frac{1}{T}\sum_{t=1}^{T}\Big[\langle F_t(\hat{x}_t), x - x^*\rangle + \frac{\beta_t L_t^2}{m^2}\hat{\eta}^2\Big]. \tag{A.113}$$

Let P be defined as follows:

$$P = \frac{1}{T}\sum_{t=1}^{T}\Big[\Big(||F_t^{eff}(x)||_p - ||F_t(x^*)||_q\Big)\frac{||F_t^{eff}(x)||_p}{m}. \tag{A.114}$$

Let $R(x) = \frac{1}{2\eta}||x||_2^2$. Rearranging the bounds and assuming the same FTRL regret rate above imply that

$$P \leq \frac{1}{T\hat{\eta}}\sum_{t=1}^{T}\Big[\langle F_t(\hat{x}_t), x - x^*\rangle + 2\frac{\beta_t L_t^2}{m^2}\hat{\eta}^2\Big] \tag{A.115}$$

$$\leq \frac{1}{T\hat{\eta}}\Big[\frac{B^2}{2\eta} + (\eta + \frac{2\beta_{\max}}{m^2}\hat{\eta}^2)TL^2\Big] = P^u. \tag{A.116}$$

Taking derivatives with respect to η and $\hat{\eta}$ and setting equal to zero gives:

$$\frac{\partial P^u}{\partial \eta} = \frac{1}{\hat{\eta}}\Big(- \frac{B^2}{2\eta^2} + TL^2\Big) = 0 \tag{A.117}$$

$$\implies \eta = \frac{B}{L\sqrt{2T}} \tag{A.118}$$

$$\frac{\partial P^u}{\partial \hat{\eta}} = -\frac{1}{\hat{\eta}^2}\Big(\frac{B^2}{2\eta} + \eta T L^2\Big) + \frac{2\beta_{\max}}{m^2} TL^2 = 0 \tag{A.119}$$

$$= -\frac{1}{\hat{\eta}^2}\Big(\frac{\sqrt{2}}{2} BL\sqrt{T} + \frac{\sqrt{2}}{2} BL\sqrt{T}\Big) + \frac{2\beta_{\max}}{m^2} TL^2 \tag{A.120}$$

$$= -\frac{1}{\hat{\eta}^2}\sqrt{2} BL\sqrt{T} + \frac{2\beta_{\max}}{m^2} TL^2 = 0 \tag{A.121}$$

$$\implies -\frac{1}{\hat{\eta}^2}\frac{B}{L\sqrt{2T}} + \frac{\beta_{\max}}{m^2} = 0 = -\frac{\eta}{\hat{\eta}^2} + \frac{\beta_{\max}}{m^2} \tag{A.122}$$

$$\implies \hat{\eta} = m\sqrt{\frac{\eta}{\beta_{\max}}} \tag{A.123}$$

$$\implies P \leq 4\frac{\sqrt{\beta_{\max} BL^3}}{m(2T)^{1/4}} = DT^{-1/4}. \tag{A.124}$$

Let $||F_t^{eff}(x)||_p = C_t ||F_t(x^*)||_q$. Then,

$$\implies \frac{1}{m}(C_t - 1)C_t ||F_t(x^*)||_q^2 \leq P \tag{A.125}$$

$$\implies C_t \leq 1 + \frac{T^{-1/8}}{||F_t(x^*)||_q}\sqrt{2mD} \to 1 \text{ at a rate } \propto T^{-1/8} \tag{A.126}$$

$$\implies ||F_t^{eff}(x)||_p \leq ||F_t(x^*)||_q + T^{-1/8}\sqrt{2mD} \text{ on average.} \tag{A.127}$$

□

To summarize, our algorithm minimizes regret at rate that implies that the average norm of the effective vector field at each step is approaching the norm of the vector field at optimality.

To demonstrate the importance of our choices for η and $\hat{\eta}$, consider solving $VI(F(x) = Ax, \mathbb{R}^n)$ where $A = -A^\top$ with $\eta = \hat{\eta} = T^{-1/2}$. Note that $x^* = 0$ is the unique solution to this problem. Then

$$x_{k+1} = \Big[(1 - \eta\hat{\eta})I - \eta A\Big] x_k = Jx_k \tag{A.128}$$

and the norm of the iterates change as

$$||x_{k+1}||^2 = x_k^\top J^\top J x_k = \Big[(1 - \eta\hat{\eta})^2 + \eta^2\Big] ||x_k||^2 \tag{A.129}$$

$$||x_T||^2 = \Big[(1 - \eta\hat{\eta})^2 + \eta^2\Big]^T ||x_0||^2 = \gamma^T ||x_0||^2. \tag{A.130}$$

In the limit as $T \to \infty$, γ^T converges to $\frac{1}{e}$! Therefore, the naive algorithm does not converge for this problem. On the other hand, if $\eta = T^{-1/2}$ and $\hat{\eta} = T^{-1/4}$, then $\gamma^T \to 0$ in the limit meaning, as expected, our proposed step size choice does converge.

A.6 A Curl Bound for a Different Path Integral

The path integral loss presented in the main body of the thesis requires knowledge of x^*. Ideally, we would like to observe some version of a loss in order to track progress during training and gain more insights into the performance of our algorithm. Instead of computing the path integral starting at x^*, we can consider starting at some other reference point, o_t, for which we have a good value estimate, i.e., $f(o_t)$ is known. Then we can later measure our performance relative to x^* by comparing two path integrals: one that integrates from o_t to x and one that integrates from o_t to x^*. Using this new definition, we can immediately derive a linear upper bound for the alternative loss as follows:

$$\begin{aligned} f^{alt}(x_t) &= \int_{o_t}^{x_t} - \int_{o_t}^{x^*} \\ &\leq \langle F(x_t), x_t - o_t \rangle - \langle F(o_t), x^* - o_t \rangle. \end{aligned} \tag{A.131}$$

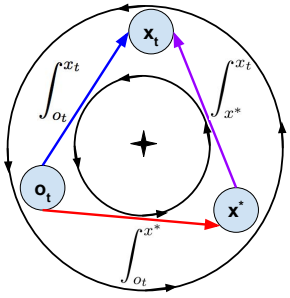


Figure A.1: Illustrative comparison of *two-step*, $\int_{o_t}^{x_t} - \int_{o_t}^{x^*}$, to *one-step* loss, $\int_{x^*}^{x_t}$.

In comparison, our original loss is,

$$f(x_t) = \int_{x^*}^{x_t} \leq \langle F(x_t), x_t - x^* \rangle = \langle F(x_t), x_t - o_t \rangle - \langle F(x_t), x^* - o_t \rangle. \tag{A.132}$$

Unfortunately, we cannot simplify the difference between the latter terms, $\langle F(x_t) - F(o_t), x^* - o_t \rangle$, and so it is not clear if one of these losses upper bounds the other. However, we can bound the difference between the two losses using Stokes' theorem and bounds on F and its derivatives. The difference between the two losses is equal to the magnitude of the path integral around the triangle in Figure A.1. By Whitney's extension of Stokes' theorem to perimeters with corners [116],

$$\begin{aligned} |\oint_{\partial\Sigma} \langle F, dx \rangle| = |\int_{\Sigma} \nabla \times F' \cdot d\Sigma| &\leq \max_{\Sigma} ||\nabla \times F'|| \cdot \int_{\Sigma} dA \\ &\leq \sigma_{\max}(J - J^\top) \cdot (\text{Area of } \triangle) \end{aligned} \tag{A.133}$$

where $\sigma_{\max}(A)$ denotes the maximum singular value of matrix A, F' is the projection of F onto the triangle $\triangle$, and Σ ($\partial\Sigma$) is the two dimensional area (perimeter) formed by the path around the triangle. The bound on the norm of the curl is proven in Lemma 5 below. Note that if we set $o_t = x^*$, the triangle collapses and the losses are the same. Also, if we set $o_t = x_{t-1}$, the triangle collapses in the limit as $T \to \infty$. This is because $||x_t - x_{t-1}|| \leq \frac{\eta}{m} L_t$ and $\eta \propto T^{-1/2}$. Using $o_t = x_{t-1}$ is particularly

appealing because in many online settings, we have access to all historical play and can compute an accurate estimate of the value of x_{t-1}.

The bound outlined above applies to the path integral loss that we used for strongly and strictly monotone maps. To bound the difference between path integrals constructed for monotone fields, we need to consider the following difference:

$$\overbrace{\Big(\int_{o_t \to \hat{x}_t} + \int_{\hat{x}_t \to x_t}\Big) - \Big(\int_{o_t \to \hat{x}_t^*} + \int_{\hat{x}_t^* \to x^*}\Big)}^{\hat{f}^{alt}(x_t)} - \overbrace{\Big(\int_{x^* \to \hat{x}_t} + \int_{\hat{x}_t \to x}\Big)}^{\hat{f}(x_t)} \tag{A.134}$$

$$= \int_{o_t \to \hat{x}_t} - \int_{x^* \to \hat{x}_t} - \int_{\hat{x}_t^* \to x*} - \int_{o_t \to \hat{x}^*} \tag{A.135}$$

where we have omitted the integrands, $\langle F(z), dz \rangle$, to avoid clutter. Notice that these four integrals trace out the path $o_t \to \hat{x}_t \to x^* \to \hat{x}^* \to o_t$. In order to use Stokes' theorem, we need to form a 2-manifold over the perimeter formed by the path. We can construct a surface manifold using two triangles: $(o_t, \hat{x}_t, x^*)$ and $(o_t, \hat{x}_t^*, x^*)$. The area of the second triangle is $\mathcal{O}(\hat{\eta})$ because $||\hat{x}_t^* - x^*|| \leq \frac{\hat{\eta}}{m} L_t$. In addition, the vertices of the first triangle approach (o_t, x, x^*) as $\hat{\eta}$ goes to zero.

Lemma 5 (Curl Bound)**.** *The curl of the vector field over any 2-manifold is bounded by the maximum singular value of two times the skew-symmetric part of the Jacobian:* $||\nabla \times F'||_2 \leq \sigma_{\max}(J - J^\top)$.

Proof. The proof outline is as follows. The triangle in Figure A.1 defines a 2-D plane in an n-dimensional ambient space. We are interested in the path integral around the triangle. Each element of the path integral consists of an inner product of the field F with a differential vector along the curve. Any components of F that are orthogonal to this differential vector evaluate to zero. The triangle is 2-D hence, its perimeter is 2-D, which means we may consider a projection of F, $F_{:2}$, onto the 2-D plane defined by the triangle. We can think of this projection as a rotation of F, $F^R = R \cdot F$, followed by a projection in which we extract the first two dimensions of $F_{:2} = \Pi_\Delta(F^R)$. The

curl is defined for 3-D so we will actually append a third dimension whose component is identically zero. Define F' to be this augmented projection of the field.

Using the fact that $\nabla \times F' = (J - J^\top)F'$ results in a vector that is perpendicular to F', we find that

$$(\nabla \times F') \times F' = (J(F') - J(F')^\top)F' \tag{A.136}$$

$$= ||\nabla \times F'||_2 ||F'||_2 \sin\theta \mathbf{n} = ||\nabla \times F'||_2 ||F'||_2 \mathbf{n}. \tag{A.137}$$

We can then bound the norm of the curl by recognizing the following as the induced 2-norm of a matrix:

$$||\nabla \times F'||_2 = ||(J(F') - J(F')^\top)F'||_2 / ||F'||_2 \tag{A.138}$$

$$\leq \sigma_{\max}(J(F') - J(F')^\top) \tag{A.139}$$

$$= ||J(F') - J(F')^\top||_2 \tag{A.140}$$

$$= ||J(F_{:2}) - J(F_{:2})^\top||_2 \quad \text{because } J(F') \text{ is simply } J(F_{:2}) \text{ augmented with zeros along the third row and third column.} \tag{A.141}$$

$$\leq ||J(F^R) - J(F^R)^\top||_2 \quad \text{because the principal submatrix just removes entries.} \tag{A.142}$$

$$= ||R(J(F) - J(F)^\top)||_2 \quad \text{by linearity of the Jacobian.} \tag{A.143}$$

$$\leq ||R||_2 ||J(F) - J(F)^\top||_2 \quad \text{L}_p \text{ induced norms } (p = \infty) \text{ are submultiplicative.} \tag{A.144}$$

$$\leq ||J(F) - J(F)^\top||_2 \quad \text{rotation matrix has unit spectral bound.} \tag{A.145}$$

$$= \sigma_{\max}(J - J^\top) = \sqrt{\lambda_{\max}(A^\top A)} \quad \text{where } A = J - J^\top. \tag{A.146}$$

□

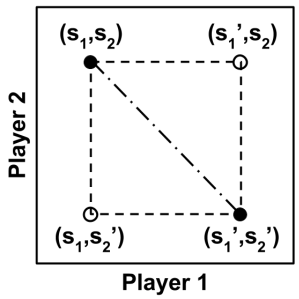


Figure A.2: Illustrative comparison of *auto*-welfare to a *game-agnostic* loss. Online optimization provides theory for regret measured only along the edges of the square (axis aligned), while online monotone equilibration additionally measures regret along diagonals (any line).

A.7 Online Monotone Games and *Auto*-Welfare

In this section, we define monotone games and *auto*-welfare.

Definition 3 (Monotone Game)**.** *A game is monotone if the map,* $F : \mathcal{X} \to \mathbb{R}^n$*, formed by concatenating the subgradients of all* N *player cost functions,* $f^{(i)}(x)$*, is monotone. More concretely, let* $F = [z^{(1)}, \ldots, z^{(N)}]$ *where* $z^{(i)} \in \partial_{x^{(i)}} f^{(i)}$ *is any subgradient of* $f^{(i)}$ *w.r.t.* $x^{(i)}$*. A game is monotone if* F *satisfies Equation* (1.26)*.*

Essentially, a game is monotone if gradient descent with an infinitesimally small step size, e.g., GIGA [125], does not cause the player strategies to diverge away from the equilibrium point. In online monotone games, an adversary may choose a new monotone game for the players to play at every time step, i.e., $f^{(i)}(x)$ becomes $f_t^{(i)}(x)$.

We begin our discussion of regret with a trivial application of online optimization to games. Online optimization provides theory for bounding the regret of an algorithm's prediction, x_t, when comparing to a baseline, x^*, chosen in retrospect. In other words, if we were to go back in time and play this baseline against the same exact sequence of environments, how much better would we do? To measure this, we can sum up the differences between the algorithm's loss and the baseline loss at each time step, t, as in OCO (see Algorithm 1).

In order to use this theory in a game, we simply focus our attention on one player and treat the rest of the players as part of the adversarial environment. In this way, online optimization can provide regret bounds for each player if we imagine replaying the game but with all other players forced to replay the same actions as before. This is largely unsatisfying given that it seems to have taken the game aspect out of the game. Ideally, our regret measure would leave the game environment intact and allow all players to change their actions. In this regard, welfare regret is far more satisfying because it measures the sum of all player payoffs with respect to a baseline that allows all players to change their actions. Unfortunately, bounding welfare regret often requires properties like smoothness.

As a compromise, we propose *auto*-welfare. Consider player 1 in an N-player game. Player 1 receives a payoff or *reward* for changing her strategy, however, her reward depends on all other player adjustments as well. Player 1 never knows how the other $N-1$ players are going to change their strategies, so it is reasonable for her to measure the portion of her reward that is due to her strategy change alone. Such a measurement provides valuable feedback on her decision to update her strategy, and this measurement is exactly what *auto*-welfare sums for all players. Therefore, *auto*-welfare can be thought of as measuring how "satisfied" the players as a whole are with their decision making given that they only have control over themselves. In contrast, welfare measures how "satisfied" the players as a whole are with the outcome of the game.

We can compute *auto*-welfare, W^a, with a path integral,

$$W_t^a(x_t) = W_{o_t}^a + \int_{x:o_t \to x_t} \langle -F_t(x), dx \rangle, \tag{A.147}$$

where $F_t(x)$ is an output of the game map (see Definition 3), o_t is any reference vector with known *auto*-welfare, $W_{o_t}^a$, and $x : o_t \to x_t$ is the straight line path from o_t to x_t through $\mathcal{X}$. Figure A.2 illustrates the flexibility *auto*-welfare provides over the

game-agnostic loss provided by online optimization theory. This formulation has been considered for converting symmetric VIs into equivalent optimization problems [2, 46], however, to our knowledge has not been leveraged for asymmetric VIs, which represent a wider class of games.

We can rewrite *auto*-welfare as follows to reveal its relationship to standard welfare, W:

$$W_t^a(x_t) = W_{o_t}^a + \int_{x:o_t \to x_t} \langle -F_t(x), dx \rangle \tag{A.148}$$

$$= W_{o_t}^a + \sum_{i=1}^{N} \int_{x:o_t \to x_t} \langle -\partial f_{t,i}^{(i)}(x), dx^{(i)} \rangle \tag{A.149}$$

$$= W(x_t) - \sum_{i=1}^{N} \sum_{j \neq i}^{N} \underbrace{\int_{x:o_t \to x_t} \langle -\partial f_{t,j}^{(i)}(x), dx^{(j)} \rangle}_{i\text{'s reward due to } j\text{'s strategy change}} \tag{A.150}$$

where $\partial f_{t,j}^{(i)}(x)$ is a subgradient of agent i's expected loss function with respect to agent j's strategy, $x_t^{(j)}$, evaluated at x. Therefore, $W_t^a(x_t)$ gives welfare minus the rewards resulting from intra-team inefficiencies. We call this *auto*-welfare because it sums the portions of the player's welfare that can be attributed to its own strategy.

A.8 Algorithmic Game Theory: A Venn Diagram

Here, we consider cost-minimization games where $C_i(\mathbf{s})$ is player i's cost function and $C(\mathbf{s}) = \sum_{i=1}^{K} C_i(\mathbf{s})$. Player i's strategy set is $\mathbf{s}_i$ and $\mathbf{s}_{-i}$ represents the strategy sets of all players except player i. The results of this section are summarized in Table 2.1 in the main body of the thesis.

Definition 4 (Smooth Game)**.** *A cost-minimization game is* (λ,μ)-smooth *if for every two outcomes* $\mathbf{s}$ *and* $\mathbf{s}^*$*,*

$$\sum_{i=1}^{K} C_i(\mathbf{s}_i^*, \mathbf{s}_{-i}) \leq \lambda \cdot C(\mathbf{s}^*) + \mu \cdot C(\mathbf{s}). \tag{A.151}$$

Definition 5 (Convex Game)**.** *A cost-minimization game is convex if $C_i(\mathbf{s}_i, \mathbf{s}_{-i})$ is convex in $\mathbf{s}_i \; \forall i$.*

Definition 6 (Monotone Game)**.** *A cost-minimization game is monotone if the game dynamics are monotone. Here, we assume all players are running OMP, i.e.,*

$$F = \begin{pmatrix} \nabla_{\mathbf{s}_0} C_0 \\ \cdots \\ \nabla_{\mathbf{s}_K} C_K \end{pmatrix}. \tag{A.152}$$

Monotonicity requires that the symmetrized Jacobian of F be positive semidefinite: $J + J^\top \succeq 0$.

Definition 7 (Socially-Convex Game)**.** *A cost-minimization game is socially-convex if*

1. *There exists $\lambda_i > 0$ such that $\sum_{i=1}^K \lambda_i = 1$, $g(\mathbf{s}) = \sum_{i=1}^K \lambda_i C_i(\mathbf{s})$ is convex in $\mathbf{s}$, and*
2. *$C_i(\mathbf{s}_i, \mathbf{s}_{-i})$ is concave in $\mathbf{s}_{-i} \; \forall i$.*

The definition was originally written for concave.

Theorem 6 (Monotone $\implies$ Convex)**.** *If a game is monotone, it is also convex.*

Proof. For each player i, we show that $C_i(\mathbf{s}_i, \mathbf{s}_{-i})$ is convex in $\mathbf{s}_i$ for any fixed $\mathbf{s}_{-i}$ (i.e., $\mathbf{s}_{-i} = \mathbf{s}'_{-i}$). The associated map is given by Equation (A.152). Let $\mathcal{X}_i \coloneqq \{\mathbf{s}, \mathbf{s}' \in \mathcal{X} \; s.t. \; \mathbf{s}_{-i} = \mathbf{s}'_{-i}\}$. Starting with the definition of monotonicity, we have

$$\langle F(\mathbf{s}) - F(\mathbf{s}'), \mathbf{s} - \mathbf{s}' \rangle \geq 0 \ \forall \ \mathbf{s}, \mathbf{s}' \in \mathcal{X} \tag{A.153}$$

$$\implies \langle F(\mathbf{s}) - F(\mathbf{s}'), \mathbf{s} - \mathbf{s}' \rangle \geq 0 \forall \ \mathbf{s}, \mathbf{s}' \in \mathcal{X}_i \tag{A.154}$$

$$= \sum_j \langle F_j(\mathbf{s}) - F_j(\mathbf{s}'), \mathbf{s}_j - \mathbf{s}'_j \rangle \geq 0 \ \forall \ \mathbf{s}, \mathbf{s}' \in \mathcal{X}_i \tag{A.155}$$

$$= \sum_{j \neq i} \langle F_j(\mathbf{s}) - F_j(\mathbf{s}'), \underbrace{\mathbf{s}_j - \mathbf{s}'_j}_{0} \rangle \geq 0 \ \forall \ \mathbf{s}, \mathbf{s}' \in \mathcal{X}_i \tag{A.156}$$

$$+ \langle F_i(\mathbf{s}) - F_i(\mathbf{s}'), \mathbf{s}_i - \mathbf{s}'_i \rangle \geq 0 \ \forall \ \mathbf{s}, \mathbf{s}' \in \mathcal{X}_i \tag{A.157}$$

$$= \langle \nabla C_i(\mathbf{s}) - \nabla C_i(\mathbf{s}'), \mathbf{s}_i - \mathbf{s}'_i \rangle \geq 0 \ \forall \ \mathbf{s}, \mathbf{s}' \in \mathcal{X}_i \tag{A.158}$$

which is the definition of convexity, so C_i is convex in $\mathbf{s}_i$. □

Theorem 7 (Socially-Convex $\implies$ Convex)**.** *Lemma 2.2 in the work of Even-Dar et al. [2009] with convex swapped for concave.*

Theorem 8 (Socially-Convex $\implies$ $\lambda-$Monotone)**.** *A game that is socially-convex with parameters λ implies a scaling of the game with the same parameters that is monotone (credit to Peng Shi).*

Proof. Let $C'_i = \lambda_i C_i$ and let J' be the Jacobian of the map, F', corresponding to C'_i (see Equation (A.152)). In addition, define the following matrices

1. D such that $D_{ii} = \lambda_i \frac{\partial^2 C_i}{\partial \mathbf{s}_i^2}$ and $D_{ij} = 0 \ \forall i \neq j$.

2. G^k is such that $\forall i \ G^k_{ik} = G^k_{ki} = 0$ and $\forall i \neq k, j \neq k$, $G^k_{ij} = \lambda_k \frac{\partial^2 C_k}{\partial \mathbf{s}_i \mathbf{s}_j}$.

3. H is the Hessian of $g(\mathbf{s}) = \sum_k \lambda_k C_k(\mathbf{s})$ (i.e., $H_{ij} = \sum_k \lambda_k \frac{\partial^2 C_k}{\partial \mathbf{s}_i \mathbf{s}_j}$).

Note that D, $-G^k$, and H are all positive semidefinite matrices. This follows from the fact that player costs are convex in their own strategies, concave in other players' strategies, and the socially-convex condition respectively.

Continuing the proof, for every $i \neq j$,

$$(D - \sum_k G^k + H)_{ij} = 0 - \sum_{k \neq i,j} \lambda_k \frac{\partial^2 C_k}{\partial \mathbf{s}_i \mathbf{s}_j} + \sum_k \lambda_k \frac{\partial^2 C_k}{\partial \mathbf{s}_i \mathbf{s}_j} \tag{A.159}$$

$$= \lambda_i \frac{\partial^2 C_i}{\partial \mathbf{s}_i \mathbf{s}_j} + \lambda_j \frac{\partial^2 C_j}{\partial \mathbf{s}_i \mathbf{s}_j} \tag{A.160}$$

$$= (J' + J'^T)_{ij}. \tag{A.161}$$

Moreover, for every $i = j$,

$$(D - \sum_k G^k + H)_{ii} = \frac{\partial^2 C_i}{\partial \mathbf{s}_i^2} - \sum_{k \neq i} \lambda_k \frac{\partial^2 C_k}{\partial \mathbf{s}_i^2} + \sum_k \lambda_k \frac{\partial^2 C_k}{\partial \mathbf{s}_i^2} \tag{A.162}$$

$$= 2\lambda_i \frac{\partial^2 C_i}{\partial \mathbf{s}_i^2} \tag{A.163}$$

$$= (J' + J'^T)_{ii}. \tag{A.164}$$

Therefore, $(J + J'^T) = D - \sum_k G^k + H$. Each of the matrices $(D, -G^k, H)$ is positive semidefinite, therefore $(J + J'^T) \succeq 0$, hence F' is monotone. □

A.8.1 a. Smooth

The following cost-minimization game is $(\frac{1}{2}, \frac{1}{2})$-smooth:

$$C_1 = C_2 = -\cos(r) - \cos(c). \tag{A.165}$$

Proof.

$$\sum_{i=1}^{K} C_i(s_i^*, \mathbf{s}_{-i}) = -\cos(r^*) - \cos(c) - \cos(r) - \cos(c^*) \tag{A.166}$$

$$\leq \lambda \cdot C(\mathbf{s}^*) = -\cos(r^*) - \cos(c^*) \tag{A.167}$$

$$+ \mu \cdot C(\mathbf{s}) = -\cos(r) - \cos(c) \tag{A.168}$$

$$= -\cos(r^*) - \cos(c) - \cos(r) - \cos(c^*). \tag{A.169}$$

□

This game is not convex, therefore, it is neither monotone nor socially-convex.

A.8.2 b. Smooth, Convex

Consider the following cost-minimization game:

$$C_1 = r^2(\sin(c) + 1.25) \tag{A.170}$$

$$C_2 = c^2(\sin(r) + 1.25). \tag{A.171}$$

This game is $(10, 0)$-smooth.

Proof.

$$\sum_{i=1}^{K} C_i(s_i^*, \mathbf{s}_{-i}) = r^{*2}(\sin(c) + 1.25) + c^{*2}(\sin(r) + 1.25) \tag{A.172}$$

$$\leq 2.25(r^{*2} + c^{*2}) \tag{A.173}$$

$$\lambda \cdot C(\mathbf{s}^*) = 10[r^{*2}(\sin(c^*) + 1.25) + c^{*2}(\sin(r^*) + 1.25)] \tag{A.174}$$

$$\geq 2.5(r^{*2} + c^{*2}) \tag{A.175}$$

$$2.25(r^{*2} + c^{*2}) \leq 2.5(r^{*2} + c^{*2}). \tag{A.176}$$

□

Clearly, C_1 is convex in r and C_2 is convex in c, therefore the game is convex.

The corresponding map is not monotone:

Proof.

$$F = \begin{pmatrix} 2r(\sin(c)+1.25) \\ 2c(\sin(r)+1.25) \end{pmatrix} \tag{A.177}$$

$$J = 2\begin{pmatrix} \sin(c)+1.25 & r\cos(c) \\ c\cos(r) & \sin(r)+1.25 \end{pmatrix} \tag{A.178}$$

$$J_s = 2\begin{pmatrix} \sin(c)+1.25 & \frac{r}{2}\cos(c)+\frac{c}{2}\cos(r) \\ \frac{r}{2}\cos(c)+\frac{c}{2}\cos(r) & \sin(r)+1.25 \end{pmatrix} \tag{A.179}$$

$$J_s\Big|_{r=c=-\frac{\pi}{4}} = 2\begin{pmatrix} -\frac{\sqrt{2}}{2}+1.25 & -\frac{\pi}{4}\cos(-\frac{\pi}{4}) \\ -\frac{\pi}{4}\cos(-\frac{\pi}{4}) & -\frac{\sqrt{2}}{2}+1.25 \end{pmatrix} \not\succeq 0. \tag{A.180}$$

□

C_1 is not concave with respect to c. Likewise, C_2 is not concave with respect to r. Therefore, this game is not socially-convex.

A.8.3 c. Smooth, Convex, Monotone

Consider the following cost-minimization game:

$$C_1 = r^2 + c^2 \tag{A.181}$$

$$C_2 = r^2 + c^2. \tag{A.182}$$

This game is $(\frac{1}{2}, \frac{1}{2})$-smooth:

Proof.

$$\sum_{i=1}^{K} C_i(s_i^*, \mathbf{s}_{-i}) = r^{*2} + c^2 + r^2 + c^{*2} \tag{A.183}$$

$$\lambda \cdot C(\mathbf{s}^*) = r^{*2} + c^{*2} \tag{A.184}$$

$$\mu \cdot C(\mathbf{s}) = r^2 + c^2 \tag{A.185}$$

$$r^{*2} + c^2 + r^2 + c^{*2} \leq r^{*2} + c^2 + r^2 + c^{*2}. \tag{A.186}$$

□

Clearly, C_1 is convex in r and C_2 is convex in c, therefore the game is convex.

The corresponding map is monotone:

Proof.

$$F = \left(\begin{smallmatrix} 2r \\ 2c \end{smallmatrix}\right) \tag{A.187}$$

$$J = J_s = \left(\begin{smallmatrix} 2 & 0 \\ 0 & 2 \end{smallmatrix}\right) \succeq 0. \tag{A.188}$$

□

C_1 is not concave with respect to c. Likewise, C_2 is not concave with respect to r. Therefore, this game is not socially-convex.

A.8.4 d. Smooth, Convex, Socially-Convex

Consider the following cost-minimization game (inspired by modified Tail Drop policy in routing networks) over $(r, c) \in (0, 1]^2 = \mathcal{X}$:

$$C_1 = -\frac{1}{2}(\frac{r}{r+c}) \tag{A.189}$$

$$C_2 = -\frac{c}{r+c}. \tag{A.190}$$

This game is $(\frac{1}{2}, -1)$-smooth:

Proof.

$$\sum_{i=1}^{K} C_i(s_i^*, \mathbf{s}_{-i}) = -\frac{1}{2}(\frac{r^*}{r^*+c}) - \frac{c^*}{r+c^*} \tag{A.191}$$

$$\leq 0 \text{ over } \mathcal{X} \tag{A.192}$$

$$\mu \cdot C(\mathbf{s}) = 1 - \frac{1}{2}(\frac{r}{r+c}) \geq \frac{1}{2} \tag{A.193}$$

$$\lambda \cdot C(\mathbf{s}^*) = -\frac{1}{2}(1 - \frac{1}{2}(\frac{r^*}{r^*+c^*})) \geq -\frac{1}{2} \tag{A.194}$$

$$\sum_{i=1}^{K} C_i(s_i^*, \mathbf{s}_{-i}) \leq 0 \leq \mu \cdot C(\mathbf{s}^*) + \lambda \cdot C(\mathbf{s}^*). \tag{A.195}$$

□

C_1 is convex in r over $\mathcal{X}$ and C_2 is convex in c over $\mathcal{X}$, therefore the game is convex:

Proof.

$$\frac{\partial^2 C_1}{r^2} = \frac{c}{(r+c)^3} \geq 0 \text{ over } \mathcal{X} \tag{A.196}$$

$$\frac{\partial^2 C_2}{c^2} = \frac{2r}{(r+c)^3} \geq 0 \text{ over } \mathcal{X}. \tag{A.197}$$

□

The corresponding map is not monotone:

Proof.

$$F = -\frac{1}{(r+c)^2}\begin{pmatrix} \frac{c}{2} \\ r \end{pmatrix} \tag{A.198}$$

$$J = \frac{1}{(r+c)^3}\begin{pmatrix} c & \frac{c-r}{2} \\ r-c & 2r \end{pmatrix} \tag{A.199}$$

$$J_s = \frac{1}{(r+c)^3}\begin{pmatrix} c & \frac{r-c}{4} \\ \frac{r-c}{4} & 2r \end{pmatrix} \tag{A.200}$$

$$Det(J_s) = 2rc - \frac{1}{16}(r-c)^2 \tag{A.201}$$

$$Det(J_s)|_{r=0.01,c=1} = -0.041 \leq 0. \tag{A.202}$$

The determinant of J_s is negative over a subset of the domain (e.g., $r \leq 0.01, c = 1$), therefore, J_s is not positive semidefinite. Hence, F is not monotone. □

Let $\lambda_1 = \frac{2}{3}$ and $\lambda_2 = \frac{1}{3}$. Then $\lambda_1 C_1 + \lambda_2 C_2 = -\frac{1}{3}$, which is convex in (r, c). Also, C_1 is concave with respect to c and C_2 is concave with respect to r, therefore, this game is socially-convex:

Proof.

$$\frac{\partial^2 C_1}{\partial c^2} = -\frac{r}{(r+c)^3} \leq 0 \text{ over } \mathcal{X} \tag{A.203}$$

$$\frac{\partial^2 C_2}{\partial r^2} = -\frac{2c}{(r+c)^3} \leq 0 \text{ over } \mathcal{X}. \tag{A.204}$$

□

A.8.5 e. Smooth, Convex, Monotone, Socially-Convex

Consider the following cost-minimization game:

$$C_1 = r \tag{A.205}$$

$$C_2 = c. \tag{A.206}$$

This game is $(1, 0)$-smooth:

Proof.

$$\sum_{i=1}^{K} C_i(s_i^*, \mathbf{s}_{-i}) = r^* + c^* \tag{A.207}$$

$$\lambda \cdot C(\mathbf{s}^*) = r^* + c^* \tag{A.208}$$

$$r^* + c^* \leq r^* + c^*. \tag{A.209}$$

□

Clearly, C_1 is convex in r and C_2 is convex in c, therefore the game is convex.

The corresponding map is monotone:

Proof.

$$F = \left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right) \tag{A.210}$$

$$J = J_s = \left(\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right) \succeq 0. \tag{A.211}$$

□

Let $\lambda_1 = \lambda_2 = \frac{1}{2}$. Then $\lambda_1 C_1 + \lambda_2 C_2 = \frac{1}{2}(r + c)$, which is convex in (r, c). Also, C_1 is concave with respect to c and C_2 is concave with respect to r, therefore, this game is socially-convex.

A.8.6 f. Convex

Consider the following cost-minimization game:

$$C_1 = r^2 + \frac{r}{c^2 + \frac{1}{4}} - \frac{9}{5}c \tag{A.212}$$

$$C_2 = c^2 + \frac{c}{r^2 + \frac{1}{4}} - \frac{9}{5}r. \tag{A.213}$$

This game is not smooth:

Proof. Consider $(r, c) = (0, 0)$ and $(r^*, c^*) = (1, 1)$.

$$\sum_{i=1}^{K} C_i(s_i^*, \mathbf{s}_{-i}) = 10 \tag{A.214}$$

$$\mu \cdot C(\mathbf{s}) = 0 \tag{A.215}$$

$$\lambda \cdot C(\mathbf{s}^*) = 0 \tag{A.216}$$

$$10 \not\leqslant 0. \tag{A.217}$$

□

C_1 is convex in r and C_2 is convex in c, therefore the game is convex.

The corresponding map is not monotone:

Proof.

$$F = \begin{pmatrix} 2r + \frac{1}{c^2+\frac{1}{4}} \\ 2c + \frac{1}{r^2+\frac{1}{4}} \end{pmatrix} \tag{A.218}$$

$$J = \begin{pmatrix} 2 & -\frac{2c}{(c^2+\frac{1}{4})^2} \\ -\frac{2r}{(r^2+\frac{1}{4})^2} & 2 \end{pmatrix} \tag{A.219}$$

$$J_s = \begin{pmatrix} 2 & -\frac{c}{(c^2+\frac{1}{4})^2} - \frac{r}{(r^2+\frac{1}{4})^2} \\ -\frac{r}{(r^2+\frac{1}{4})^2} - \frac{c}{(c^2+\frac{1}{4})^2} & 2 \end{pmatrix} \tag{A.220}$$

$$J_s\Big|_{r=c=\frac{1}{4}} = \begin{pmatrix} 2 & -5.12 \\ -5.12 & 2 \end{pmatrix} \not\succeq 0. \tag{A.221}$$

□

This game is not socially-convex because C_1 is not concave with respect to c and likewise for C_2 and r. For example, $C_1(r = 1, c) = 1 + \frac{1}{c^2+\frac{1}{4}} - \frac{9}{5}c$ is not concave with respect to c.

A.8.7 g. Convex, Monotone

Consider the following cost-minimization game:

$$C_1 = r^2 + c^2 - 2 \tag{A.222}$$

$$C_2 = r^2 + c^2 + r + c - 2. \tag{A.223}$$

This game is not smooth:

Proof. Consider $(r, c) = (1, -1)$ and $(r^*, c^*) = (-1, 1)$.

$$\sum_{i=1}^{K} C_i(s_i^*, \mathbf{s}_{-i}) = 2 \tag{A.224}$$

$$\mu \cdot C(\mathbf{s}) = 0 \tag{A.225}$$

$$\lambda \cdot C(\mathbf{s}^*) = 0 \tag{A.226}$$

$$2 \not\leq 0. \tag{A.227}$$

□

C_1 is convex in r and C_2 is convex in c, therefore the game is convex.

The corresponding map is monotone:

Proof.

$$F = \begin{pmatrix} 2r \\ 2c+1 \end{pmatrix} \tag{A.228}$$

$$J = J_s = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \succeq 0. \tag{A.229}$$

□

This game is not socially-convex because C_1 is not concave with respect to c and likewise for C_2 and r.

A.8.8 h. Convex, Socially-Convex

Consider the following cost-minimization game (inspired by modified Tail Drop policy in routing networks) over $(r, c) \in (0, 1]^2 = \mathcal{X}$:

$$C_1 = -\frac{1}{2}\left(\frac{r}{r+c}\right) + \frac{3}{4} \tag{A.230}$$

$$C_2 = -\frac{c}{r+c}. \tag{A.231}$$

This game is not smooth:

Proof. Consider $(r, c) = (1, 1)$ and $(r^*, c^*) = (\frac{1}{2}, \frac{1}{2})$.

$$\sum_{i=1}^{K} C_i(s_i^*, \mathbf{s}_{-i}) = \frac{1}{4} \tag{A.232}$$

$$\mu \cdot C(\mathbf{s}^*) = 0 \tag{A.233}$$

$$\lambda \cdot C(\mathbf{s}^*) = 0 \tag{A.234}$$

$$\frac{1}{4} \not\leq 0. \tag{A.235}$$

□

C_1 is convex in r over $\mathcal{X}$ and C_2 is convex in c over $\mathcal{X}$, therefore the game is convex:

Proof.

$$\frac{\partial^2 C_1}{r^2} = \frac{c}{(r+c)^3} \geq 0 \text{ over } \mathcal{X} \tag{A.236}$$

$$\frac{\partial^2 C_2}{c^2} = \frac{2r}{(r+c)^3} \geq 0 \text{ over } \mathcal{X}. \tag{A.237}$$

□

The corresponding map is not monotone:

Proof.

$$F = -\frac{1}{(r+c)^2} \begin{pmatrix} \frac{c}{2} \\ r \end{pmatrix} \tag{A.238}$$

$$J = \frac{1}{(r+c)^3} \begin{pmatrix} c & \frac{c-r}{2} \\ r-c & 2r \end{pmatrix} \tag{A.239}$$

$$Js = \frac{1}{(r+c)^3} \begin{pmatrix} c & \frac{r-c}{4} \\ \frac{r-c}{4} & 2r \end{pmatrix} \tag{A.240}$$

$$Det(J_s) = 2rc - \frac{1}{16}(r-c)^2 \tag{A.241}$$

$$Det(J_s)|_{r=0.01, c=1} = -0.041 \leq 0. \tag{A.242}$$

The determinant of J_s is negative over a subset of the domain (e.g., $r \leq 0.01, c = 1$), therefore, J_s is not positive semidefinite. Hence, F is not monotone. □

Let $\lambda_1 = \frac{2}{3}$ and $\lambda_2 = \frac{1}{3}$. Then $\lambda_1 C_1 + \lambda_2 C_2 = \frac{1}{6}$, which is convex in (r, c). Also, C_1 is concave with respect to c and C_2 is concave with respect to r, therefore, this game is socially-convex:

Proof.

$$\frac{\partial^2 C_1}{\partial c^2} = -\frac{r}{(r+c)^3} \leq 0 \text{ over } \mathcal{X} \tag{A.243}$$

$$\frac{\partial^2 C_2}{\partial r^2} = -\frac{2c}{(r+c)^3} \leq 0 \text{ over } \mathcal{X}. \tag{A.244}$$

□

A.8.9 i. Convex, Monotone, Socially-Convex

Consider the following cost-minimization game:

$$C_1 = r^2 - 1 \tag{A.245}$$

$$C_2 = c^2 + r + c - 1. \tag{A.246}$$

This game is not smooth:

Proof. Consider $(r, c) = (1, -1)$ and $(r^*, c^*) = (-1, 1)$.

$$\sum_{i=1}^{K} C_i(s_i^*, \mathbf{s}_{-i}) = 2 \tag{A.247}$$

$$\mu \cdot C(\mathbf{s}) = 0 \tag{A.248}$$

$$\lambda \cdot C(\mathbf{s}^*) = 0 \tag{A.249}$$

$$2 \not\leq 0. \tag{A.250}$$

□

C_1 is convex in r and C_2 is convex in c, therefore the game is convex.

The corresponding map is monotone:

Proof.

$$F = \begin{pmatrix} 2r \\ 2c+1 \end{pmatrix} \tag{A.251}$$

$$J = J_s = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \succeq 0. \tag{A.252}$$

□

Let $\lambda_1 = \lambda_2 = \frac{1}{2}$. Then $\lambda_1 C_1 + \lambda_2 C_2 = \frac{1}{2}(r^2 + c^2 + r + c - 2)$, which is convex in (r, c). Also, C_1 is concave with respect to c and C_2 is concave with respect to r, therefore, this game is socially-convex.

A.9 Concave Games

Several well known concave (utility) games are (convex loss) monotone. We test the following games for monotonicity and interpret the path integral loss over their fields.

A.9.1 Linear Cournot Competition

In linear *Cournot* competition, N firms compete for customers by adjusting the quantity of goods they produce, x_i. Firms pay a cost for producing those goods, $c_i(x_i)$, which is assumed to be a convex function in x_i. The prices for goods are set by the consumer demand markets according to a price function, $p(x) = a - b\sum_k x_k$, with $a, b > 0$. The firms attempt to maximize their utility or profit functions, $u_i(x) = x_i p(x) - c_i(x_i)$. Here, we show that the map associated with the game, $F(x) = \{-\frac{\partial u_0}{\partial x_0}, \ldots, -\frac{\partial u_N}{\partial x_N}\}$, is monotone.

First we derive the first and second partial derivatives:

$$\frac{\partial u_i}{\partial x_i} = p(x) - bx_i - \frac{\partial c_i}{\partial x_i} \tag{A.253}$$

$$\frac{\partial^2 u_i}{\partial x_i^2} = -2b - \frac{\partial^2 c_i}{\partial x_i^2} \tag{A.254}$$

$$\frac{\partial^2 u_i}{\partial x_i x_j} = -b. \tag{A.255}$$

These derivatives, in turn, define the Jacobian, $Jac(F)$, which can be decomposed into a constant matrix with all entires equal to b and a diagonal matrix consisting of $b + \frac{\partial^2 c_i}{\partial x_i^2}$. A constant matrix with positive entries b is rank-1 with eigenvalues $\{Nb\} + \{0\}^{N-1}$. The cost functions, c_i, are assumed to be convex, therefore, the diagonal matrix is positive-definite. This implies that the sum of the two *symmetric* matrices is positive definite. It follows that F is monotone.

Let $v(t) = o + (x - o)t$ and $dv = (x - o)dt$. Then the path integral over $-F$ is

$$\int_{v:o\to x}\langle -F(v), dv\rangle = \int_0^1 \sum_i (a - bv_i(t) - b\sum_k v_k(t))(x_i - o_i)dt - \sum_i (c_i(x) - c_i(o)) \tag{A.256}$$

$$= \sum_i (x_i - o_i)\int_0^1 (a - bv_i(t) - b\sum_k v_k(t))dt - \sum_i c_i(x) \tag{A.257}$$

$$= \sum_i (x_i - o_i)\int_0^1 (a - bo_i - b(x_i - o_i)t - b\sum_k o_k + (x_k - o_k)t)dt$$

$$- \sum_i (c_i(x) - c_i(o)) \tag{A.258}$$

$$= \sum_i (x_i - o_i)(at - bo_i t - b(x_i - o_i)\frac{t^2}{2} - b\sum_k o_k t + (x_k - o_k)\frac{t^2}{2})\Big|_0^1$$

$$- \sum_i (c_i(x) - c_i(o)) \tag{A.259}$$

$$= \sum_i (x_i - o_i)(a - bo_i - \frac{b}{2}(x_i - o_i) - b\sum_k o_k + \frac{1}{2}(x_k - o_k))$$

$$- \sum_i (c_i(x) - c_i(o)) \tag{A.260}$$

$$= \sum_i (x_i - o_i)(a - b\frac{o_i + x_i}{2} - b\sum_k \frac{o_k + x_k}{2}) - \sum_i (c_i(x) - c_i(o)) \tag{A.261}$$

$$= \sum_i x_i p(z_i) - c_i(x) - \sum_i o_i p(z_i) - c_i(o) \tag{A.262}$$

where $z_i = \frac{1}{2}(o_i + x_i + \sum_k o_k + x_k)$.

So *auto*-welfare is calculating profits with player specific prices. Specifically, each player's price is set as a deviation from the average supply of o and x. If o is set to the origin, z_i is half the total market supply except with player i's supply at full. More generally, if player i chooses to flood the market with good, x_i, *auto*-welfare computes its contribution to the sum with a lower price point.

A.9.2 Linear Resource Allocation

In a resource allocation game, N users share a communication channel with finite capacity (e.g., $C = 1$). Each user i submits a bid, $x_i \in [\epsilon > 0, 1]$, to the communication network which then allocates a fraction of the communication channel to each user according an allocation function, $M_i(x) = x_i / \sum_k x_k$. Each user plays to maximize its utility, $u_i(x) = \psi_i(M_i(x)) - \alpha_i x_i$, with $\alpha_i > 0$. Here, we consider a simplified value function, $\psi_i(z) = \beta z$, with $\beta > 0$ and show that the map associated with the game, $F(x) = \{-\frac{\partial u_0}{\partial x_0}, \ldots, -\frac{\partial u_N}{\partial x_N}\}$, is monotone.

First we derive the first and second partial derivatives:

$$\frac{\partial M_i(x)}{\partial x_i} = \frac{1}{\sum_k x_k}\Big[1 - \frac{x_i}{\sum_k x_k}\Big] \tag{A.263}$$

$$\frac{\partial M_i(x)}{\partial x_j} = \frac{1}{\sum_k x_k}\Big[0 - \frac{x_i}{\sum_k x_k}\Big] \tag{A.264}$$

$$\frac{\partial^2 M_i(x)}{\partial x_i^2} = -\frac{1}{(\sum_k x_k)^2}\Big[2 - \frac{2x_i}{\sum_k x_k}\Big] \tag{A.265}$$

$$\frac{\partial^2 M_i(x)}{\partial x_i x_j} = -\frac{1}{(\sum_k x_k)^2}\Big[1 - \frac{2x_i}{\sum_k x_k}\Big] \tag{A.266}$$

$$\frac{\partial^2 u_i}{\partial x_i^2} = \beta \frac{\partial^2 M_i(x)}{\partial x_i^2} \tag{A.267}$$

$$\frac{\partial^2 u_i}{\partial x_i x_j} = \beta \frac{\partial^2 M_i(x)}{\partial x_i x_j}. \tag{A.268}$$

These derivatives, in turn, define the Jacobian, $Jac(F)$, which can be decomposed into a rank-1 matrix, M, with constant rows and an identity matrix, $\mathbb{I}_N$. Let $z_i = x_i / \sum_k x_k \in (0, 1]$:

$$Jac(F)_{ij} = \underbrace{\frac{\beta}{(\sum_k x_k)^2}}_{\geq 0}\Big[\underbrace{1 - 2z_i}_{M_{ij}} + \underbrace{\mathbb{I}(i = j)}_{\mathbb{I}_N}\Big]. \tag{A.269}$$

We can prove $Jac(F)$ is monotone by showing $\frac{1}{2}(Jac(F) + Jac(F)^\top) \succeq 0$. As a first step, we'll lower bound the eigenvalues of a symmetrized M:

$$M_{ij}^{(s)} = \frac{1}{2}(M + M^\top)_{ij} \tag{A.270}$$

$$= 1 - (z_i + z_j) \text{ is at most rank-2} \tag{A.271}$$

$$\implies \lambda(M^{(s)}) = \{\lambda_{lo}, \lambda_{hi}\} + \{0\}^{N-2} \tag{A.272}$$

$$\text{Tr}(M^{(s)}) = \lambda_{lo} + \lambda_{hi} = N - 2 \tag{A.273}$$

$$||M^{(s)}||_1 = ||M^{(s)}||_\infty = |1 - 2z_i| + \sum_{j \neq i} |1 - (z_i + z_j)| \tag{A.274}$$

$$= |1 - 2z_i| + \sum_{j \neq i} 1 - (z_i + z_j) \tag{A.275}$$

$$\text{and } z_i, z_j > \epsilon, \sum_k z_k = 1 \implies z_i + z_j < 1$$

$$= |1 - 2z_i| + (N-2)(1 - z_i) \leq N - 1 \tag{A.276}$$

$$\rho(M^{(s)}) = ||M^{(s)}||_2 \leq \sqrt{||M^{(s)}||_1 ||M^{(s)}||_\infty} = N - 1 \qquad \textit{Holder's inequality} \tag{A.277}$$

$$\text{Assume } \min \lambda(M^{(s)}) = \lambda_{lo} < -1 \implies \max \lambda(M^{(s)}) = \lambda_{hi} > N - 1 \text{ contradicts } \rho(M^{(s)}) \tag{A.278}$$

$$\implies \min \lambda(M^{(s)}) \geq -1. \tag{A.279}$$

The eigenvalues of $M^{(s)}$ are lower bounded by -1, therefore, the eigenvalues of the sum of $M^{(s)}$ and an identity matrix are lower bounded by 0:

$$J^{(s)} = \frac{1}{2}(Jac(F) + Jac(F)^2) = \underbrace{\frac{\beta}{(\sum_k x_k)^2}}_{\geq 0} \left[M^{(s)} + \mathbb{1}_N\right] \tag{A.280}$$

$$\implies \min \lambda(J^{(s)}) \geq 0 \tag{A.281}$$

$$\implies F \text{ is monotone } \surd. \tag{A.282}$$

Next, we'll compare welfare and *auto*-welfare. For the moment, consider welfare with $\alpha_i = 0$:

$$W = \sum_i u_i = \beta \sum_i \frac{x_i}{\sum_k x_k} = \beta. \tag{A.283}$$

Notice that without α_i, welfare regret, $\sum_t W(x_t) - W(x^*)$, is identically zero and is a pointless quantity to maximize. If we include α_i,

$$W = \sum_i u_i = \beta - \sum_i \alpha_i x_i, \tag{A.284}$$

whose maximizer has a simple closed-form solution for any $\alpha_i > 0$: $x_i = \epsilon$. This result is independent of the parameters of the utility functions, β and α_i. Next, we'll compute *auto*-welfare to contrast. Let $v(t) = o + (x - o)t$ and $dv = (x - o)dt$. The critical component of *auto*-welfare is the path integral. Then

$$\int_{v:o\to x} \langle -F(v), dv\rangle = \int_0^1 \sum_i \Big(\frac{\beta}{\sum_k v_k(t)}\Big[1 - \frac{v_i(t)}{\sum_k v_k(t)}\Big] - \alpha_i\Big)(x_i - o_i)dt \tag{A.285}$$

$$= \sum_i \int_0^1 \frac{\beta}{\sum_k v_k(t)}\Big[1 - \frac{v_i(t)}{\sum_k v_k(t)}\Big](x_i - o_i)dt - \sum_i \int_0^1 \alpha_i(x_i - o_i)dt \tag{A.286}$$

$$= \sum_i \int_0^1 \frac{\beta(x_i - o_i)}{\sum_k o_k + (x_k - o_k)t}\Big[1 - \frac{o_i + (x_i - o_i)t}{\sum_k o_k + (x_k - o_k)t}\Big]dt - \sum_i \alpha_i(x_i - o_i) \tag{A.287}$$

$$= \sum_i \int_0^1 \frac{\beta(x_i - o_i)}{\sum_k o_k + t\sum_k (x_k - o_k)}\Big[1 - \frac{o_i + t(x_i - o_i)}{\sum_k o_k + t\sum_k (x_k - o_k)}\Big]dt$$
$$- \sum_i \alpha_i(x_i - o_i) \tag{A.288}$$

$$= \sum_i \beta(x_i - o_i) \int_0^1 \frac{1}{s_o + t(s_x - s_o)}\Big[1 - \frac{o_i + t(x_i - o_i)}{s_o + t(s_x - s_o)}\Big]dt - \sum_i \alpha_i(x_i - o_i). \tag{A.289}$$

Breaking apart the left integrand, we first compute the following:

$$\int_0^1 \frac{1}{s_o + t(s_x - s_o)} dt = \frac{\ln(s_o + t(s_x - s_o))}{s_x - s_o}\Big|_0^1 = \frac{\ln(s_x/s_o)}{s_x - s_o}. \tag{A.290}$$

We'll use integration by parts on the other part. Let $u = o_i + t(x_i - o_i)$, $dv = (s_o + t(s_x - s_o))^{-2}dt$, $v = -\frac{1}{s_x - s_o}(s_o + t(s_x - s_o))^{-1}$, and $du = x_i - o_i dt$. Then

$$\int_0^1 \frac{o_i + t(x_i - o_i)}{(s_o + t(s_x - s_o))^2} dt = \int_0^1 u dv = uv\Big|_0^1 - \int_0^1 v du \tag{A.291}$$

$$uv\Big|_0^1 = -\frac{o_i + t(x_i - o_i)}{(s_x - s_o)(s_o + t(s_x - s_o))}\Big|_0^1 \tag{A.292}$$

$$= \frac{o_i}{s_o(s_x - s_o)} - \frac{x_i}{s_x(s_x - s_o))} \tag{A.293}$$

$$-\int_0^1 v du = \int_0^1 \frac{1}{s_x - s_o}(s_o + t(s_x - s_o))^{-1}(x_i - o_i)dt \tag{A.294}$$

$$= \frac{x_i - o_i}{s_x - s_o}\int_0^1 \frac{1}{s_o + t(s_x - s_o)} dt \tag{A.295}$$

$$= \frac{x_i - o_i}{s_x - s_o}\frac{\ln(s_x/s_o)}{s_x - s_o}. \tag{A.296}$$

Combining the two results, we find

$$\int_{v:o\to x} \langle -F(v), dv\rangle = \sum_i \beta(x_i - o_i)\Big[\frac{\ln(s_x/s_o)}{s_x - s_o}(1 - \frac{x_i - o_i}{s_x - s_o}) + \frac{x_i}{s_x(s_x - s_o)} - \frac{o_i}{s_o(s_x - s_o)}\Big]$$
$$- \sum_i \alpha_i(x_i - o_i) \tag{A.297}$$

$$= \sum_i \beta\frac{x_i - o_i}{s_x - s_o}\Big[\ln(s_x/s_o)(1 - \frac{x_i - o_i}{s_x - s_o}) + \frac{x_i}{s_x} - \frac{o_i}{s_o}\Big] - \sum_i \alpha_i(x_i - o_i) \tag{A.298}$$

$$= \beta\ln(s_x/s_o)\sum_i \frac{x_i - o_i}{s_x - s_o} - \beta\ln(s_x/s_o)\sum_i (\frac{x_i - o_i}{s_x - s_o})^2$$
$$+ \beta\sum_i \frac{x_i - o_i}{s_x - s_o}\Big[\frac{x_i}{s_x} - \frac{o_i}{s_o}\Big] - \sum_i \alpha_i(x_i - o_i) \tag{A.299}$$

$$= \beta\ln(s_x/s_o)\Big(1 - \frac{||x - o||_2^2}{(s_x - s_o)^2}\Big) + \beta\sum_i \frac{x_i - o_i}{s_x - s_o}\Big[\frac{x_i}{s_x} - \frac{o_i}{s_o}\Big] - \sum_i \alpha_i(x_i - o_i). \tag{A.300}$$

Let $W_o^a = 0$ and $o = x^*$, then

$$W^a(x) = \beta \ln(s_x/s_{x^*})\Big(1 - \frac{||x - x^*||_2^2}{(s_x - s_{x^*})^2}\Big) + \beta \sum_i \frac{x_i - x_i^*}{s_x - s_{x^*}}\Big[\frac{x_i}{s_x} - \frac{x_i^*}{s_{x^*}}\Big] - \sum_i \alpha_i (x_i - x_i^*). \tag{A.301}$$

Finding a closed-form solution for a global optimizer of Equation (A.301) seems daunting not to mention the fact that the optimizer, x^*, appears in the optimization function. Fortunately, Theorem 4 states that a global optimizer of Equation (A.301) is also a solution the corresponding VI($F, \mathcal{X} = [\epsilon, 1]^N$). If we assume $x = x^*$ lies in the interior of $\mathcal{X}$, then $F(x^*) = 0$:

$$F(x) = \beta \frac{1}{\sum_k x_k}(1 - \frac{x_i}{\sum_k x_k}) - \alpha_i = 0 \implies x_i = s_x(1 - \alpha_i \frac{s_x}{\beta}) \tag{A.302}$$

$$s_x = \sum_i x_i = s_x \sum_i (1 - \alpha_i \frac{s_x}{\beta}) \implies s_x\Big[\frac{\sum_i \alpha_i}{\beta} s_x + (1 - N)\Big] = 0 \tag{A.303}$$

$$\implies s_x = \emptyset \text{ or } \frac{\beta(N-1)}{\sum_i \alpha_i}. \tag{A.304}$$

Combining the two results, we find that the global optimizer is

$$x_i^* = \frac{\beta(N-1)}{\sum_k \alpha_k}\Big(1 - \frac{\alpha_i}{\sum_k \alpha_k}(N-1)\Big). \tag{A.305}$$

As the cost coefficients, α_i, grow relative to the revenue coefficients, β, the optimal bid size drops. Also, as the number of users, N, increases, the optimal bid size increases, albeit with diminishing returns. Hence, *auto*-welfare has a rich dependence on the utility parameters and number of users, which is very different from the complete independence given by welfare.

The linear lower bound on *auto*-welfare regret is given by

$$regret_{W^a} = -regret_1 \geq \langle -F(x), x - x^* \rangle \tag{A.306}$$

$$= \sum_i \left[\beta\gamma_i \frac{x_i}{\sum_k x_k} - \alpha_i x_i \right] - \sum_i \left[\beta\gamma_i \frac{x_i^*}{\sum_k x_k} - \alpha_i x_i^* \right] \tag{A.307}$$

$$\text{where } \gamma_i = (1 - \frac{x_i}{\sum_k x_k}) \tag{A.308}$$

$$= \sum_i u_i^\gamma(x) - \sum_i u_i^\gamma(x^*) \tag{A.309}$$

where u_i^γ is the original utility function, u_i, with "revenues" ($\psi_i(M_i(x))$) reweighted by γ_i. In this case, *auto*-welfare regret is actually computing standard welfare with reweighted "revenues". High revenues are weighted lower and low revenues are weighted higher, which naturally encourages a more even distribution of resources.

A.9.3 Congestion Control Protocols

Similar utility functions can be used to model a congestion control protocol with a *tail-drop* policy. In this game, a router drops packets if the total number of packets exceeds the network capacity (e.g., $C = 1$). In this case, the utility functions are defined piecewise:

$$u_i(x) = \begin{cases} x_i & \sum_k x_k \leq 1 \\ \beta \frac{x_i}{\sum_k x_k} - (\beta - 1)x_i & \sum_k x_k > 1. \end{cases} \tag{A.310}$$

For $\sum_k x_k \leq 1$, the utility functions are linear, thus the Jacobian of the associated game map, $F(x) = \{-\frac{\partial u_0}{\partial x_0}, \ldots, -\frac{\partial u_N}{\partial x_N}\}$, is a zero matrix, which is positive semidefinite. Monotonicity for the second case follows the same proof as for the linear resource allocation game above.

A.10 Machine Learning Network Motivation

The example presented in Subsection 2.7.2 demonstrates our proposed no-regret algorithm on a cloud-based machine learning network. Our network is motivated by expectations of the next era of machine learning. Data is often the difference between a high performing model and a mediocre one; for some data hungry models (e.g., deep learning), *Big Data* launches them to state-of-the-art results. We expect *Big Data* to drive a mature digital supply chain capable of supporting an economy where producers provide data for consumers (i.e., machine learning models) to consume. Unlike the present, this commodity will not be transferred into local storage for consumption on personal machines; rather, it will be transmitted in batches, immediately consumed for training, and discarded to allow room for the next batch. Our model of a cloud-based machine learning network (MLN) is trivially adapted from the service oriented internet (SOI) model proposed in the work of Nagurney and Wolf [2014]. In the original SOI model, service providers (e.g., Netflix, Amazon) stream content (e.g., movies, music). In our MLN model, service providers (e.g., Twitter, Wikipedia) stream machine learning data. Service providers control the quantity of data (i.e., # of samples $\times$ # of features) flowing through the market. Network providers charge service providers a fee for transmitting their data to consumers. The price different consumer markets are willing to pay service providers to stream data over a network of a certain quality is given by demand functions, price(quantity,quality). Given these relationships, service providers and network providers attempt to maximize their profits by varying their respective controls (quantity,quality) over the network. These relationships are parameterized so that we can instantiate ten five-firm networks by drawing parameters from uniform distributions over predefined ranges (code available @ github.com/all-umass/VI-Solver).

A.11 GTD Algorithms

The gradient temporal difference learning algorithm (GTD) is a Reinforcement Learning algorithm that can be used to evaluate a target policy by observing a separate behavior policy. The GTD update rules are

$$y_{t+1} = y_t + \alpha_t(b - A\theta_t - My_t) \tag{A.311}$$

$$\theta_{t+1} = \theta_t + \alpha_t(A^\top y_t) \tag{A.312}$$

where $M = \mathbf{I}$ or M is a covariance matrix. Either way, M is symmetric positive definite.

These update rules can be derived from a two-player game with appropriate agent loss functions, $f^{(i)}$. For example, if

$$f^{(1)}(y, \theta) = -y^\top(b - A\theta) + \frac{1}{2}y^\top My \tag{A.313}$$

$$f^{(2)}(y, \theta) = -\theta^\top A^\top y, \tag{A.314}$$

then online simultaneous gradient descent with $F = [\nabla_y f^{(1)}, \nabla_\theta f^{(2)}]$ gives the same updates as GTD. Consider rewriting this update as $[y_{t+1}, \theta_{t+1}] = [y_t, \theta_t] - \alpha F([y_t, \theta_t])$. Then

$$F = \begin{pmatrix} My + A\theta - b \\ -A^\top y \end{pmatrix} \tag{A.315}$$

$$= \begin{pmatrix} M & A \\ -A^\top & 0 \end{pmatrix} \begin{pmatrix} y \\ \theta \end{pmatrix} + \begin{pmatrix} -b \\ 0 \end{pmatrix} \tag{A.316}$$

$$= Jx + d \tag{A.317}$$

$$J_s = \begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix} \succeq 0. \tag{A.318}$$

Therefore, F is monotone.

By Theorem 2, the corresponding path integral loss that these dynamics bound is

$$f(x) = \frac{1}{2}[x^\top J^\top x + x^\top (J - J^\top)x^* - x^{*\top} J^\top x^*] + d^\top (x - x^*) \tag{A.319}$$

$$= \frac{1}{2}(y^\top My - y^{*\top} My^*) + y^\top A\theta^* - y^{*\top} A\theta - b^\top (y - y^*) \tag{A.320}$$

$$= \frac{1}{2}||y||_M^2, \tag{A.321}$$

where $y^* = 0$ and $A\theta^* - b = 0$. A measure of how well θ is performing is missing from the loss. This confirms our theoretical motivation for using the modified path integral loss displayed below:

$$f(x) = \frac{1}{2}[\hat{x}^\top J^\top \hat{x} + \hat{x}^\top (J - J^\top)x^* - x^{*\top} J^\top x^*] + d^\top (\hat{x} - x^*) \tag{A.322}$$

$$+ \frac{1}{2}[x^\top J^\top x + x^\top (J - J^\top)\hat{x} - \hat{x}^\top J^\top \hat{x}] + d^\top (x - \hat{x}) \tag{A.323}$$

$$= ||y||^2_{\frac{1}{2}M + \hat{\eta} AA^\top} + ||b - A\theta||^2_{\hat{\eta} I} - \langle y, b - A\theta \rangle_{\hat{\eta} M} \tag{A.324}$$

where $||v||_A^2 = z^\top Az$ and $\langle u, v \rangle_A = u^\top Av$. Note that this modified loss contains terms that encourage $A\theta = b$, $y = 0$, and y to align with $A\theta - b$. In the GTD algorithms, y was originally introduced as an auxiliary variable to estimate $\mathbb{E}[\rho\delta\phi] = b - A\theta$, so it is reassuring that the modified loss contains these terms.

A.12 Constant-Linear GANs

The Constant-Linear GAN is a Wasserstein-type GAN with constant generator and linear discriminator. The minimax objective is

$$\min_G \max_d V(G, d) = \mathbb{E}_{x \sim p_{\text{data}(x)}}[d^\top x] - \mathbb{E}_{z \sim p_z(z)}[d^\top Gz] \tag{A.325}$$

where $x \in R^n$, $z \in R^m$, $d \in R^n$, $G \in R^{n \times m}$.

For simplicity of exhibition, we will estimate the expectations with a single sample $x \sim p_{\text{data}}(x)$ and $z \sim p_z(z)$, however, the result applies to batches as well. The minimax objective simplifies to

$$\min_G \max_d V(G, d) = d^\top (x - Gz) \tag{A.326}$$

$$= \sum_i d_i (x_i - \sum_j G_{ij} z_j). \tag{A.327}$$

Let $g = \begin{pmatrix} G_{11} \\ \vdots \\ G_{1m} \\ G_{i1} \\ \vdots \\ G_{im} \\ \vdots \end{pmatrix}$ be a flattened version of the matrix G.

Also, let $A = \begin{pmatrix} z_1 & 0 & \cdots \\ z_j & 0 & \cdots \\ z_m & 0 & \cdots \\ 0 & z_1 & \cdots \\ 0 & z_j & \cdots \\ 0 & z_m & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} \in \mathbb{R}^{mn \times n}$.

Then,

$$F = \begin{pmatrix} -d_1 z_1 \\ -d_1 z_j \\ -d_1 z_m \\ \vdots \\ -d_i z_j \\ \sum_j G_{1j} z_j - x_1 \\ \sum_j G_{ij} z_j - x_i \\ \sum_j G_{nj} z_j - x_n \end{pmatrix} \tag{A.328}$$

$$= \begin{pmatrix} 0^{mn \times mn} & -A \\ A^\top & 0^{n \times n} \end{pmatrix} \begin{pmatrix} g \\ d \end{pmatrix} + \begin{pmatrix} 0 \\ -x \end{pmatrix} \tag{A.329}$$

$$= J\gamma + b \tag{A.330}$$

$$J_s = \left(\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right) \succeq 0. \tag{A.331}$$

Therefore, F is monotone. If G and d are regularized with $\alpha||\cdot||_2^2$, then F is strongly monotone with parameter α.

In the subsequent section (Appendix A.13), we show that the corresponding path integral loss that OEG bounds is

$$V(\hat{G}, d^*) - V(G^*, \hat{d}) + V(G, \hat{d}) - V(\hat{G}, d). \tag{A.332}$$

The minimax objective is linear in the random variables x and z, which allows us to move weights outside the expectations. Let $b = G_{[:,-1]}$ denote the last column and $G_{[:,:-1]}$ the preceding of the columns. We'll assume $p(z)$ has mean zero and x has mean μ. The objective simplifies to

$$\min_G \max_{d,||d||\leq 1} V(G,d) = d^\top(\mathbb{E}[x] - G_{[:,:-1]}\mathbb{E}[z] - G_{[:,-1]}) \tag{A.333}$$

$$\min_b \max_{d,||d||\leq 1} V(b,d) = d^\top(\mu - b) \tag{A.334}$$

$$\min_b V(b) = \frac{(\mu-b)^\top(\mu-b)}{||\mu-b||} = 0 \tag{A.335}$$

$$\Rightarrow b = \mu, d = 0. \tag{A.336}$$

Therefore, affine GANs only learn the mean of the distribution.

A.13 Path Integral Loss for Minimax Games

Here we evaluate the path integral loss for minimax games:

$$\min_{x_1}\max_{x_2} V(x_1,x_2) \tag{A.337}$$

$$dV = \langle\frac{\partial V}{\partial x_1}, dx_1\rangle + \langle\frac{\partial V}{\partial x_2}, dx_2\rangle \tag{A.338}$$

$$f(x) = \int_{z:o\to x}\langle F(z), dz\rangle \tag{A.339}$$

$$= \int_{z:o\to x}\langle\frac{\partial V}{\partial z_1}, dz_1\rangle + \langle -\frac{\partial V}{\partial z_2}, dz_2\rangle \tag{A.340}$$

$$= \int_{\substack{z_1:o_1\to x_1\\ z_2:o_2\to o_2}} dV - \int_{\substack{z_1:o_1\to o_1\\ z_2:o_2\to x_2}} dV \tag{A.341}$$

$$= V(x_1,o_2) - V(o_1,o_2) - V(o_1,x_2) + V(o_1,o_2) \tag{A.342}$$

$$= V(x_1,o_2) - V(o_1,x_2). \tag{A.343}$$

Similarly, the modified path integral loss is

$$f(x) = V(\hat{x}_1,o_2) - V(o_1,\hat{x}_2) + V(x_1,\hat{x}_2) - V(\hat{x}_1,x_2). \tag{A.344}$$

A.14 Composition of Monotone Fields

Let F and G be monotone maps from $\mathbb{R}^n \to \mathbb{R}^n$. Consider the composition $F \circ G$. The composition of monotone fields is not necessarily monotone. Take $F = G = Ix$,

where I is the identity matrix. Then $F \circ G = Ix^2$ where the power is applied elementwise. This field represents the gradient of $\frac{1}{3}x^3$, which is a non-convex function, therefore $F \circ G$ is non-monotone.

APPENDIX B

LINEAR QUADRATIC GANS AND CROSSING-THE-CURL

This appendix supplements Chapter 3, but also provides additional material for the curious reader.

B.1 A Survey of Candidate Theories Continued

In this section, we survey several alternative theories for studying GANs. Ultimately, we select Variational Inequalities based on our research.

B.1.1 Algorithmic Game Theory

Algorithmic Game Theory (AGT) offers results on convergence to equilibria when a game, possibly online, is convex [41], socially-convex [33], or smooth [97]. A convex game is one in which all player losses are convex in their respective variables, i.e. $f_i(x_i, x_{-i})$ is convex in x_i. A socially-convex game adds the additional requirements that 1) there exists a strict convex combination of the player losses that is convex and 2) each player's loss is concave in the variables of each of the other players. In other words, the players as a whole are cooperative, yet individually competitive. Lastly, smoothness ensures that "the externality imposed on any one player by the actions of the others is bounded" [97]. In a zero-sum game such as Equation (3.1), one player's gain is exactly the other player's loss making smoothness an unlikely fit for studying GANs.

B.1.2 Differential Games

Differential games [13, 38] consider more general dynamics such as $\ddot{x} = -F(x)$, not just first order ODEs, however, the focus is on systems that separate control, u, and state x, i.e. $\dot{x} = -F(x(t), u(t), t)$. More specific to our interests, Differential Nash Games can be expressed as Differential VIs, a specific class of infinite dimensional VIs with explicit state dynamics and explicit controls; these, in turn, can be framed as infinite dimensional VIs without an explicit state.

B.1.3 Equivalence of Monotonicity to Euclidean Contraction

Strongly-monotone maps are equivalent to contraction operators with respect to Euclidean distance. Consider the following iterative update:

$$x_{k+1} = G(x_k). \tag{B.1}$$

Assume the operator G is known to be a $(1-\gamma)$-contraction ($\gamma \in (0,1)$) with respect to the distance function, D:

$$D(G(x); G(y)) \leq (1-\gamma)D(x; y) \implies D(G(x); G(y))D(x; y) \leq (1-\gamma)D(x; y)^2. \tag{B.2}$$

If D is Euclidean distance, then

$$||G(x) - G(y)||||x - y|| \leq (1-\gamma)||x - y||^2. \tag{B.3}$$

Rewrite the update to better fit the form of variational inequality updates:

$$x_{k+1} = x_k - \eta\Big(\frac{x_k - G(x_k)}{\eta}\Big). \tag{B.4}$$

where $F(x) = \frac{x - G(x)}{\eta}$.

If $G(x)$ is a $(1-\gamma)$-contraction. Then

$$\langle F(x)-F(y),x-y\rangle = \frac{1}{\eta}||x-y||^2 - \frac{1}{\eta}\langle G(x)-G(y),x-y\rangle \tag{B.5}$$

$$\geq \frac{1}{\eta}(||x-y||^2 - ||G(x)-G(y)||||x-y||) \text{ by Cauchy-Schwarz} \tag{B.6}$$

$$\geq \frac{1}{\eta}(||x-y||^2 - (1-\gamma)||x-y||^2) \tag{B.7}$$

$$= \frac{\gamma}{\eta}||x-y||^2 > 0. \tag{B.8}$$

Therefore, F is strongly monotone with parameter $\frac{\gamma}{\eta}$.

In addition,

$$||F(x)-F(y)|| = ||\frac{x-G(x)}{\eta} - \frac{y-G(y)}{\eta}|| \tag{B.9}$$

$$\leq \frac{1}{\eta}\Big[||x-y|| + ||G(x)-G(y)||\Big] \text{ by triangle-inequality} \tag{B.10}$$

$$\leq \frac{1}{\eta}\Big[||x-y|| + (1-\gamma)||x-y||\Big] \text{ by contraction} \tag{B.11}$$

$$= \frac{2-\gamma}{\eta}||x-y||. \tag{B.12}$$

Therefore, F is also smooth with parameter $\frac{2-\gamma}{\eta}$.

Assume the mapping F is strongly monotone with parameter γ and smooth with parameter β, i.e., $||F(x)-F(y)|| \leq \beta||x-y||$. As before, define $G(x) = x - \eta F(x)$. Then

$$||G(x)-G(y)||^2 = \langle G(x)-G(y),G(x)-G(y)\rangle \tag{B.13}$$

$$= \langle (x-\eta F(x)) - (y-\eta F(y),(x-\eta F(x)) - (y-\eta F(y)\rangle \tag{B.14}$$

$$= ||x-y||^2 - 2\eta\langle F(x)-F(y),x-y\rangle + \eta^2||F(x)-F(y)||^2 \tag{B.15}$$

$$\leq (1-2\eta\gamma+\eta^2\beta^2)||x-y||^2. \tag{B.16}$$

In order for G to be a contraction, $|1-2\eta\gamma+\eta^2\beta^2|$ must be less than 1. This implies that $0<\eta<\frac{2\gamma}{\beta^2}$. If this condition is met, then G is a $\sqrt{1-2\eta\gamma+\eta^2\beta^2}$-contraction. If η is set optimally to $\frac{\gamma}{\beta^2}$, then G is a $\sqrt{1-\frac{\gamma^2}{\beta^2}}$-contraction.

B.2 Nash Equilibrium vs VI Solution

Theorem 9. *Theorem 3.1 Repeated from Cavazzuti et al. [2002]. Let $(\mathbf{C},\mathcal{X})$ be a cost minimization game with player cost functions C_i and feasible set $\mathcal{X}$. Let x^* be a Nash equilibrium. Let $F=[\frac{\partial C_1}{\partial x_1},\ldots,\frac{\partial C_N}{\partial x_N}]$. Then*

$$\langle F(x^*),x-x^*\rangle\geq 0 \quad \forall x\in \underbrace{(\{x^*+\mathbf{I}_{\mathcal{X}}(x^*)\}\cap\mathcal{X})}_{\mathcal{X}'}\subseteq\mathcal{X} \tag{B.17}$$

where $\mathbf{I}_{\mathcal{X}}(x^*)$ is the internal cone at x^* (defined on p. 494). The internal cone represents the smallest pointed cone containing the union of all possible unilateral deviations by players from the equilibrium point. Note that $\mathcal{X}'\subseteq\mathcal{X}$. When $C_i(\mathbf{x}_i,\mathbf{x}_{-i})$ is pseudoconvex in $\mathbf{x}_i$ for all i, this condition is also sufficient. Note that this is implied if F is pseudomonotone, i.e. pseudomonotonicity of F is a stronger condition.

To summarize the main takeaway, if $\langle F(x^*),x-x^*\rangle\geq 0\ \forall x\in\mathcal{X}$, i.e., x^* solves $VI(F,\mathcal{X})$, and F is pseudomonotone, then x^* is a Nash equilibrium.

B.3 Table of Maps Considered in Analysis

All maps corresponding to the (w_1,b)-subsystem in Table B.1 maintain the desired unique fixed point, $F(x^*)=0$, where $x^*=(w_1^*,b^*)=(0,\mu)$.

For the (w_2,a)-subsystem, all maps except F_{lin} with certain settings of (α,β,γ) and F_{con} maintain the desired unique fixed point, $x^*=(w_2^*,a^*)=(0,\sigma)$. F_{con} introduces an additional spurious fixed point at

Name	Map
F	$[-\nabla_\phi V; \nabla_\theta V]$
$F^{w_1,b}$	$[b-\mu, -w_1]^\top$
$F^{w_1,b}_{alt}$	$[b-\mu+\rho_k w_1, -w_1]^\top$
$F^{w_1,b}_{unr}$	$[b-\mu, \rho_k \Delta k(b-\mu) - w_1]^\top$
$F^{w_1,b}_{reg}$	$[b-\mu, -w_1 + 2\eta(b-\mu)]^\top$
$F^{w_1,b}_{con}$	$[w_1, b-\mu]^\top$
$F^{w_1,b}_{eg}$	$[w_1, b-\mu]^\top$
$F^{w_1,b}_{cc}$	$[w_1, b-\mu]^\top$
$F^{w_1,b}_{\eta cc}$	$[b-\mu+\eta w_1, -w_1+\eta(b-\mu)]^\top$
$F^{w_1,b}_{lin}$	$[\alpha(b-\mu)+(\beta+\gamma)w_1, -\alpha w_1+(\beta+\gamma)(b-\mu)]^\top$
$F^{w_2,a}$	$[a^2-\sigma^2, -2w_2 a]^\top$
$F^{w_2,a}_{alt}$	$[a^2-\sigma^2, 2\rho_k a^3 - 2a(\rho_k\sigma^2 + w_2)]^\top$
$F^{w_2,a}_{unr}$	$[a^2-\sigma^2, 4\rho_k\Delta k a^3 - 2a(2\rho_k\Delta k\sigma^2 + w_2)]^\top$
$F^{w_2,a}_{reg}$	$[a^2-\sigma^2, -2w_2 a + 4\eta a(\sigma^2+a^2)]^\top$
$F^{w_2,a}_{con}$	$[a^2-\sigma^2+4\beta w_2 a^2, 2a\beta(a^2-\sigma^2)+4\beta w_2^2 a - 2w_2 a]^\top$
$F^{w_2,a}_{eg}$	$[4w_2 a^2, 2a(a^2-\sigma^2) - 4w_2^2 a]^\top$
$F^{w_2,a}_{cc}$	$[4w_2 a^2, 2a(a^2-\sigma^2)]^\top$
$F^{w_2,a}_{eg'}$	$[w_2, \frac{a^2-\sigma^2-2w_2^2}{2a}]^\top$
$F^{w_2,a}_{cc'}$	$[w_2, \frac{a^2-\sigma^2}{2a}]^\top$
$F^{w_2,a}_{lin}$	$[\alpha(a^2-\sigma^2)+4(\beta+\gamma)w_2 a^2, 2a(\beta+\gamma)(a^2-\sigma^2)+4(\beta-\gamma)w_2^2 a - 2\alpha w_2 a]^\top$
$F^{W_2,A}_{cc}$	$2[\forall i < N : \sum_{d\le i} A_{id}A_{Nd} - \Sigma_{iN}\ , \forall i < N : -\sum_{d<N} A_{di}W_{dN}]^\top$

Table B.1: Table of vector field maps where V is the minimax objective, ρ_k is a stepsize, Δk is # of *unrolled* steps, Σ is the sample covariance matrix, N is the row of A being learned, and $\alpha, \gamma, \beta, \eta$ are hyperparameters.

$$a = \sqrt{\frac{-3 + \sqrt{9 + 32\sigma^2\beta^2}}{16\beta^2}}, \tag{B.18}$$

$$w_2 = \frac{\sigma^2 - a^2}{4\beta a^2}. \tag{B.19}$$

F_{con} is a special case of F_{lin} where $\alpha = 1$, $\beta = 1$, and $\gamma = 0$.

B.4 Minimax Solution to Constrained Multivariate LQ-GAN is Unique

Proposition 9. *Assume $z \sim p(z)$ and $y \sim p(y)$ are both in $\mathbb{R}^n$. If W_2 is constrained to be symmetric and A is constrained to be of Cholesky form, i.e., lower triangular with positive diagonal, then the unique minimax solution to Equation (3.6) is $(W_2^*, w_1^*, A^*, b^*) = (\mathbf{0}, \mathbf{0}, \Sigma^{1/2}, \mu)$ where $\Sigma^{1/2}$ is the unique, non-negative square root of Σ.*

Proof.

$$V(G,D) = \mathbb{E}_{y\sim\mathcal{N}(\mu,\Sigma)}\Big[y^\top W_2 y + w_1^\top y\Big] + \mathbb{E}_{z\sim\mathcal{N}(0,I_n)}\Big[-(Az+b)^\top W_2(Az+b) - w_1^\top(Az+b)\Big] \tag{B.20}$$

$$= \mathbb{E}_{y\sim\mathcal{N}(\mu,\Sigma)}\Big[\sum_i\sum_j W_{2ij}y_iy_j + \sum_i w_{1i}y_i\Big] \tag{B.21}$$

$$- \mathbb{E}_{z\sim\mathcal{N}(0,I_n)}\Big[\sum_i\sum_j W_{2ij}(b_i + \sum_k A_{ik}z_k)(b_j + \sum_k A_{jk}z_k) + \sum_i w_{1i}(b_i + \sum_k A_{ik}z_k)\Big]. \tag{B.22}$$

Taking derivatives and setting equal to zero, we find that the fixed point at the interior is unique:

$$\dot{W}_2 = \mathbb{E}_{y\sim\mathcal{N}(\mu,\Sigma)}\Big[yy^\top\Big] - \mathbb{E}_{z\sim\mathcal{N}(0,I_n)}\Big[(Az+b)(Az+b)^\top\Big] \tag{B.23}$$

$$\dot{w}_1 = \mathbb{E}_{y\sim\mathcal{N}(\mu,\Sigma)}\Big[y\Big] - \mathbb{E}_{z\sim\mathcal{N}(0,I_n)}\Big[(Az+b)\Big] \tag{B.24}$$

$$\dot{A} = \mathbb{E}_{z\sim\mathcal{N}(0,I_n)}\Big[(W_2+W_2^\top)Azz^\top + (W_2+W_2^\top)bz^\top + w_1z^\top\Big] \tag{B.25}$$

$$\dot{b} = \mathbb{E}_{z\sim\mathcal{N}(0,I_n)}\Big[(W_2+W_2^\top)Az + (W_2+W_2^\top)b + w_1\Big]. \tag{B.26}$$

Setting derivatives equal to zero:

$$\dot{w}_1 = \mu - b = 0 \Rightarrow b = \mu \tag{B.27}$$

$$\dot{W}_2 = \mathbb{E}_{y\sim\mathcal{N}(\mu,\Sigma)}\Big[(y-\mu)(y-\mu)^\top + \mu y^\top + y\mu^\top - \mu\mu^\top\Big] - \mathbb{E}_{z\sim\mathcal{N}(0,I_n)}\Big[(Az+b)(Az+b)^\top\Big] \tag{B.28}$$

$$= \Sigma + \mu\mu^\top - \mathbb{E}_{z\sim\mathcal{N}(0,I_n)}\Big[Azz^\top A^\top + Azb^\top + b(Az)^\top + bb^\top\Big] \tag{B.29}$$

$$= \Sigma + \mu\mu^\top - AA^\top - bb^\top = \Sigma - AA^\top = 0 \Rightarrow A = \Sigma^{1/2} \tag{B.30}$$

$$\dot{A} = \mathbb{E}_{z\sim\mathcal{N}(0,I_n)}\Big[(W_2+W_2^\top)Azz^\top + (W_2+W_2^\top)bz^\top + w_1z^\top\Big] \tag{B.31}$$

$$= (W_2+W_2^\top)A = 0 \Rightarrow W_2 + W_2^\top = 0 \Rightarrow W_2 = -W_2^\top = 0 \tag{B.32}$$

$$\dot{b} = \mathbb{E}_{z\sim\mathcal{N}(0,I_n)}\Big[(W_2+W_2^\top)Az + (W_2+W_2^\top)b + w_1\Big] \tag{B.33}$$

$$= (W_2+W_2^\top)b + w_1 = w_1 = 0. \tag{B.34}$$

The last implication in Equation (B.30) follows because A is constrained to be of Cholesky form, i.e., lower triangular with positive diagonal, and every symmetric positive definite matrix has a unique Cholesky decomposition.

The second to last implication of Equation (B.32) follows because $A = \Sigma^{1/2}$ is necessarily full rank. Note this implies $A^\top$ is also full rank. The null space of a full rank matrix is the zeros vector, which implies $W_2 + W_2^\top = 0$. W_2 is symmetric, so this implies $W_2 = 0$. □

B.5 Divergence of Simultaneous Gradient Descent for the (w_1, b)-Subsystem

Consider the case where the mean of $p(z)$ is zero:

$$F^{w_1,b} = [b, -w_1] = J^{w_1,b}x, \tag{B.35}$$

$$J^{w_1,b} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \tag{B.36}$$

$$x_k = [w_{1,k}, b_k]^\top, \tag{B.37}$$

$$x_{k+1} = x_k - \rho_k F^{w_1,b}(x_k), \tag{B.38}$$

$$x^* = [0, 0]. \tag{B.39}$$

We will show that simultaneous gradient descent always produces an iterate that is farther away from the equilibrium than the previous iterate, i.e. $||x_{k+1} - x^*||^2/||x_k - x^*||^2 > 1$.

$$||x_{k+1} - x^*||^2/||x_k - x^*||^2 = ||x_k - \rho_k J^{w_1,b} x_k||^2/||x_k||^2 \tag{B.40}$$

$$= ||(I - \rho_k J^{w_1,b})x_k||^2/||x_k||^2 \tag{B.41}$$

$$= \frac{x_k^\top (I - \rho_k J^{w_1,b})^\top (I - \rho_k J^{w_1,b}) x_k}{x_k^\top x_k} \tag{B.42}$$

$$= \frac{x_k^\top M x_k}{x_k^\top x_k} \quad \text{Rayleigh quotient of } M \tag{B.43}$$

$$\geq \lambda_{\min}(M), \tag{B.44}$$

where

$$M = (I - \rho_k J^{w_1,b})^\top (I - \rho_k J^{w_1,b}) \tag{B.45}$$

$$= \begin{bmatrix} 1+\rho_k^2 & 0 \\ 0 & 1+\rho_k^2 \end{bmatrix}, \tag{B.46}$$

$$\lambda_{\min}(M) = 1 + \rho_k^2 > 1. \tag{B.47}$$

Therefore, simultaneous gradient descent diverges from the equilibrium of the (w_1, b)-subsystem for any step size scheme, ρ_k.

B.6 Derivation of *Crossing-the-Curl*

Here, we derive our proposed technique in 3-D, however, the result of the derivation can be computed in arbitrary dimensions:

$$(\nabla \times F) \times F = -F \times (\nabla \times F) \tag{B.48}$$

$$= -v \times (\nabla \times F) \text{ where } v = F \tag{B.49}$$

$$= -\nabla_F (v \cdot F) + (v \cdot \nabla) F \text{ where } \nabla_F \text{ is Feynman notation} \tag{B.50}$$

$$= -\Big(v_1 \Big[\frac{\partial F_1}{\partial x_1}, \ldots, \frac{\partial F_1}{\partial x_n} \Big] + \ldots + v_n \Big[\frac{\partial F_n}{\partial x_1}, \ldots, \frac{\partial F_n}{\partial x_n} \Big] \Big) \tag{B.51}$$

$$+ \Big(v_1 \frac{\partial}{\partial x_1} + \ldots + v_n \frac{\partial}{\partial x_n} \Big) F \tag{B.52}$$

$$= (J - J^\top) F. \tag{B.53}$$

B.7 Monotonicity: Definitions and Requirements

For all $x \in \mathcal{X}$ and $x' \in \mathcal{X}$,

$$\langle F(x) - F(x'), x - x' \rangle (> 0, \geq s||x - x'||^2) \geq 0 \quad \text{(strictly, s-strongly)-monotone,} \tag{B.54}$$

$$\langle F(x'), x - x' \rangle \geq 0 \implies \langle F(x), x - x' \rangle (> 0) \geq 0 \quad \text{(strictly-)pseudomonotone,} \tag{B.55}$$

$$\langle F(x'), x - x' \rangle > 0 \implies \langle F(x), x - x' \rangle \geq 0 \quad \text{quasimonotone.} \tag{B.56}$$

While we used these definitions in our analysis for certain cases, the following alternate requirements proposed in [22] made the complete analysis of the system tractable. We restate them here for convenience. Note that we what we refer to as condition (B) in the main body of the paper is actually a stronger version of condition (C) below with $v = (x^* - x)/t$.

Consider the following conditions:

(A) For all $x \in \mathcal{X}$ and $v \in \mathbb{R}^n$ such that $v^\top F(x) = 0$ we have $v^\top J(x) v \geq 0$.

(B) For all $x \in \mathcal{X}$ and $v \in \mathbb{R}^n$ such that $F(x) = 0$, $v^\top J(x) v = 0$, and $v^\top F(x + \tilde{t} v) > 0$ for some $\tilde{t} < 0$, we have that for all $\bar{t} > 0$, there exists $t \in (0, \bar{t}]$ such that $t \in I_{x,v}$ and $v^\top F(x + tv) \geq 0$.

(C) For all $x \in \mathcal{X}$ and $v \in \mathbb{R}^n$ such that $F(x) = 0$ and $v^\top J(x) v = 0$, we have that for all $\bar{t} > 0$, there exists $t \in (0, \bar{t}]$ such that $t \in I_{x,v}$ and $v^\top F(x + tv) \geq 0$.

Theorem 10 ([22], Theorem 3)**.** *Let $F : \mathcal{X} \to \mathbb{R}^n$ be differentiable on the open convex set $\mathcal{X} \subset \mathbb{R}^n$.*

(i) F is quasimonotone on $\mathcal{X}$ if and only if (A) and (B') hold.

(ii) F is pseudomonotone on $\mathcal{X}$ if and only if (A) and (C') hold.

B.8 A Comparison of Monotonicity and Hurwitz

The monotonicity and Hurwitz properties are complementary.

B.8.1 Hurwitz Does Not Imply Quasimonotonicity

Let $F(x) = Jx$, $J = \begin{bmatrix} 1 & 4 \\ -1 & 1 \end{bmatrix}$, $S = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, and $v = SJx = [-x_1 + x_2, -x_1 - 4x_2]^\top$. Then $\lambda_{1,2}(J) = 1 \pm 2i$ so J is Hurwitz, and

$$[v^\top Jv]\Big|_{(-1,1)} = [x_1^2 + 3x_1x_2 + x_2^2]\Big|_{(-1,1)} = -1, \tag{B.57}$$

which, by condition (A), implies F is not quasimonotone.

B.8.2 Monotonicity Does Not Imply Hurwitz

Let $F(x) = Jx$ and $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. Then $\lambda_{1,2}(J) = \pm i$ so J is not Hurwitz, but

$$J + J^\top = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \succeq 0, \quad \lambda_{1,2} = 0, \tag{B.58}$$

so F is monotone.

B.8.3 Monotonicity and Hurwitz Can Overlap

Let $F(x) = Jx$ and $J = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Then $\lambda_{1,2}(J) = 1$ so J is Hurwitz and

$$J + J^\top = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \succeq 0, \quad \lambda_{1,2} = 1, \tag{B.59}$$

so F is monotone.

Proposition 10 ((Strict, Strong)-Monotonicity Implies Hurwitz)**.** *If F is differentiable and strictly-monontone, then the Jacobian of F, J, is Hurwitz. If F is differentiable and s-strongly-monotone, then J is Hurwitz with* $\min(\mathbb{R}(\lambda)) \geq s$.

Proof. Assume A is a real, square matrix and A is either positive definite or strongly-positive definite, i.e. $v^\top Av \succeq 0$ or $v^\top Av \succeq s||v||^2$ with $v \in \mathbb{C}^n$. Let * denote the conjugate transpose and note that $\langle u, w\rangle = u^*w$. Let $\lambda = a + bi$ be a potentially complex eigenvalue of A and v be its corresponding eigenvector, i.e. $Av = \lambda v$. We aim to prove that if A satisfies the above assumptions, then $a > 0$, i.e., A is Hurwitz.

$$\langle (A + A^\top)v, v\rangle = \langle Av, v\rangle + \langle A^\top v, v\rangle \tag{B.60}$$

$$\langle A^\top v, v\rangle = (A^\top v)^* v \tag{B.61}$$

$$= v^*(A^\top)^* v \tag{B.62}$$

$$= v^*(Av) \text{ because } A \text{ is real} \tag{B.63}$$

$$= \langle v, Av\rangle \tag{B.64}$$

$$0 < (\text{ or } s||v||^2 \leq)\langle \frac{1}{2}(A + A^\top)v, v\rangle \tag{B.65}$$

$$= \frac{1}{2}(\langle Av, v\rangle + \langle v, Av\rangle) \tag{B.66}$$

$$= \frac{1}{2}((a + bi)\langle v, v\rangle + \overline{(a + bi)}\langle v, v\rangle) \tag{B.67}$$

$$= \frac{1}{2}[(a + bi)||v||^2 + (a - bi)||v||^2] \tag{B.68}$$

$$= a||v||^2 \tag{B.69}$$

$$\Rightarrow a > 0 \text{ or } a \geq s. \tag{B.70}$$

If F is (strictly, strongly)-monotone, then the Jacobian of F is a real, square, (positive definite, strongly-positive definite) matrix, therefore, it matches the above assumptions. Hence, the conclusion follows. □

B.9 *Crossing-the-Curl* Can Make Monotone Fields, Non-Monotone

Here, we provide examples of negative results for *Crossing-the-Curl*. This is to emphasize that our proposed technique can cause problems if not used with cau-

tion. The headings below describe the before and afters when applying our proposed technique to the map $F(x) = Jx$.

Monotone to Non-Monotone.

$$J = \begin{bmatrix} 4 & 1 \\ -1 & 1 \end{bmatrix} \tag{B.71}$$

$$J^{sym} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}, \lambda_{1,2} = 4, 1 \tag{B.72}$$

$$J_{cc}^{sym} = \begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix}, \lambda_{1,2} = 5, -1 \tag{B.73}$$

Increase in condition number: $\kappa = {}^{11}\!/_{5} \to 4$.

$$J = \begin{bmatrix} 1 & {}^{1}\!/_{4} \\ -1 & 1 \end{bmatrix} \tag{B.74}$$

$$J^{sym} = \begin{bmatrix} 1 & -{}^{3}\!/_{8} \\ -{}^{3}\!/_{8} & 1 \end{bmatrix}, \lambda_{1,2} = {}^{11}\!/_{8}, {}^{5}\!/_{8} \tag{B.75}$$

$$J_{cc}^{sym} = \begin{bmatrix} {}^{5}\!/_{4} & 0 \\ 0 & {}^{5}\!/_{16} \end{bmatrix}, \lambda_{1,2} = {}^{5}\!/_{4}, {}^{5}\!/_{16} \tag{B.76}$$

Saddle becomes Monotone.

$$J = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \tag{B.77}$$

$$J^{sym} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \lambda_{1,2} = -1, 1 \tag{B.78}$$

$$J_{cc}^{sym} = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}, \lambda_{1,2} = 4, 0 \tag{B.79}$$

Unstable point becomes stable.

$$J = \begin{bmatrix} -2 & 1 \\ -1 & -1 \end{bmatrix} \tag{B.80}$$

$$J^{sym} = \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix}, \lambda_{1,2} = -2, -1 \tag{B.81}$$

$$J^{sym}_{cc} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \lambda_{1,2} = 3, 1 \tag{B.82}$$

$F^{w_2,a}_{eg'}$ becomes non-monotone.

$$F = \begin{bmatrix} w_2 \\ \frac{a^2-\sigma^2-2w_2^2}{2a} \end{bmatrix} \tag{B.83}$$

$$F_{cc} = \begin{bmatrix} -\frac{w_2(a^2-\sigma^2-2w_2^2)}{2a^2} \\ \frac{w_2^2}{a} \end{bmatrix} \tag{B.84}$$

$$Tr[J_{cc}]\Big|_{w_2=0,a=2\sigma} = -\frac{3}{8} \Rightarrow J_{cc} \not\succeq 0 \tag{B.85}$$

Proposition 11. Crossing-the-Curl *forces monotonicity for normal, affine fields.*

Proof. Let $F = Jx + b$ and assume J is normal, i.e., $JJ^\top = J^\top J$. Then

$$F_{cc} = (J^\top - J)F \tag{B.86}$$
$$= (J^\top - J)(Jx + b) \tag{B.87}$$
$$J_{cc} = (J^\top - J)J \tag{B.88}$$
$$= (J^\top J - JJ) \tag{B.89}$$
$$J_{cc}^{sym} = \frac{2J^\top J - JJ - J^\top J^\top}{2} \tag{B.90}$$
$$= \frac{J^\top J + JJ^\top - JJ - J^\top J^\top}{2} + \frac{J^\top J - JJ^\top}{2} \tag{B.91}$$
$$= \frac{J^\top J + JJ^\top - JJ - J^\top J^\top}{2} \text{ because } J \text{ is normal} \tag{B.92}$$
$$= \frac{-(J - J^\top)(J - J^\top)}{2} \tag{B.93}$$
$$= \frac{(J - J^\top)^\top (J - J^\top)}{2} \tag{B.94}$$
$$z^\top J_{cc}^{sym} z = \frac{1}{2}[(J - J^\top)z]^\top [(J - J^\top)z] \tag{B.95}$$
$$= \frac{1}{2}||(J - J^\top)z||^2 \geq 0 \Rightarrow J_{cc} \succeq 0. \tag{B.96}$$

□

B.10 Analysis of the (w_1, b)-Subsystem

Proposition 12. *Unrolled GANs and Alternating Updates are Monotone for the (w_1, b)-subsystem.*

Proof. In Unrolled GANs, the generator computes the gradient of V assuming the discriminator has already made several updates. Define the discriminator's update as

$$w_{1,k+1} = w_{1,k} - \rho F_{w_1}(w_{1,k}, b_k) = U_k(w_{1,k}), \tag{B.97}$$

and denote the composition of U, Δk-times as

$$U_k^{\Delta k}(w_{1,k}) = U_k(\cdots(U_k(U_k(w_{1,k}))\cdots) \tag{B.98}$$

where Δk is some positive integer. Then the update for Unrolled GANs is

$$w_{1,k+1} = w_{1,k} - \rho\frac{\partial V(w_{1,k}, b_k)}{\partial w_1} \tag{B.99}$$

$$b_{k+1} = b_k - \rho\frac{\partial V(U_k^{\Delta k}(w_{1,k}), b_k)}{\partial b}. \tag{B.100}$$

In the case of the (w_1, b)-subsystem, we can write these unrolled updates out explicitly. Remember $F = [b - \mu, -w_1]^\top$, so

$$U_k(w_{1,k}) = w_{1,k} - \rho(b_k - \mu), \tag{B.101}$$

$$U_k^{\Delta k}(w_{1,k}), b_k) = w_{1,k} - \rho\Delta k(b_k - \mu). \tag{B.102}$$

Plugging this back in, we find

$$w_{1,k+1} = w_{1,k} - \rho(b_k - \mu) \tag{B.103}$$

$$b_{k+1} = b_k - \rho(\rho\Delta k(b_k - \mu) - w_{1,k}), \tag{B.104}$$

where the corresponding map is $F^{unr} = [b_k - \mu, \rho\Delta k(b_k - \mu) - w_{1,k}]$. Taking a look at the Jacobian, we find

$$J^{unr} = \begin{bmatrix} 0 & 1 \\ -1 & \rho\Delta k \end{bmatrix} \tag{B.105}$$

$$J_{sym}^{unr} = \begin{bmatrix} 0 & 0 \\ 0 & \rho\Delta k \end{bmatrix} \succeq 0. \tag{B.106}$$

Now, consider alternating updates:

$$w_{1,k+1} = w_{1,k} - \rho(b_{k+1} - \mu) \tag{B.107}$$

$$= w_{1,k} - \rho(b_k - \rho(-w_{1,k}) - \mu) \tag{B.108}$$

$$b_{k+1} = b_k - \rho(-w_{1,k}). \tag{B.109}$$

Here, we considered updating b first, but the (w_1, b)-subsystem is perfectly symmetric, so the analysis holds either way. If w_1 is updated first, this is equivalent to Unrolled GAN with $\Delta k = 1$ (see Equation B.104). The Jacobian is

$$J^{alt} = \begin{bmatrix} \rho & 1 \\ -1 & 0 \end{bmatrix} \tag{B.110}$$

$$J^{alt}_{sym} = \begin{bmatrix} \rho & 0 \\ 0 & 0 \end{bmatrix} \succeq 0. \tag{B.111}$$

The Jacobian's for Unrolled GAN and alternating descent are both positive semidefinite, therefore, their maps are monotone (but not strictly-monotone). Note that these results imply neither is Hurwitz either because both Jacobians exhibit a zero eigenvalue. □

Proposition 13. *F_{lin}, F_{cc}, F_{eg}, and F_{con} are strongly-monotone for the (w_1, b)-subsystem (includes multivariate case). F and F_{reg} are monotone, but not strictly monotone. Moreover, F_{lin}, F_{cc}, F_{eg}, F_{con}, and F_{reg} are Hurwitz for the (w_1, b)-subsystem (includes multivariate case). F is not Hurwitz.*

Proof. We start with the original map, $F^{w_1,b}$, and its Jacobian.

$$F = \begin{bmatrix} b - \mu \\ -w_1 \end{bmatrix} \tag{B.112}$$

$$J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \tag{B.113}$$

$$J = J_{sym} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \succeq 0 \tag{B.114}$$

The symmetrized Jacobian is positive semidefinite, therefore this system is monotone. Also, the real parts of the eigenvalues of its Jacobian are zero, therefore, J is not Hurwitz.

Now we analyze $F_{cc}^{w_1,b}$, $F_{eg}^{w_1,b}$, and $F_{con}^{w_1,b}$, which as discussed in the main body, are equivalent.

$$F_{cc} = F_{eg} = F_{con} = \begin{bmatrix} w_1 \\ b - \mu \end{bmatrix} \tag{B.115}$$

$$J = J_{sym} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \succeq 1 \tag{B.116}$$

The symmetrized Jacobian is positive definite with a minimum eigenvalue of 1, therefore this system is 1-strongly-monotone. By Proposition 10, the Jacobians of these maps are Hurwitz for the (w_1, b)-subsystem.

Now we analyze the generalization $F_{lin}^{w_1,b} = (\alpha I - \beta J^\top - \gamma J) F^{w_1,b}$.

$$F_{lin} = \begin{bmatrix} \alpha(b-\mu) + (\beta+\gamma)w_1 \\ -\alpha w_1 + (\beta+\gamma)(b-\mu) \end{bmatrix} \tag{B.117}$$

$$J = \begin{bmatrix} (\beta+\gamma)I & \alpha I \\ -\alpha I & (\beta+\gamma)I \end{bmatrix} \tag{B.118}$$

$$J_{sym} = \begin{bmatrix} (\beta+\gamma)I & 0 \\ 0 & (\beta+\gamma)I \end{bmatrix} \succeq \beta+\gamma \tag{B.119}$$

The symmetrized Jacobian is positive definite with a minimum eigenvalue of $(\beta+\gamma)$, therefore this system is $(\beta+\gamma)$-strongly-monotone. By Proposition 10, $J_{lin}^{w_1,b}$ is Hurwitz for the (w_1, b)-subsystem.

Now we analyze the regularized-gradient algorithm, $F_{reg}^{w_1,b}$.

$$F_{reg} = \begin{bmatrix} b-\mu \\ -w_1 + 2\eta(b-\mu) \end{bmatrix}, \eta > 0 \tag{B.120}$$

$$J_{reg} = \begin{bmatrix} 0 & I \\ -I & 2\eta I \end{bmatrix}, \lambda_{1,2} = \eta \pm \sqrt{\eta^2 - 1} \Rightarrow \mathbb{R}(\lambda_{1,2}) > 0 \tag{B.121}$$

$$J_{regsym} = \begin{bmatrix} 0 & 0 \\ 0 & 2\eta I \end{bmatrix} \succeq 0 \tag{B.122}$$

Therefore, this map is monotone (but not strictly or strongly-monotone). Also, the real parts of the eigenvalues of its Jacobian are strictly positive, therefore, $J_{reg}^{w_1,b}$ is Hurwitz.

Note that for F_{cc}, F_{eg}, F_{con}, and F_{lin}, J is symmetric, therefore, F is the gradient of some function, $f(w_1, b) = \frac{1}{2}(w_1^2 + (b-\mu)^2)$. Also, note that the standard algorithm with step size $\rho_k = \frac{1}{k+1}$ is equivalent to the standard running estimate of the mean: $\mu_{k+1} = \frac{k}{k+1}\mu_k + \frac{1}{k+1}x_k$ where x_k is the k-th sample.

□

B.11 A Linear Combination of F, JF, and $J^\top F$ is Not Quasimonotone for the 1-d LQ-GAN

The Jacobian of F_{lin}, written below, will be useful for the proof. The proof proceeds by process of elimination, ruling out different regions of the space $[\alpha, \beta, \gamma] \in \mathbb{R}^3$ by showing that any F_{lin} with those constants is not quasimonotone.

$$(\alpha I + \beta J^\top - \gamma J)F = \begin{bmatrix} \alpha & 0 & -2(\beta+\gamma)a & -2(\beta+\gamma)b \\ 0 & \alpha & 0 & -(\beta+\gamma) \\ 2(\beta+\gamma)a & 0 & \alpha - 2(\beta-\gamma)w_2 & 0 \\ 2(\beta+\gamma)b & (\beta+\gamma) & 0 & \alpha - 2(\beta-\gamma)w_2 \end{bmatrix} \begin{bmatrix} -\sigma^2 + a^2 + b^2 \\ b \\ -2w_2 a \\ -2w_2 b - w_1 \end{bmatrix} \tag{B.123}$$

$$= \begin{bmatrix} \alpha(-\sigma^2 + a^2 + b^2) + 4(\beta+\gamma)w_2(a^2+b^2) + 2(\beta+\gamma)w_1 b \\ \alpha b + (\beta+\gamma)(2w_2 b + w_1) \\ 2a(\beta+\gamma)(-\sigma^2 + a^2 + b^2) + 4(\beta-\gamma)w_2^2 a - 2\alpha w_2 a \\ 2(\beta+\gamma)b(-\sigma^2 + a^2 + b^2) + (\beta+\gamma)b + (2(\beta-\gamma)w_2 - \alpha)(2w_2 b + w_1) \end{bmatrix} \tag{B.124}$$

Specifically, we first consider the sign of $\beta+\gamma$. Lemma 6 rules out negative values. Lemma 7 rules out positive values when $\sigma^2 \leq 1/2$, and Lemma 8 rules out positive values when $\sigma^2 > 1/2$. Corollary 2 concludes that $\beta + \gamma = 0$.

Next, given $\beta + \gamma = 0$, we consider the sign of α. Lemmas 9 and 10 rule out positive values of α when β is greater than or less than or equal to zero respectively, i.e., α cannot be positive. Similarly, Lemmas 11 and 12 rule out negative values of α when β is less than or greater than or equal to zero respectively, i.e., α cannot be negative. Corollary 3 concludes that $\alpha = 0$.

Lastly, given that $\beta + \gamma = \alpha = 0$, Lemmas 13 and 14 prove that β cannot be greater than or less than or equal to zero respectively. Corollary 4 concludes that

$\beta = \gamma = 0$. Therefore, the only quasimonotone linear combination is the trivial one resulting in $F = 0$, which completes the proof.

Lemma 6. *For F_{lin} to be quasimonotone, $\beta + \gamma$ must not be strictly less than zero, i.e. $\beta + \gamma \not< 0$.*

Proof. Consider

$$y = [0, 0, \sigma, -\sigma] \tag{B.125}$$

$$x = [0, 0, \sigma, \sigma] \tag{B.126}$$

$$\langle F(y), x - y \rangle = 2\sigma F_b(y) = -2\sigma^2(\beta + \gamma)(1 - 2\sigma^2 + 2\sigma^2 + 2\sigma^2) \tag{B.127}$$

$$= -2\sigma^2(\beta + \gamma)(1 + 2\sigma^2) \tag{B.128}$$

$$\langle F(x), x - y \rangle = 2\sigma F_b(x) = 2\sigma^2(\beta + \gamma)(1 + 2\sigma^2) \tag{B.129}$$

If $(\beta + \gamma) < 0$, then this system is not quasimonotone. Therefore, assume $(\beta + \gamma) \geq 0$ from now on. □

Lemma 7. *If $\sigma^2 \leq \frac{1}{2}$, for F_{lin} to be quasimonotone, $\beta + \gamma$ must not be strictly greater than zero, i.e. $\beta + \gamma \not> 0$.*

Proof. We will use a different parameterization of F_{lin} for this part of the proof.

$$J_{skew} = (J^\top - J)/2 \tag{B.130}$$

$$J_{sym} = (J^\top + J)/2 \tag{B.131}$$

$$\beta = (\hat{\beta} + \hat{\gamma})/2 \tag{B.132}$$

$$\gamma = (\hat{\beta} - \hat{\gamma})/2 \tag{B.133}$$

$$\hat{\beta} = \beta + \gamma \tag{B.134}$$

$$\hat{\gamma} = \beta - \gamma \tag{B.135}$$

The linear combination is now defined as

$$(\alpha I + \hat{\beta} J_{skew} + \hat{\gamma} J_{sym})F = \begin{bmatrix} \alpha & 0 & -2\hat{\beta}a & -2\hat{\beta}b \\ 0 & \alpha & 0 & -\hat{\beta} \\ 2\hat{\beta}a & 0 & \alpha - 2\hat{\gamma}w_2 & 0 \\ 2\hat{\beta}b & \hat{\beta} & 0 & \alpha - 2\hat{\gamma}w_2 \end{bmatrix} \begin{bmatrix} -\sigma^2 + a^2 + b^2 \\ b \\ -2w_2 a \\ -2w_2 b - w_1 \end{bmatrix} \tag{B.136}$$

$$= \begin{bmatrix} \alpha(-\sigma^2 + a^2 + b^2) + 4\hat{\beta}w_2(a^2 + b^2) + 2\hat{\beta}w_1 b \\ \alpha b + \hat{\beta}(2w_2 b + w_1) \\ 2a\hat{\beta}(-\sigma^2 + a^2 + b^2) + 4\hat{\gamma}w_2^2 a - 2\alpha w_2 a \\ 2\hat{\beta}b(-\sigma^2 + a^2 + b^2) + \hat{\beta}b + (2\hat{\gamma}w_2 - \alpha)(2w_2 b + w_1) \end{bmatrix} \tag{B.137}$$

In order for a system to be quasimonotone, we require condition (A) (among other properties). We will now show that this property is not satisfied for F_{lin} with $\hat{\beta} > 0$ by considering two different cases.

Case 1: Consider the (w_2, a)-subsystem. Let

$$v = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \overbrace{\begin{bmatrix} \alpha(-\sigma^2 + a^2 + b^2) + 4\hat{\beta}w_2(a^2 + b^2) + 2\hat{\beta}w_1 b \\ \alpha b + \hat{\beta}(2w_2 b + w_1) \\ 2a\hat{\beta}(-\sigma^2 + a^2 + b^2) + 4\hat{\gamma}w_2^2 a - 2\alpha w_2 a \\ 2\hat{\beta}b(-\sigma^2 + a^2 + b^2) + \hat{\beta}b + (2\hat{\gamma}w_2 - \alpha)(2w_2 b + w_1) \end{bmatrix}}^{F_{lin}} \tag{B.138}$$

$$= \begin{bmatrix} -2a\hat{\beta}(-\sigma^2 + a^2 + b^2) - 4\hat{\gamma}w_2^2 a + 2\alpha w_2 a \\ 0 \\ \alpha(-\sigma^2 + a^2 + b^2) + 4\hat{\beta}w_2(a^2 + b^2) + 2\hat{\beta}w_1 b \\ 0 \end{bmatrix} \tag{B.139}$$

Above, we premultiply F_{lin} by a skew symmetric matrix, which ensures $v^\top F_{lin} = F_{lin}^\top A_{skew} F_{lin} = 0$.

The relevant portion of the Jacobian of F_{lin} is

$$J_{lin}^{w_2,a} = \begin{bmatrix} 4\hat{\beta}(a^2 + b^2) & 2a\alpha + 8\hat{\beta}w_2 a \\ 8\hat{\gamma}w_2 a - 2\alpha a & 2\hat{\beta}(-\sigma^2 + 3a^2 + b^2) + 4\hat{\gamma}w_2^2 - 2\alpha w_2 \end{bmatrix} \tag{B.140}$$

Consider $x = [0, 0, c\sigma, 0]$ and both $\hat{\beta}$ and α fixed.

$$v^\top J_{lin}^{w_2,a} v = \lim_{c \to 0^+} 2\hat{\beta}(-1 + c^2)^2 \sigma^6 [\alpha^2(-1 + 3c^2) + 8\hat{\beta}^2 c^4 \sigma^2] \tag{B.141}$$

$$= -2\hat{\beta}\sigma^6 \alpha^2 \geq 0 \tag{B.142}$$

This implies either $\alpha = 0$ or $\hat{\beta} \leq 0$ for the system to be quasimonotone.

Case 2: Consider the (a, b)-subsystem. Let

$$v = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \overbrace{\begin{bmatrix} \alpha(-\sigma^2 + a^2 + b^2) + 4\hat{\beta}w_2(a^2 + b^2) + 2\hat{\beta}w_1 b \\ \alpha b + \hat{\beta}(2w_2 b + w_1) \\ 2a\hat{\beta}(-\sigma^2 + a^2 + b^2) + 4\hat{\gamma}w_2^2 a - 2\alpha w_2 a \\ 2\hat{\beta}b(-\sigma^2 + a^2 + b^2) + \hat{\beta}b + (2\hat{\gamma}w_2 - \alpha)(2w_2 b + w_1) \end{bmatrix}}^{F_{lin}} \tag{B.143}$$

$$= \begin{bmatrix} 0 \\ 0 \\ -2\hat{\beta}b(-\sigma^2 + a^2 + b^2) - \hat{\beta}b - (2\hat{\gamma}w_2 - \alpha)(2w_2 b + w_1) \\ 2a\hat{\beta}(-\sigma^2 + a^2 + b^2) + 4\hat{\gamma}w_2^2 a - 2\alpha w_2 a \end{bmatrix} \tag{B.144}$$

The relevant portion of the Jacobian of F_{lin} is

$$J_{lin}^{a,b} = \begin{bmatrix} 2\hat{\beta}(-\sigma^2 + 3a^2 + b^2) + 4\hat{\gamma}w_2^2 - 2\alpha w_2 & 4ab\hat{\beta} \\ 4ab\hat{\beta} & 2\hat{\beta}(-\sigma^2 + a^2 + 3b^2) + \hat{\beta} + 2w_2(2\hat{\gamma}w_2 - \alpha) \end{bmatrix} \tag{B.145}$$

Consider $\alpha = 0$ and $x = [0, 0, \frac{\sigma}{10}, \frac{\sigma}{2}]$. Then

$$v^\top J_{lin}^{a,b} v = \hat{\beta}^3 \sigma^4 \underbrace{(-1.44 + 4.46842\sigma^2 - 3.37146\sigma^4)}_{<0 \ \forall \ \sigma^2 \in (0,1/2]} \tag{B.146}$$

Then $\alpha = 0 \Rightarrow \hat{\beta} \leq 0$. In either case, $\hat{\beta}$ must be nonpositive. Therefore, $\hat{\beta} = \beta + \gamma \not\geq 0$. □

Alternate Proof for Lemma 7. Part of the proof in Lemma 7 looks at the limit in which a approaches 0. One might presume a simple fix is to constrain a to be larger

than some small value, e.g., 1e-10, and use a large $\hat{\beta}$ value. Here, we show that even using $a = \frac{\sigma}{100}$ breaks quasimonotonicity. The variance of the data distribution is assumed to be unknown, which would make it very difficult to select a proper lower bound for a that maintains quasimonotonicity within the feasible region.

Consider $x = [0, -1, \frac{\sigma}{100}, \frac{\sigma}{2}]$ and the (a, b)-subsystem as in Lemma 7. Then

$$\begin{aligned} v^\top J v = &\Big(-5.9976\hat{\beta}\sigma^2 \Big)\alpha^2 + \Big(\hat{\beta}^2\sigma^3(-5.9976 + 8.9976\sigma^2)\Big)\alpha \\ &+ \hat{\beta}^3\sigma^4(-1.4994 + 4.4997\sigma^2 - 3.375\sigma^4). \end{aligned} \tag{B.147}$$

If $\hat{\beta} > 0$, then this is a concave quadratic form in α. To find where this function is positive, we need to find its roots.

$$\alpha_{\pm} = \frac{-\Big(\hat{\beta}^2\sigma^3(-5.9976 + 8.9976\sigma^2)\Big)}{2\Big(-5.9976\hat{\beta}\sigma^2\Big)} \tag{B.148}$$

$$\pm \frac{\sqrt{A-B}}{2\Big(-5.9976\hat{\beta}\sigma^2\Big)} \tag{B.149}$$

where (B.150)

$$A = \Big(\hat{\beta}^2\sigma^3(-5.9976 + 8.9976\sigma^2)\Big)^2 \tag{B.151}$$

$$B = 4\Big(-5.9976\hat{\beta}\sigma^2\Big)\Big(\hat{\beta}^3\sigma^4(-1.4994 + 4.4997\sigma^2 - 3.375\sigma^4)\Big) \tag{B.152}$$

$$\sqrt{A-B}^2 = \hat{\beta}^4\sigma^6(5.9976^2 - (2)(5.9976)(8.9976)\sigma^2 + 8.7616^2\sigma^4) \tag{B.153}$$

$$+ 4(5.9976)\hat{\beta}^4\sigma^6(-1.4994 + 4.4997\sigma^2 - 3.375\sigma^4) \tag{B.154}$$

$$= \hat{\beta}^4\sigma^6\Big(5.9976^2 - (4)(1.4994)(5.9976) + (5.9976)[(4.4997)(4) - (2)(8.9976)]\sigma^2 \tag{B.155}$$

$$+ [8.9976^2 - (4)(5.9976)(3.375)]\sigma^4\Big) \tag{B.156}$$

$$= \hat{\beta}^4\sigma^6\Big(0.02159136\sigma^2 - 0.01079424\sigma^4\Big) \tag{B.157}$$

$$= \hat{\beta}^4\sigma^8\Big(0.02159136 - 0.01079424\sigma^2\Big) \tag{B.158}$$

$$\frac{\sqrt{A-B}}{2(-5.9976\hat{\beta}\sigma^2)} = -\hat{\beta}\sigma^2\sqrt{(0.02159136 - 0.01079424\sigma^2)/(2^2 * 5.9976^2)} \tag{B.159}$$

$$= -\hat{\beta}\sigma^2\sqrt{0.00015006002 - 0.00007502\sigma^2} \tag{B.160}$$

$$\frac{-b}{2a} = \frac{-\Big(\hat{\beta}^2\sigma^3(-5.9976 + 8.9976\sigma^2)\Big)}{2\Big(-5.9976\hat{\beta}\sigma^2\Big)} \tag{B.161}$$

$$= \hat{\beta}\sigma(-\frac{1}{2} + 0.75010004001\sigma^2) \tag{B.162}$$

$$\alpha_{\pm} = \hat{\beta}\sigma\Big(-\frac{1}{2} + 0.75010004001\sigma^2 \pm \sigma\sqrt{0.00015006002 - 0.00007502\sigma^2}\Big) \tag{B.163}$$

$$\alpha^2 > \hat{\beta}^2\sigma^2(-0.48 + .751\sigma^2)^2 \qquad \text{assuming } \sigma^2 < 1/2 \tag{B.164}$$

$$\hat{\beta}^2 < \frac{1}{\sigma^2(-0.48 + .751\sigma^2)^2}\alpha^2 \tag{B.165}$$

The α root with smaller magnitude provides an upper bound for $\hat{\beta}^2$.

Now consider again $x = [0, 0, \frac{\sigma}{100}, 0]$ and Equation (B.141) with $c = \frac{1}{100}$.

$$v^\top J_{lin} v = 2\hat{\beta}(-1 + c^2)^2 \sigma^6 [\alpha^2(-1 + 3c^2) + 8\hat{\beta}^2 c^4 \sigma^2] \tag{B.166}$$

$$\hat{\beta}^2 \geq \alpha^2 \frac{1 - 3c^2}{8c^4\sigma^2} \tag{B.167}$$

$$\hat{\beta}^2 > \frac{12496250}{\sigma^2}\alpha^2 \tag{B.168}$$

This provides a lower bound for $\hat{\beta}^2$.

$$\hat{\beta}^{hi} - \hat{\beta}_{lo} = \alpha^2\Big(\frac{1}{\sigma^2(-0.48 + .751\sigma^2)^2} - \frac{12496250}{\sigma^2}\Big) \tag{B.169}$$

$$= \frac{\alpha^2}{\sigma^2}\Big(\frac{1}{(-0.48 + .751\sigma^2)^2} - 12496250\Big) \tag{B.170}$$

$$< \frac{\alpha^2}{\sigma^2}\Big(95 - 12496250\Big) \qquad \text{assuming } \sigma^2 < 1/2 \tag{B.171}$$

$$< 0 \tag{B.172}$$

The upper bound we require for $\hat{\beta}$ is greater than the lower bound, therefore, no $\hat{\beta}$ will satisfy quasimonotonicity. □

Lemma 8. *If $\sigma^2 > \frac{1}{2}$, for F_{lin} to be quasimonotone, $\beta + \gamma$ must not be strictly greater than zero, i.e. $\beta + \gamma \not> 0$.*

Proof. For this proof, we make use of the traditional definition of quasimonotonicity. Consider

$$c = \frac{1}{2}\sqrt{\sigma^2 - \frac{1}{2}} \tag{B.173}$$

$$y = [0, 0, c, -c] \tag{B.174}$$

$$x = [0, 0, c, c] \tag{B.175}$$

$$\langle F(y), x - y\rangle = 2cF_b(y) = -2(\beta+\gamma)c^2(1 + 2c^2 + 2c^2 - 2\sigma^2) \tag{B.176}$$

$$= -2(\beta+\gamma)c^2(1 + \sigma^2 - \frac{1}{2} - 2\sigma^2) = -2(\beta+\gamma)c^2(\frac{1}{2} - \sigma^2) \tag{B.177}$$

$$= 2(\beta+\gamma)c^2 \underbrace{(\sigma^2 - \frac{1}{2})}_{>0}, \tag{B.178}$$

$$\langle F(x), x - y\rangle = 2cF_b(x) = -2(\beta+\gamma)c^2 \underbrace{(\sigma^2 - \frac{1}{2})}_{>0}. \tag{B.179}$$

If $(\beta+\gamma) > 0$, then this system is not quasimonotone. In either case, $(\beta+\gamma) \not\geqslant 0$. □

Corollary 2 (F_{lin} requires $\beta+\gamma = 0$ for quasimonotonicity.)**.** *Together, Lemmas 6, 7 and 8 imply that* $(\beta+\gamma)$ *must be* 0 *to satisfy quasimonotonicity.*

Lemma 9. *If* $(\beta+\gamma) = 0$ *and* $\alpha > 0$*, for* F_{lin} *to be quasimonotone,* β *must not be strictly greater than zero, i.e.* $\beta \not\geqslant 0$*.*

Proof. For this proof, we make use of the traditional definition of quasimonotonicity. Consider

$$y = [0, 0, c\sigma, 0], c > 1 \tag{B.180}$$

$$x = [1, 0, \underbrace{(c - \sqrt{c^2-1})}_{>0}\sigma, 0] \tag{B.181}$$

$$\langle F(y), x - y\rangle = F_{w_2}(y) - \sqrt{c^2-1}\sigma F_a(y) = \alpha\sigma^2 \overbrace{(-1 + c^2)}^{>0} \tag{B.182}$$

$$\langle F(x), x - y\rangle = F_{w_2}(x) - \sqrt{c^2-1}\sigma F_a(x) \tag{B.183}$$

$$= \alpha\sigma^2(-1 + (c - \sqrt{c^2-1})^2) - \sqrt{c^2-1}\sigma(8\beta(c - \sqrt{c^2-1})\sigma - 2\alpha(c - \sqrt{c^2-1})\sigma) \tag{B.184}$$

$$= \alpha\sigma^2(-1 + (c - \sqrt{c^2-1})^2 + 2(c - \sqrt{c^2-1})\sqrt{c^2-1}) - 8(c - \sqrt{c^2-1})\sqrt{c^2-1}\sigma^2\beta \tag{B.185}$$

$$= \alpha\sigma^2(-1 + c^2 - 2c\sqrt{c^2-1} + c^2 - 1 + 2c\sqrt{c^2-1} - 2(c^2-1)) - 8(c\sqrt{c^2-1} - c^2 + 1) \tag{B.186}$$

$$= -8\underbrace{(c\sqrt{c^2-1} - c^2 + 1)}_{>0}\sigma^2\beta. \tag{B.187}$$

If $(\beta + \gamma) = 0$ and $\alpha > 0$, then $\beta \leq 0$ for the system to be quasimonotone. □

Lemma 10. *If $(\beta + \gamma) = 0$, for F_{lin} to be quasimonotone, α must not be strictly greater than zero, i.e. $\alpha \not> 0$.*

Proof. We will assume $\alpha > 0$, which by Lemma 9 implies $\beta \leq 0$. This will lead to a contradiction. Consider

$$y = [1, 0, 4\sigma, 0] \tag{B.188}$$

$$x = [0, 0, 2\sigma, 0] \tag{B.189}$$

$$\langle F(y), x - y\rangle = -F_{w_2}(y) - 2\sigma F_a(y) = -15\alpha\sigma^2 - 2\sigma(32\sigma\beta - 8\sigma\alpha) \tag{B.190}$$

$$= \alpha\sigma^2 - 64\beta\sigma^2, \tag{B.191}$$

$$\langle F(x), x - y\rangle = -F_{w_2}(x) - 2\sigma F_a(x) = -3\alpha\sigma^2. \tag{B.192}$$

If $(\beta+\gamma)=0$ and $\alpha>0$ (implies $\beta\leq 0$), then $\langle F(y), x-y\rangle > 0$ and $\langle F(x), x-y\rangle < 0$, which breaks quasimonotonicity. Therefore, $\alpha \not> 0$. □

Lemma 11. *If $(\beta+\gamma)=0$ and $\alpha<0$, for F_{lin} to be quasimonotone, β must not be strictly less than zero, i.e. $\beta \not< 0$.*

Proof. Consider

$$y = [0, 0, c\sigma, 0], c > 1 \tag{B.193}$$

$$x = [-1, 0, (c+\sqrt{c^2-1})\sigma, 0] \tag{B.194}$$

$$\langle F(y), x-y\rangle = -F_{w_2}(y) + \sqrt{c^2-1}\sigma F_a(y) = -\alpha\sigma^2 \overbrace{(-1+c^2)}^{>0} \tag{B.195}$$

$$\langle F(x), x-y\rangle = -F_{w_2}(x) + \sqrt{c^2-1}\sigma F_a(x) \tag{B.196}$$

$$= -\alpha\sigma^2(-1+(c+\sqrt{c^2-1})^2) + \sqrt{c^2-1}\sigma(8\beta(c+\sqrt{c^2-1})\sigma + 2\alpha(c+\sqrt{c^2-1})\sigma) \tag{B.197}$$

$$= \alpha\sigma^2(1-(c+\sqrt{c^2-1})^2 + 2\sqrt{c^2-1}(c+\sqrt{c^2-1})) + 8(c+\sqrt{c^2-1})\sqrt{c^2-1}\sigma^2\beta \tag{B.198}$$

$$= \alpha\sigma^2(1-c^2-c^2+1-2c\sqrt{c^2-1}+2c\sqrt{c^2-1}+2c^2-2) \tag{B.199}$$

$$+ 2\sqrt{c^2-1}(c+\sqrt{c^2-1}))\beta \tag{B.200}$$

$$= 2\underbrace{\sqrt{c^2-1}(c+\sqrt{c^2-1})}_{>0}\beta. \tag{B.201}$$

If $\alpha<0$, then $\beta \geq 0$ to maintain quasimonotonicity. □

Lemma 12. *If $(\beta+\gamma)=0$, for F_{lin} to be quasimonotone, α must not be strictly less than zero, i.e. $\alpha \not< 0$.*

Proof. We will assume $\alpha < 0$, which by 11 implies $\beta \geq 0$. This will lead to a contradiction.

$$y = [-1, 0, c\sigma, 0], c = \frac{1}{4} \tag{B.202}$$

$$x = [0, 0, d\sigma, 0], d = \frac{3}{2} \tag{B.203}$$

$$\langle F(y), x - y\rangle = F_{w_2}(y) + (d-c)\sigma F_a(y) = \alpha\sigma^2 \overbrace{(-1 + c^2)}^{<0} + (d-c)\sigma(8c\sigma\beta + 2c\sigma\alpha) \tag{B.204}$$

$$= \alpha\sigma^2(-1 + c^2 + 2c(d-c)) + 8c(d-c)\sigma^2\beta \tag{B.205}$$

$$= \alpha\sigma^2(-1 - c^2 + 2cd) + 8c(d-c)\sigma^2\beta \tag{B.206}$$

$$= -\frac{5}{16}\alpha\sigma^2 + 40\sigma^2\beta, \tag{B.207}$$

$$\langle F(x), x - y\rangle = F_{w_2}(x) + (d-c)\sigma F_a(x) = \alpha\sigma^2 \underbrace{(-1 + d^2)}_{>0} \tag{B.208}$$

$$= \frac{5}{4}\alpha\sigma^2. \tag{B.209}$$

If $(\beta+\gamma) = 0$ and $\alpha < 0$ (implies $\beta \geq 0$), then $\langle F(y), x-y\rangle > 0$ and $\langle F(x), x-y\rangle < 0$, which breaks quasimonotonicity. Therefore, $\alpha \geq 0$. □

Corollary 3. *Together, Corollary 2 and Lemmas 9-12 imply that α must equal zero for F_{lin} to be quasimonotone.*

Lemma 13. *If $(\beta + \gamma) = 0$ and $\alpha = 0$, for F_{lin} to be quasimonotone, β must not be strictly greater than zero, i.e. $\beta \not> 0$.*

Proof. Consider

$$y = [1, 0, 1, 0] \tag{B.210}$$

$$x = [1, -7, 2, 1] \tag{B.211}$$

$$\langle F(y), x - y\rangle = -7F_{w_1}(y) + F_a(y) + F_b(y) = 8\beta \tag{B.212}$$

$$\langle F(x), x - y\rangle = -7F_{w_1}(x) + F_a(x) + F_b(x) = 16\beta + 4\beta(2-7) \tag{B.213}$$

$$= -4\beta \tag{B.214}$$

If $\beta > 0$, then this system is not quasimonotone. Therefore, $\beta \leq 0$. □

Lemma 14. *If $(\beta+\gamma)=0$ and $\alpha=0$, for F_{lin} to be quasimonotone, β must not be strictly less than zero, i.e. $\beta \not< 0$.*

Proof. Consider

$$y = [1, 0, 2, 0] \tag{B.215}$$

$$x = [1, 1, 1, 1] \tag{B.216}$$

$$\langle F(y), x-y\rangle = F_{w_1}(y) - F_a(y) + F_b(y) = -16\beta \tag{B.217}$$

$$\langle F(x), x-y\rangle = F_{w_1}(x) - F_a(x) + F_b(x) = -8\beta + 12\beta = 4\beta \tag{B.218}$$

If $\beta < 0$, then this system is not quasimonotone. Therefore, $\beta \geq 0$. □

Corollary 4 $((\beta+\gamma)=0, \alpha=0 \Rightarrow \beta=\gamma=0)$**.** *Together, Lemmas 13 and 14 imply that $\beta=0$, which, along with Corollary 2, imply that $\gamma=0$ as well.*

Corollary 5. *[$\alpha=\beta=\gamma=0$] Together, Corollaries 2 and 3, and 4 imply that there is no non-trivial linear combination that induces a quasimonotone LQ-GAN system.*

Corollary 6. *F_{cc}, F_{eg}, F_{con}, and F are not quasimonotone for the LQ-GAN system.*

Proof. These maps are all linear combinations of F, JF and $J^\top F$, therefore, by Corollary 5, they are not quasimonotone for the LQ-GAN system. □

B.12 Analysis of the (w_2, a)-Subsystem

Note that if a map is not quasimonotone for the (w_2, a)-subsystem, then it is not quasimonotone for the full system. This is because an analysis of the (w_2, a)-subsystem is equivalent to an analysis of a subspace of the full system with $w_1 = b = 0$.

Proposition 14. *F is not quasimontone for the (w_2, a)-subsystem. Also, its Jacobian is not Hurwitz.*

Proof.

$$F = \begin{bmatrix} -\sigma^2 + a^2 + b^2 \\ b \\ -2w_2 a \\ -2w_2 b - w_1 \end{bmatrix} \tag{B.219}$$

$$y = [\sigma, 0, 3\sigma, 0] \tag{B.220}$$

$$x = [3\sigma, 0, 5\sigma, 0] \tag{B.221}$$

$$\langle F(y), x - y \rangle = 2\sigma F_{w_2}(y) + 2\sigma F_a(y) = 2\sigma(-\sigma^2 + 9\sigma^2) + 2\sigma(-6\sigma^2) \tag{B.222}$$

$$= 4\sigma^3 \tag{B.223}$$

$$\langle F(x), x - y \rangle = 2\sigma F_{w_2}(x) + 2\sigma F_a(x) = 2\sigma^3(-1 + 25) + 2\sigma^3(-30) \tag{B.224}$$

$$= -12\sigma^3 \tag{B.225}$$

Therefore, F is not quasimonotone.

The Jacobian of F for the (w_2, a)-subsystem is

$$J^{w_2,a} = \begin{bmatrix} 0 & 2a \\ -2a & -2w_2 \end{bmatrix}. \tag{B.226}$$

The trace of $J^{w_2,a}$ is strictly negative for $w_2 > 0$, which implies $J^{w_2,a}$ has an eigenvalue with strictly negative real part. Therefore, $J^{w_2,a}$ is not Hurwitz. □

Proposition 15. *F_{reg} is not quasimonotone for the (w_2, a)-subsystem. Also, its Jacobian is not Hurwitz.*

Proof.

$$F_{reg} = \begin{bmatrix} -\sigma^2 + a^2 + b^2 \\ b \\ -2w_2 a + 4\eta a(-\sigma^2 + a^2 + b^2) \\ -2w_2 b - w_1 + 4\eta b(-\sigma^2 + a^2 + b^2) + 2\eta b \end{bmatrix} \tag{B.227}$$

In order for a system to be quasimonotone, we require condition (A) (among other properties). We will now show that this property is not satisfied for the gradient-regularized system.

Consider the point $x = [w_2, 0, a, 0]$ and let v be defined as follows:

$$v = [2w_2 a^2 + 4\eta a^2(\sigma^2 - a^2), 0, a(a^2 - \sigma^2), 0] \tag{B.228}$$

where v is actually derived by considering the field formed by *crossing the curl* for the 2-D subspace with w_2 and a only.

$F_{reg}^{\top} v$ is 0 as expected.

$$F_{reg}^{\top} v = -2w_2 a^2(\sigma^2 - a^2) - 4\eta a^2(\sigma^2 - a^2)^2 + 2w_2 a^2(\sigma^2 - a^2) + 4\eta a^2(\sigma^2 - a^2)^2 \tag{B.229}$$

$$= 0 \tag{B.230}$$

It suffices to consider the submatrix of the Jacobian corresponding to w_2 and a only when computing $v^{\top} J v$:

$$\frac{1}{2}v^\top J_{reg} = \begin{bmatrix} 2w_2a^2 + 4\eta a^2(\sigma^2 - a^2) & a(a^2 - \sigma^2) \end{bmatrix} \begin{bmatrix} 0 & a \\ -a & -w_2 - 2\eta(\sigma^2 - 3a^2) \end{bmatrix} \tag{B.231}$$

$$= \begin{bmatrix} -a^2(a^2 - \sigma^2) & 2w_2a^3 + 4\eta a^3(\sigma^2 - a^2) - w_2a(a^2 - \sigma^2) + 2\eta a(a^2 - \sigma^2)(3a^2 - \sigma^2) \end{bmatrix} \tag{B.232}$$

$$= \begin{bmatrix} -a^2(a^2 - \sigma^2) & w_2a(a^2 + \sigma^2) + 2\eta a(a^2 - \sigma^2)^2 \end{bmatrix} \tag{B.233}$$

$$\frac{1}{2}v^\top J_{reg}v = \begin{bmatrix} -a^2(a^2 - \sigma^2) & w_2a(a^2 + \sigma^2) + 2\eta a(a^2 - \sigma^2)^2 \end{bmatrix} \begin{bmatrix} 2w_2a^2 + 4\eta a^2(\sigma^2 - a^2) \\ a(a^2 - \sigma^2) \end{bmatrix} \tag{B.234}$$

$$= -2w_2a^4(a^2 - \sigma^2) + 4\eta a^4(a^2 - \sigma^2)^2 + w_2a^2(a^2 + \sigma^2)(a^2 - \sigma^2) + 2\eta a^2(a^2 - \sigma^2)^3 \tag{B.235}$$

$$= w_2a^2(a^2 - \sigma^2)[-2a^2 + (a^2 + \sigma^2)] + 2\eta a^2(a^2 - \sigma^2)^2[2a^2 + (a^2 - \sigma^2)] \tag{B.236}$$

$$= -w_2a^2(a^2 - \sigma^2)^2 + 2\eta a^2(a^2 - \sigma^2)^2(3a^2 - \sigma^2) \tag{B.237}$$

If $w_2 > 0$ and $a < \frac{\sigma}{\sqrt{3}}$, then there isn't an $\eta \geq 0$ that will make this system quasimonotone.

The Jacobian of $F_{reg}^{w_2,a}$ for the (w_2, a)-subsystem is

$$J_{reg}^{w_2,a} = \begin{bmatrix} 0 & 2a \\ -2a & -2w_2 - 4\eta(\sigma^2 - 3a^2) \end{bmatrix}. \tag{B.238}$$

The trace of $J^{w_2,a}$ is strictly negative for $w_2 > 0$ and $a < \sigma/\sqrt{3}$, which implies $J_{reg}^{w_2,a}$ has an eigenvalue with strictly negative real part. Therefore, $J_{reg}^{w_2,a}$ is not Hurwitz. □

Proposition 16. *F_{unr} is not quasimonotone or Hurwitz for the (w_2, a)-subsystem. Also, its Jacobian is not Hurwitz.*

Proof. We consider Unrolled GAN as described in [76]. Some of the necessary arithmetic can be found in the supplementary Mathematica notebook. Define the discriminator's update as

$$w_{2,k+1} = w_{2,k} - \alpha F_{w_2}(w_{2,k}, a_k) = U_k(w_{2,k}), \tag{B.239}$$

where $\alpha > 0$ is a step size, and denote the composition of U, Δk-times as

$$U_k^{\Delta k}(w_{2,k}) = U_k(\cdots(U_k(U_k(w_{2,k}))\cdots) \tag{B.240}$$

where Δk is some positive integer. Then the update for Unrolled GANs is

$$w_{2,k+1} = w_{2,k} - \alpha\frac{\partial V(w_{2,k}, a_k)}{\partial w_2} \tag{B.241}$$

$$a_{k+1} = a_k - \alpha\frac{\partial V(U_k^{\Delta k}(w_{2,k}), a_k)}{\partial a}. \tag{B.242}$$

In the case of the (w_2, a)-subsystem, we can write these unrolled updates out explicitly. Remember $F = [a^2 - \sigma^2, -2aw_2]$, so

$$U_k(w_{1,k}) = w_{2,k} - \alpha(a_k^2 - \sigma^2), \tag{B.243}$$

$$U_k^{\Delta k}(w_{2,k}), a_k) = w_{2,k} - \alpha\Delta k(a_k^2 - \sigma^2). \tag{B.244}$$

Plugging this back in, we find

$$\begin{bmatrix} w_{2,k+1} \\ a_{k+1} \end{bmatrix} = \begin{bmatrix} w_{2,k} \\ a_k \end{bmatrix} - \alpha F_{unr}, \tag{B.245}$$

where the corresponding map is

$$F_{unr} = \begin{bmatrix} a^2 - \sigma^2 \\ 4\alpha\Delta k a^3 - 2a(2\alpha\Delta k\sigma^2 + w_2) \end{bmatrix}. \tag{B.246}$$

We will use the following vector to test condition (A) for quasimonotonicity of F_{unr}:

$$v = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} F_{unr}. \tag{B.247}$$

Computing $v^\top J_{unr} v$ and evaluating at $(w_2 = 1, a = \frac{\sigma^2}{\sqrt{3}})$ gives

$$v^\top J_{unr} v = -\frac{8}{9}\sigma^4 < 0, \tag{B.248}$$

therefore, F_{unr} is not quasimonotone.

If we examine the determinant of J_{unr} and evaluate it at $a = \frac{\sigma}{\sqrt{3}}$, we get

$$Det[J_{unr}]\Big|_{a=\frac{\sigma}{\sqrt{3}}} = -2w_2, \tag{B.249}$$

which is less than zero for positive w_2. Therefore, the Jacobian exhibits negative eigenvalues which means the system is not Hurwitz. □

Proposition 17. *F_{alt} is not quasimonotone or Hurwitz for the (w_2, a)-subsystem. Also, its Jacobian is not Hurwitz.*

Proof. We consider an alternating gradient descent scheme. Some of the necessary arithmetic can be found in the supplementary Mathematica notebook. First, we begin with the case where the discriminator updates first. The updates are

$$w_{2,k+1} = w_{2,k} - \alpha(a_k^2 - \sigma^2) \tag{B.250}$$

$$a_{k+1} = a_k - \alpha(-2a_k w_{2,k+1}) \tag{B.251}$$

$$= a_k - \alpha(-2a_k w_{2,k} + 2a_k \alpha(a_k^2 - \sigma^2)) \tag{B.252}$$

$$= a_k - \alpha(2\alpha a_k^3 - 2a_k(\alpha\sigma^2 + w_{2,k})), \tag{B.253}$$

where $\alpha > 0$ is a step size. The corresponding map is

$$F_{alt} = \begin{bmatrix} a^2 - \sigma^2 \\ 2\alpha a^3 - 2a(\alpha\sigma^2 + w_2) \end{bmatrix}. \tag{B.254}$$

Note the similarity to the Unrolled GAN map Equation (B.246). The maps are equivalent if $\Delta k = 1/2$. Unrolled GANs was shown to be not quasimonotone for any Δk, therefore, F_{alt} is not quasimonotone as well.

If we examine the trace of J_{alt} and evaluate it at $(w_2 = 5\alpha\sigma^2, a = \sigma)$, we get

$$Tr[J_{alt}]\Big|_{(w_2=5\alpha\sigma^2, a=\sigma)} = -6\alpha\sigma^2, \tag{B.255}$$

which is strictly negative. Therefore, the Jacobian exhibits negative eigenvalues which means the system is not Hurwitz.

Now, consider the generator updating first. The updates are

$$w_{2,k+1} = w_{2,k} - \alpha(a_{k+1}^2 - \sigma^2) \tag{B.256}$$

$$= w_{2,k} - \alpha((a_k - \alpha(-2a_k w_{2,k}))^2 - \sigma^2) \tag{B.257}$$

$$a_{k+1} = a_k - \alpha(-2a_k w_{2,k}), \tag{B.258}$$

where the corresponding map is

$$F_{alt'} = \begin{bmatrix} a^2 - \sigma^2 \\ 2\alpha a^3 - 2a(\alpha\sigma^2 + w_2) \end{bmatrix}. \tag{B.259}$$

Testing for condition (A) as before (see Equations (B.246)- (B.248)), we find that

$$v^\top J_{alt'} v = -\frac{1}{2}\sigma^4 w_2 + 4\alpha\sigma^4 w_2^2 + 16c^2\sigma^4 w_2^3 + 16c^3\sigma^4 w_2^4 + 8c^4\sigma^4 w_2^5. \tag{B.260}$$

Using Descartes' Rule of Signs [29], we can determine that this expression has exactly one positive root for w_2. This implies that $v^\top J_{alt'} v$ changes sign locally around this root when varying w_2, which means $v^\top J_{alt'} v < 0$ for some positive w_2. Therefore $F_{alt'}$ is not quasimonotone.

If we examine the determinant of $J_{alt'}$ and evaluate it at $(w_2 = 1, a = \sigma)$, we get

$$Det[J_{alt'}]\Big|_{(w_2=1,a=\sigma)} = -8\alpha(1 + 2\alpha(2 + \alpha(2 + \alpha)))\sigma^2, \tag{B.261}$$

which is less than zero for positive w_2. Therefore, the Jacobian exhibits negative eigenvalues which means the system is not Hurwitz. □

B.12.1 Monotonicity of F_{cc}, F_{eg}, and F_{con} for the (w_2, a)-Subsystem

The following propositions concern the monotonicity of F_{cc}, F_{eg}, and F_{con} for the (w_2, a)-subsystem. The field and Jacobian for F_{lin} will be helpful for proofs of their properties.

$$F_{lin}^{w_2,a} = \begin{bmatrix} \alpha(-\sigma^2 + a^2) + 4(\beta + \gamma)w_2 a^2 \\ 2a(\beta + \gamma)(-\sigma^2 + a^2) + 4(\beta - \gamma)w_2^2 a - 2\alpha w_2 a \end{bmatrix} \tag{B.262}$$

$$J_{lin}^{w_2,a} = \begin{bmatrix} 4(\beta + \gamma)a^2 & 2\alpha a + 8(\beta + \gamma)w_2 a \\ 8(\beta - \gamma)w_2 a - 2\alpha a & 2(\beta + \gamma)(-\sigma^2 + 3a^2) + 4(\beta - \gamma)w_2^2 - 2\alpha w_2 \end{bmatrix} \tag{B.263}$$

Proposition 18. *$F_{con} = F + \beta J^\top F$ is not quasimontone for the (w_2, a)-subsystem. Also, its Jacobian is not Hurwitz.*

Proof. This corresponds to F_{lin} with $\alpha = 1, \beta = \beta, \gamma = 0$. We consider three cases. Let

$$F_{con}^{w_2,a} = \begin{bmatrix} (-\sigma^2 + a^2) + 4\beta w_2 a^2 \\ 2a\beta(-\sigma^2 + a^2) + 4\beta w_2^2 a - 2w_2 a \end{bmatrix}, \tag{B.264}$$

$$J_{con}^{w_2,a} = \begin{bmatrix} 4\beta a^2 & 2a + 8\beta w_2 a \\ 8\beta w_2 a - 2a & 2\beta(-\sigma^2 + 3a^2) + 4\beta w_2^2 - 2w_2 \end{bmatrix}, \tag{B.265}$$

$$v = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} (-\sigma^2 + a^2) + 4\beta w_2 a^2 \\ 2a\beta(-\sigma^2 + a^2) + 4\beta w_2^2 a - 2w_2 a \end{bmatrix} \tag{B.266}$$

$$= \begin{bmatrix} -2a\beta(-\sigma^2 + a^2) - 4\beta w_2^2 a + 2w_2 a \\ (-\sigma^2 + a^2) + 4\beta w_2 a^2 \end{bmatrix}. \tag{B.267}$$

Case 1: Consider $x = [0, 2\sigma]$. Then

$$v^\top J_{con}^{w_2,a} v = 18\beta\sigma^6(11 + 128\beta^2\sigma^2), \tag{B.268}$$

which implies $\beta \geq 0$ for the system to be quasimonotone.

Case 2: Consider $x = [0, 1/2\sigma]$. Then

$$v^\top J_{con}^{w_2,a} v = \frac{9}{32}\beta\sigma^6(-1 + 2\beta^2\sigma^2), \tag{B.269}$$

which, combined with above, implies $\beta \geq \frac{1}{\sqrt{2}\sigma} \approx \frac{0.707}{\sigma}$ for the system to be quasimonotone.

Case 3: Consider $x = [2\sigma, \sigma]$. Then

$$v^\top J_{con}^{w_2,a} v = 64\beta\sigma^6(1 + 4\beta\sigma(1 - 7\beta\sigma)). \tag{B.270}$$

The quantity in parentheses must be positive for this system to be quasimonotone. This quantity is a concave quadratic form with an upper root of $\approx \frac{0.273}{\sigma}$. This implies $\beta \leq \approx \frac{0.273}{\sigma}$ for the system to be quasimonotone.

The last two results cannot be satisfied by a single β, therefore, this system is not quasimonotone.

For completeness, we analyze the limit where the F term is ignored. Consider $a = c\sigma$.

$$v^\top J_{con}^{w_2,a} v = 16c^4(1 + 6c^2 - 119c^4)\sigma^8 \tag{B.271}$$

This is negative for $c = 1$, therefore, this system is not quasimonotone.

The trace of $J_{con}^{w_2,a}$ is strictly negative for $w_2 = 0$ and $a < \sigma/\sqrt{5}$, which implies $J_{con}^{w_2,a}$ has an eigenvalue with strictly negative real part. Therefore, $J_{con}^{w_2,a}$ is not Hurwitz. □

Proposition 19. *$F_{con} = \beta J^\top F$ is not quasimontone for the (w_2, a)-subsystem. Also, its Jacobian is not Hurwitz.*

Proof. This corresponds to F_{lin} with $\alpha = 0, \beta = \beta, \gamma = 0$. We consider two cases.

$$F_{con}^{w_2,a} = \begin{bmatrix} 4\beta w_2 a^2 \\ 2a\beta(-\sigma^2 + a^2) + 4\beta w_2^2 a \end{bmatrix} \tag{B.272}$$

$$J_{con}^{w_2,a} = \begin{bmatrix} 4\beta a^2 & 8\beta w_2 a \\ 8\beta w_2 a & 2\beta(-\sigma^2 + 3a^2) + 4\beta w_2^2 \end{bmatrix} \tag{B.273}$$

$$v = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 4\beta w_2 a^2 \\ 2a\beta(-\sigma^2 + a^2) + 4\beta w_2^2 a \end{bmatrix} \tag{B.274}$$

$$= \begin{bmatrix} -2a\beta(-\sigma^2 + a^2) - 4\beta w_2^2 a \\ 4\beta w_2 a^2 \end{bmatrix} \tag{B.275}$$

Case 2: Consider $x = [0, c\sigma]$. Then

$$v^\top J_{con}^{w_2,a} v = 16\beta^3 c^4 \sigma^8 (c^2 - 1)^2 \tag{B.276}$$

which, for $c \neq 1$, implies $\beta \geq 0$ for the system to be quasimonotone.

Case 2: Consider $x = [2c\sigma, c\sigma]$. Then

$$v^\top J_{con}^{w_2,a} v = -16\beta^3 c^4 \sigma^8 (-1 - 6c^2 + 119c^4) \tag{B.277}$$

which, for $c = 1$, implies $\beta \leq 0$ for the system to be quasimonotone. Combined with above, this implies $\beta = 0$ for the system to be quasimonotone. In conclusion, $\beta J^\top F$ is not quasimonotone.

The trace of $J_{con}^{w_2,a}$ is strictly negative for $w_2 = 0$ and $a < \sigma/\sqrt{5}$, which implies $J_{con}^{w_2,a}$ has an eigenvalue with strictly negative real part. Therefore, $J_{con}^{w_2,a}$ is not Hurwitz. □

Proposition 20. *$F_{eg} = F - \gamma JF$ requires $\gamma \to \infty$ to be pseudomonotone for (w_2, a)-subsystem*

Proof. This corresponds to F_{lin} with $\alpha = 1, \beta = 0, \gamma = \gamma$. We consider two cases.

$$F_{eg}^{w_2,a} = \begin{bmatrix} (-\sigma^2 + a^2) + 4\gamma w_2 a^2 \\ 2a\gamma(-\sigma^2 + a^2) - 4\gamma w_2^2 a - 2w_2 a \end{bmatrix} \tag{B.278}$$

$$J_{eg}^{w_2,a} = \begin{bmatrix} 4\gamma a^2 & 2a + 8\gamma w_2 a \\ -8\gamma w_2 a - 2a & 2\gamma(-\sigma^2 + 3a^2) - 4\gamma w_2^2 - 2w_2 \end{bmatrix} \tag{B.279}$$

$$v = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} (-\sigma^2 + a^2) + 4\gamma w_2 a^2 \\ 2a\gamma(-\sigma^2 + a^2) - 4\gamma w_2^2 a - 2w_2 a \end{bmatrix} \tag{B.280}$$

$$= \begin{bmatrix} -2a\gamma(-\sigma^2 + a^2) + 4\gamma w_2^2 a + 2w_2 a \\ (-\sigma^2 + a^2) + 4\gamma w_2 a^2 \end{bmatrix} \tag{B.281}$$

Case 1: Consider $y = [\sigma, 3\sigma]$ and $x = [3\sigma, 5\sigma]$. Then

$$\langle F(y), x - y \rangle = 2\sigma F_{w_2}(y) + 2\sigma F_a(y) = 2\sigma^3 \Big[8 + 36\gamma\sigma + 48\gamma\sigma - 12\gamma\sigma - 6 \Big] \tag{B.282}$$

$$= 4\sigma^3(1 + 36\sigma\gamma) \tag{B.283}$$

$$\langle F(x), x - y \rangle = 2\sigma F_{w_2}(x) + 2\sigma F_a(x) = 12\sigma^3(-1 + 60\sigma\gamma) \tag{B.284}$$

Then $\gamma \leq -\frac{1}{36\sigma} \approx -\frac{0.027}{\sigma}$ or $\gamma \geq \frac{1}{60\sigma} \approx \frac{0.017}{\sigma}$ for the system to be quasimonotone.

Case 2: Consider $y = [\sigma, 20\sigma]$ and $x = [20\sigma, 5\sigma]$. Then

$$\langle F(y), x - y\rangle = 19\sigma F_{w_2}(y) - 15\sigma F_a(y) = \sigma^3(8181 - 207800\sigma\gamma) \tag{B.285}$$

$$\langle F(x), x - y\rangle = 19\sigma F_{w_2}(x) - 15\sigma F_a(x) = 32\sigma^3(108 + 4825\sigma\gamma) \tag{B.286}$$

Then $\gamma \geq \frac{8181}{207800\sigma} \approx \frac{0.039}{\sigma}$ or $\gamma \geq \frac{108}{4825\sigma} \approx -\frac{0.022}{\sigma}$ for the system to be quasimonotone. The latter condition is more lenient, so the former is unnecessary.

For the system to be quasimonotone in both scenarios, we require that $\gamma \geq \frac{1}{60\sigma}$. This implies γ must be arbitrarily large for small σ. In the limit, the effect of F on the system is negligible. We consider this limit next. □

Proposition 21. *$F_{eg} = -\gamma JF$ is pseudomonotone for (w_2, a)-subsystem.*

Proof. Consider $x = [w_2, c\sigma]$ w.l.o.g.

Note this system is 2-D, therefore, there is only 1 vector v (aside from scaling) that is perpendicular to F.

$$v^\top Jv = 16c^4\sigma^6((-1 + c^2)^2\sigma^2 + 2(1 + c^2)w2^2) \geq 0 \;\forall\; c > 0, w_2 \tag{B.287}$$

$$\langle F(x), x - x^*\rangle = 2c\sigma^2((-1 + c)^2(1 + c)\sigma^2 + 2w_2^2) \geq 0 \;\forall\; c > 0, w_2 \tag{B.288}$$

This satisfies conditions (A) and (C), therefore, this system is pseudomonotone.

□

Proposition 22. *$F_{eg} = F - \gamma JF$ is pseudomonotone for the constrained (w_2, a)-subsystem.*

Proof. We consider $\alpha = 1$ in this case and let the user define a feasible region for which they are confident the equilibrium exists: $w_2 \in [w_2^{\min}, w_2^{\max}]$ and $a \in [a_{\min}, a_{\max}]$—the most important bounds being those on a. We will attempt to find a value for γ that ensures the system is pseudomonotone within this region.

A partially sufficient (and necessary) condition for pseudomonotonicity is the following (see condition (C)).

$$\langle F(x), x - x^* \rangle = 2\gamma\Big(a(a-\sigma)^2(a+\sigma) + 2a\sigma w_2^2\Big) - (a-\sigma)^2 w_2 \geq 0 \tag{B.289}$$

$$\Rightarrow \gamma \geq \frac{\overbrace{(a-\sigma)^2}^{a_1} w_2}{2\Big(\underbrace{a(a-\sigma)^2(a+\sigma)}_{a_0} + \underbrace{2a\sigma}_{a_2} w_2^2\Big)} \tag{B.290}$$

We can find the w_2 that maximizes this equation for a given a by setting the derivative equal to zero and taking the positive root of the resulting quadratic. The denominator of the derivative is non-negative and only zero at equilibrium—this is not a concern because $\langle F(x), x - x^* \rangle = 0$ at equilibrium. Continuing and looking at the numerator of the derivative, we find

$$0 = a_1(a_0 + a_2 d^2) - 2a_1 a_2 d^2 \tag{B.291}$$

$$= a_1(a_0 - a_2 d^2) \tag{B.292}$$

$$d^* = \sqrt{a_0/a_2} \tag{B.293}$$

$$= \sqrt{\frac{(a-\sigma)^2(a+\sigma)}{2\sigma}}. \tag{B.294}$$

If we plug that back into the lower bound for γ, we get

$$\gamma \geq \frac{|a-\sigma|^3\sqrt{a+\sigma}/\sqrt{2\sigma}}{4a(a-\sigma)^2(a+\sigma)} \tag{B.295}$$

$$= \frac{|a-\sigma|}{4\sqrt{2}a\sigma^{1/2}\sqrt{a+\sigma}} \leq \frac{a_{\max}}{4\sqrt{2}a_{\min}^2} \tag{B.296}$$

$$\geq \frac{a_{\max}}{4\sqrt{2}a_{\min}^2} \tag{B.297}$$

The condition above along with the following (see condition (A)) are sufficient to ensure pseudomonotonicity.

$$v^\top Jv = 16a^4\gamma^3((a^2-\sigma^2)^2 + 2w_2^2(a^2+\sigma^2)) \tag{B.298}$$

$$+ 16\gamma^2 w_2 a^2(2\sigma^2 w_2^2 + (a^2-\sigma^2)^2) \tag{B.299}$$

$$+ 2\gamma((a^2-\sigma^2)^2(3a^2-\sigma^2) + w_2^2(8a^2\sigma^2 - 2(a^2-\sigma^2)^2)) \tag{B.300}$$

$$- 2w_2(a^2-\sigma^2)^2 \tag{B.301}$$

If $w_2 \leq 0$, then this quantity is greater than or equal to zero due to the result in equation (B.287), which we have already shown to be greater than zero. Therefore, we focus on $w_2 > 0$. We can divide the analysis into two cases.

Consider $3a^2 \geq \sigma^2$. In this case, all coefficients of γ terms except a γ^1 term and the last term (the constant) are positive. For simplicity, we can find the value for γ such that the first part of the β^2 coefficient is greater than the two negative terms.

$$16w_2a^2\gamma^2(a^2-\sigma^2)^2 - 4\gamma w_2^2(a^2-\sigma^2)^2 - 2w_2(a^2-\sigma^2)^2 \tag{B.302}$$

$$= 2w_2(a^2-\sigma^2)(8a^2\gamma^2 - 2w_2\gamma - 1) \geq 0 \tag{B.303}$$

$$\Rightarrow \gamma \geq \frac{2w_2 + \sqrt{4w_2^2 + 4(8a^2)}}{16a^2} \leq \frac{w_2}{8a^2} + \frac{w_2 + \sqrt{8}a}{8a^2} \tag{B.304}$$

$$\Rightarrow \gamma \geq \frac{w_2^{\max}}{4a_{\min}^2} + \frac{1}{2\sqrt{2}a_{\min}} \tag{B.305}$$

Now consider $3a^2 < \sigma^2$. One of the terms in the γ^1 coefficient is now negative. We will find a value for γ such that the γ^3 term can drown out that negative term.

$$16a^4\gamma^3(a^2-\sigma^2)^2 - 2\gamma(a^2-\sigma^2)^2(\sigma^2-3a^2) \tag{B.306}$$

$$\geq 2\gamma(a^2-\sigma^2)^2(8a^4\gamma^2-\sigma^2) \tag{B.307}$$

$$\Rightarrow \gamma \geq \frac{\sigma}{2\sqrt{2}a^2} \tag{B.308}$$

$$\Rightarrow \gamma \geq \frac{a_{\max}}{2\sqrt{2}a_{\min}^2} \tag{B.309}$$

Combining the results, we have that

$$\gamma \geq \max\left\{\frac{a_{\max}}{2\sqrt{2}a_{\min}^2}, \frac{w_2^{\max}}{4a_{\min}^2} + \frac{1}{2\sqrt{2}a_{\min}}\right\} \tag{B.310}$$

Note this bound is not tight; it is just meant to provide a satisfactory estimate. □

Proposition 23. *$F_{cc} = F + \beta(J^\top - J)F$ requires $\beta \to \infty$ to be pseudomonotone for the (w_2, a)-subsystem.*

Proof. This corresponds to F_{lin} with $\alpha = 1, \gamma = \beta/2, \beta = \beta/2$.

$$F_{cc}^{w_2,a} = \begin{bmatrix} (-\sigma^2 + a^2) + 4\beta w_2 a^2 \\ 2a\beta(-\sigma^2 + a^2) - 2w_2 a \end{bmatrix} \tag{B.311}$$

$$J_{cc}^{w_2,a} = \begin{bmatrix} 4\beta a^2 & 2a + 8\beta w_2 a \\ -2a & 2\beta(-\sigma^2 + 3a^2) - 2w_2 \end{bmatrix} \tag{B.312}$$

$$v = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} (-\sigma^2 + a^2) + 4\beta w_2 a^2 \\ 2a\beta(-\sigma^2 + a^2) - 2w_2 a \end{bmatrix} \tag{B.313}$$

$$= \begin{bmatrix} -2a\beta(-\sigma^2 + a^2) + 2w_2 a \\ (-\sigma^2 + a^2) + 4\beta w_2 a^2 \end{bmatrix} \tag{B.314}$$

Case 1: Consider $x = [0, 2\sigma]$. Then

$$v^\top J_{cc}^{w_2,a} v = 18\beta\sigma^6(11 + 128\beta^2\sigma^2) \tag{B.315}$$

implies that $\beta \geq 0$.

Case 2: Consider $x = [0, 1/2\sigma]$. Then

$$v^\top J_{cc}^{w_2,a} v = \frac{9}{32}\beta\sigma^6(-1 + 2\beta^2\sigma^2) \tag{B.316}$$

this, combined with above, implies that $\beta \geq \frac{1}{\sqrt{2}\sigma}$.

This implies β must be arbitrarily large for small σ. In the limit, the effect of F on the system is negligible. We consider this limit in Subsection 24. □

Proposition 24. *$F_{cc} = (J^\top - J)F$ is pseudomonotone for the (w_2, a)-subsystem.*

Proof.

$$F_{cc}^{w_2,a} = [8w_2 a^2, 4a(a^2 - \sigma^2)] \tag{B.317}$$

$$J_{cc}^{w_2,a} = \begin{bmatrix} 8a^2 & 16w_2 a \\ 0 & 4(3a^2 - \sigma^2) \end{bmatrix} \tag{B.318}$$

Note that the skew part of the Jacobian of F is full rank except at the boundary ($a = 0$), so $F_{cc} = (J^\top - J)F$ maintains the same fixed points. This can be seen by looking at F_{cc} above. We will simply need to constrain a to be greater than 0.

In order for a system to be quasimonotone, we require condition (A) (among other properties). We will now show that this property is satisfied for the (w_2, a)-subsystem.

Case 1: Consider the point $x = [w_2, a]$ and let v be defined as follows:

$$v = F = [-\sigma^2 + a^2, -2w_2 a]^\top. \tag{B.319}$$

$v^\top F_{cc}^{w_2,a}$ is 0 as expected.

$$v^\top F_{cc}^{w_2,a} = -8w_2a^2\sigma^2 + 8w_2a^4 - 8w_2a^4 + 8w_2a^2\sigma^2 \tag{B.320}$$

$$= 0 \tag{B.321}$$

Now, we will compute $v^\top J_{cc}^{w_2,a} v$ to see if it is greater than zero.

$$v^\top J_{cc}^{w_2,a} = \begin{bmatrix} -\sigma^2 + a^2 & -2w_2a \end{bmatrix} \begin{bmatrix} 8a^2 & 16w_2a \\ 0 & 4(3a^2 - \sigma^2) \end{bmatrix} \tag{B.322}$$

$$= \begin{bmatrix} -8\sigma^2a^2 + 8a^4 & 16w_2a(a^2 - \sigma^2) - 8w_2a(3a^2 - \sigma^2) \end{bmatrix} \tag{B.323}$$

$$= \begin{bmatrix} 8a^2(a^2 - \sigma^2) & -8w_2a(a^2 + \sigma^2) \end{bmatrix} \tag{B.324}$$

$$v^\top J_{cc}^{w_2,a} v = \begin{bmatrix} 8a^2(a^2 - \sigma^2) & -8w_2a(a^2 + \sigma^2) \end{bmatrix} \begin{bmatrix} -\sigma^2 + a^2 \\ -2w_2a \end{bmatrix} \tag{B.325}$$

$$= 8a^2(a^2 - \sigma^2)^2 + 16w_2^2a^2(a^2 + \sigma^2) \geq 0 \tag{B.326}$$

In addition to this, proving that $\langle F(x), x - x^* \rangle \geq 0$ is sufficient for proving condition (C).

$$\langle F_{cc}^{w_2,a}(y), y - x^* \rangle = 8w_2a^2w_2 + 4a(a^2 - \sigma^2)(a - \sigma) \geq 0 \tag{B.327}$$

The last two terms of the sum are always the same sign due to the square function being "monotone" and the fact that a is constrained to be non-negative. Therefore, F_{cc} is pseudomonotone. □

Proposition 25. *$F_{cc} = F + \beta(J^\top - J)F$ is pseudomonotone for the constrained (w_2, a)-subsystem.*

Proof. We consider $\alpha = 1$ in this case and let the user define a feasible region for which they are confident the equilibrium exists: $w_2 \in [w_2^{\min}, w_2^{\max}]$ and $a \in [a_{\min}, a_{\max}]$—the most important bounds being those on a. We will attempt to find a value for β that ensures the system is pseudomonotone within this region.

A partially sufficient (and necessary) condition for pseudomonotonicity is the following (see condition (C)).

$$\langle F(x), x - x^* \rangle = 2\beta\Big(a(a - \sigma)^2(a + \sigma) + 2a^2w_2^2\Big) - (a - \sigma)^2w_2 \geq 0 \tag{B.328}$$

$$\Rightarrow \beta \geq \frac{\overbrace{(a - \sigma)^2}^{a_1} w_2}{2\Big(\underbrace{a(a - \sigma)^2(a + \sigma)}_{a_0} + \underbrace{2a^2}_{a_2} w_2^2\Big)} \tag{B.329}$$

We can find the w_2 that maximizes this equation for a given a by setting the derivative equal to zero and taking the positive root of the resulting quadratic. The denominator of the derivative is non-negative and only zero at equilibrium—this is not a concern because $\langle F(x), x - x^* \rangle = 0$ at equilibrium. Continuing and looking at the numerator of the derivative, we find

$$0 = a_1(a_0 + a_2 d^2) - 2a_1 a_2 d^2 \tag{B.330}$$

$$= a_1(a_0 - a_2 d^2) \tag{B.331}$$

$$d^* = \sqrt{a_0/a_2} \tag{B.332}$$

$$= \sqrt{\frac{(a-\sigma)^2(a+\sigma)}{2a}}. \tag{B.333}$$

If we plug that back into the lower bound for β, we get

$$\beta \geq \frac{|a-\sigma|^3\sqrt{a+\sigma}/\sqrt{2a}}{4a(a-\sigma)^2(a+\sigma)} \tag{B.334}$$

$$= \frac{|a-\sigma|}{4\sqrt{2}a^{3/2}\sqrt{a+\sigma}} \leq \frac{a_{\max}}{4\sqrt{2}a_{\min}^2} \tag{B.335}$$

$$\geq \frac{a_{\max}}{4\sqrt{2}a_{\min}^2} \tag{B.336}$$

The condition above along with the following (see condition (A)) are sufficient to ensure pseudomonotonicity.

$$v^\top J v = 16a^4\beta^3((a^2-\sigma^2)^2 + 2w_2^2(a^2+\sigma^2)) \tag{B.337}$$

$$+ 32\beta^2 w_2^3 a^4 \tag{B.338}$$

$$+ 2\beta((a^2-\sigma^2)^2(3a^2-\sigma^2) + 8a^4 w_2^2) \tag{B.339}$$

$$- 2w_2(a^2-\sigma^2)^2 \tag{B.340}$$

If $w_2 \leq 0$, then this quantity is greater than or equal to zero due to the result in equation (B.326), which we have already shown to be greater than zero. Therefore, we focus on $w_2 > 0$. We can divide the analysis into two cases.

Consider $3a^2 \geq \sigma^2$. In this case, all coefficients of β terms except the last term (the constant) are positive. For simplicity, we can find the value for β such that the first part of the β^3 coefficient is greater than the last term (the constant).

$$16a^4\beta^3(a^2-\sigma^2)^2 - 2w_2(a^2-\sigma^2)^2 \geq 0 \tag{B.341}$$

$$\Rightarrow \beta \geq \frac{1}{2}\Big(\frac{w_2^{\max}}{a_{\min}^4}\Big)^{1/3} \tag{B.342}$$

Now consider $3a^2 < \sigma^2$. One of the terms in the β^1 coefficient is now negative. We will find a value for β such that the β^3 term can drown out the two negative terms.

$$16a^4\beta^3(a^2-\sigma^2)^2 - 2\beta(a^2-\sigma^2)^2(\sigma^2-3a^2) - 2w_2(a^2-\sigma^2)^2 \tag{B.343}$$

$$= \frac{(a^2-\sigma^2)^2}{16a^4}\Big[\beta^3 - \frac{2(\sigma^2-3a^2)}{16a^4}\beta - \frac{2w_2}{16a^4}\Big] \tag{B.344}$$

$$\geq \frac{(a^2-\sigma^2)^2}{16a^4}\Big[\beta^3 - \underbrace{\frac{\sigma^2}{8a^4}}_{a_0}\beta - \underbrace{\frac{w_2}{8a^4}}_{a_1}\Big] \tag{B.345}$$

$$= \frac{(a^2-\sigma^2)^2}{16a^4}\Big[3a_0^{1/2}a_1^{2/3} + 2a_1a_2^{2/3}\Big] \text{ for } \beta = a_0^{1/2} + a_1^{1/3} \tag{B.346}$$

$$\geq 0 \tag{B.347}$$

$$\Rightarrow \beta \geq a_0^{1/2} + a_1^{1/3} = \frac{1}{2\sqrt{2}}\frac{a_{\max}}{a_{\min}^2} + \frac{1}{2}\Big(\frac{w_2^{\max}}{a_{\min}^4}\Big)^{1/3} \tag{B.348}$$

This last lower bound is the greatest of the three, so it suffices to set β greater than this value to ensure the system is pseudomonotone within the given feasible region. □

Proposition 26. *F_{lin} is not monotone for the (w_2, a)-subsystem (before scaling).*

Proof. Let $F_{lin}^{w_2,a}$ be defined as follows:

$$(\alpha I + \beta J^\top - \gamma J)F = \begin{bmatrix} \alpha & -2(\beta+\gamma)a \\ 2(\beta+\gamma)a & \alpha - 2(\beta-\gamma)w_2 \end{bmatrix} \begin{bmatrix} -\sigma^2 + a^2 \\ -2w_2 a \end{bmatrix} \tag{B.349}$$

$$= \begin{bmatrix} \alpha(-\sigma^2 + a^2) + 4(\beta+\gamma)w_2 a^2 \\ 2a(\beta+\gamma)(-\sigma^2+a^2) + 4(\beta-\gamma)w_2^2 a - 2\alpha w_2 a. \end{bmatrix} \tag{B.350}$$

Its Jacobian is then

$$J_{lin}^{w_2,a} = \begin{bmatrix} 4(\beta+\gamma)a^2 & 2\alpha a + 8(\beta+\gamma)w_2 a \\ 8(\beta-\gamma)w_2 a - 2\alpha a & 2(\beta+\gamma)(-\sigma^2+3a^2) + 4(\beta-\gamma)w_2^2 - 2\alpha w_2 \end{bmatrix} \tag{B.351}$$

$$J_{sym} = \begin{bmatrix} 4(\beta+\gamma)a^2 & 8\beta w_2 a \\ 8\beta w_2 a & 2(\beta+\gamma)(-\sigma^2+3a^2) + 4(\beta-\gamma)w_2^2 - 2\alpha w_2 \end{bmatrix} \tag{B.352}$$

The trace of the symmetrized Jacobian must be non-negative to ensure monotonicity because a negative trace implies the existence of a negative eigenvalue:

$$Tr = 2(\beta+\gamma)(-\sigma^2+5a^2) + 4(\beta-\gamma)w_2^2 - 2\alpha w_2 \leq 0 \ \forall a < \frac{\sigma}{\sqrt{5}}, w_2 = 0. \tag{B.353}$$

Assume $\beta + \gamma > 0$. If $a < \sigma/\sqrt{5}$ and $w_2 = 0$, then the trace is less than zero.

Assume $\beta + \gamma < 0$. If $a > \sigma/\sqrt{5}$ and $w_2 = 0$, then the trace is less than zero.

Assume $\gamma = -\beta$. Then

$$Tr = 8\beta w_2^2 - 2\alpha w_2 = 2w_2(4\beta w_2 - \alpha). \tag{B.354}$$

If $w_2 < 0$, then $\beta \leq \frac{\alpha}{4w_2}$. If $w_2 > 0$, then $\beta \geq \frac{\alpha}{4w_2}$. Therefore, $\beta = \frac{\alpha}{4w_2}$, however, β and α are constants while w_2 is a variable. Therefore, α and β must equal zero to satisfy this for all w_2 proving that no monotone linear combination exists. □

Proposition 27. *F_{lin} is not Hurwitz for the (w_2, a)-subsystem.*

Proof. Consider $J_{lin}^{w_2,a}$ at $w_2 = 0$.

$$J_{lin}^{w_2,a} = \begin{bmatrix} 4(\beta+\gamma)a^2 & 2\alpha a \\ -2\alpha a & 2(\beta+\gamma)(-\sigma^2+3a^2) \end{bmatrix} \tag{B.355}$$

$$Tr = 2(\beta+\gamma)(5a^2-\sigma^2) \tag{B.356}$$

$$Det = 8(\beta+\gamma)^2(-\sigma^2+3a^2)a^2 + 4\alpha^2a^2 \tag{B.357}$$

If $\beta+\gamma<0$, then $a>\sigma/\sqrt{5}$ implies the existence of an eigenvalue with negative real part. If $\beta+\gamma>0$, then $a<\sigma/\sqrt{5}$ implies the existence of an eigenvalue with negative real part. If $\beta+\gamma=0$, then the real part is zero. □

Proposition 28. *There exists an $F_{lin'}$ family after scaling by $1/4a^2$ that exhibits strict-monotonicity.*

Proof. If we consider the same linear combinations above, but divide F by $4a^2$, we can obtain a family of monotone fields (see Mathematica notebook).

The trace of the corresponding symmetrized Jacobian is

$$Tr = \frac{(\beta+\gamma)(3a^2+\sigma^2)+\alpha w_2+2(\gamma-\beta)w_2^2}{2a^2}. \tag{B.358}$$

For constant β and γ and nonzero α, there exists a value for w_2 that will force the trace to be negative, therefore α must be zero. Note that γ must be greater than or equal to β to ensure that the trace cannot be made negative in the limit as w_2^2 grows to infinity.

Case 1: Consider the case where $\beta=\gamma$. Then for any fixed β, γ, and nonzero α,

$$w_2 = -(3a^2 + \sigma^2)\frac{\beta + \gamma}{\alpha} - \alpha \tag{B.359}$$

will cause the trace to be negative.

Case 2: Otherwise, consider solving the quadratic form for w_2 when $\beta + \gamma > 0$:

$$w_2 = \frac{-\alpha \pm \sqrt{\alpha^2 - 8(3a^2 + \sigma^2)(\gamma - \beta)(\beta + \gamma)}}{4(\gamma - \beta)}. \tag{B.360}$$

For the trace to be non-negative, we need the leading coefficient of the quadratic to be positive, i.e., $\gamma - \beta > 0$. We also need there to be at most 1 real root, meaning the square root must be non-positive. If $\beta + \gamma > 0$, then setting a and σ using the following formula will force the root to be positive:

$$3a^2 + \sigma^2 < \frac{\alpha^2}{8(\gamma - \beta)(\beta + \gamma)} \tag{B.361}$$

For example, set $a = \sigma$, and then set σ and w_2 as follows to force the trace to be negative:

$$\sigma = \frac{3}{4}\frac{\alpha}{\sqrt{32(\gamma - \beta)(\beta + \gamma)}}, \tag{B.362}$$

$$w_2 = -\frac{\alpha}{4(\gamma - \beta)}. \tag{B.363}$$

Case 3: If $\beta + \gamma \leq 0$, then the root is necessarily positive. Therefore, α must be set to zero.

The field and Jacobian are now wieldy enough to state:

$$F_{lin'}^{w_2,a} = (\beta+\gamma)\left[w_2, \tfrac{(a-\sigma)(a+\sigma)}{2a} - 4\left(\tfrac{\gamma-\beta}{\beta+\gamma}\right)\left(\tfrac{w_2^2}{a}\right)\right], \tag{B.364}$$

and

$$J_{lin'}^{w_2,a} = (\beta+\gamma)\begin{bmatrix} 1 & 0 \\ -2\left(\tfrac{\gamma-\beta}{\beta+\gamma}\right)\left(\tfrac{w_2}{a}\right) & \tfrac{1}{2} + \tfrac{\sigma^2}{2a^2} + \left(\tfrac{\gamma-\beta}{\beta+\gamma}\right)\left(\tfrac{w_2^2}{a^2}\right) \end{bmatrix}. \tag{B.365}$$

The trace is now

$$Tr = \frac{(\beta+\gamma)(3a^2+\sigma^2) + 2(\gamma-\beta)w_2^2}{2a^2}, \tag{B.366}$$

and is non-negative as long as both $\beta+\gamma \geq 0$ and $\gamma-\beta \geq 0$.

The determinant is

$$Det = \frac{(\beta+\gamma)^2(a^2+\sigma^2) + 4(\gamma-\beta)\beta w_2^2}{2a^2}, \tag{B.367}$$

which is non-negative as long as, in addition to the previous conditions, we have $\beta \geq 0$. The trace and determinant are both strictly positive if $\beta+\gamma > 0$.

In summary, $F_{lin'}^{w_2,a}$ is strictly-monotone, i.e., $J_{lin'}^{w_2,a} \succ 0$, if $\gamma \geq \beta \geq 0$ and $\gamma > 0$. □

Corollary 7. *The $F_{lin'}$ family includes $F_{eg'}$ ($\gamma = \gamma, \beta = 0$) and $F_{cc'}$ ($\gamma = \beta$). By Proposition 28, $F_{eg'}$ and $F_{cc'}$ are at least strictly-monotone.*

Proposition 29. *$F_{cc'}^{w_2,a}$ is $1/2$-strongly monotone and $F_{eg'}^{w_2,a}$ is only strictly-monotone.*

Proof. We will look at both maps individually.

Case $F_{cc'}^{w_2,a}$: The eigenvalues of $J_{cc'}^{w_2,a}$ are $\lambda_1 = 1$ and $\lambda_2 = \frac{1}{2}\Big(1 + \frac{\sigma^2}{a^2}\Big)$. Therefore, $J_{cc'}^{w_2,a} \succeq \frac{1}{2}$ and $F_{cc'}^{w_2,a}$ is $1/2$-strongly monotone.

Case $F_{eg'}^{w_2,a}$: The eigenvalues of a 2×2 matrix can be written in terms of the trace and determinant as

$$\lambda_{1,2} = \frac{Tr \pm \sqrt{Tr^2 - 4Det}}{2} \tag{B.368}$$

$$= \frac{Tr}{2}\Big(1 \pm \sqrt{1 - \frac{4Det}{Tr^2}}\Big). \tag{B.369}$$

Therefore, if the term $\frac{4Det}{Tr^2}$ can be made arbitrarily small, then one of the eigenvalues can made arbitrarily close to zero. On the other hand, if this quantity has a finite lower bound, then the eigenvalues are lower bounded as a constant multiple of the trace.

The trace and determinant of $J_{eg'}^{w_2,a}$ are

$$Tr = \frac{1}{2}\Big(3 + \frac{\sigma^2}{a^2}\Big) + \frac{w_2^2}{a^2} \tag{B.370}$$

$$Det = \frac{1}{2}\Big(1 + \frac{\sigma^2}{a^2}\Big). \tag{B.371}$$

and the quantity, Q, described is

$$Q = \frac{8a^2(a^2 + \sigma^2)}{(3a^2 + \sigma^2 + 2w_2^2)^2}. \tag{B.372}$$

This term can be made arbitrarily small as w_2 goes to infinity. To be more rigorous, let $a = \sigma = 1$ so that $Tr = 2 + w_2^2$ and $Det = 1$. Then

$$\lambda_{1,2} = \frac{1}{2}(w_2^2+2)\Big(1-\sqrt{1-\frac{4}{w_2^2+2}}\Big) \tag{B.373}$$

$$= \frac{1}{2}\frac{\overbrace{\Big(1-\sqrt{1-\frac{4}{w_2^2+2}}\Big)}^{top}}{\underbrace{(w_2^2+2)^{-1}}_{bot}}. \tag{B.374}$$

An application of L'Hopital's rule shows that

$$\lim_{w_2\to\infty} \frac{\partial top/\partial w_2}{\partial bot/\partial w_2} = \frac{4}{(w_2^2+2)\sqrt{1-\frac{4}{(w_2^2+2)^2}}} = 0. \tag{B.375}$$

The minimum eigenvalue only approaches zero in the limit, so $F_{eg'}^{w_2,a}$ is strictly-monotone. □

Claim 3. *$F_{cc'}^{w_2,a}$ is the gradient of the following convex function: $f_{cc'}^{w_2,a} = w_2^2 + 1/2\Big((a^2-\sigma^2) - \sigma^2\log(\frac{a^2}{\sigma^2}\Big)$.*

Proof. The Jacobian of $F_{cc'}^{w_2,a}$ is symmetric and PSD, therefore it is the Hessian of some convex function. We can integrate $F_{cc'}^{w_2,a}$ to arrive at a convex function (with arbitrary constant). Integrating $F_{cc'}^{w_2,a}$ results in the following:

$$f_{cc'}^{w_2,a} = w2^2 + 1/2\Big((a^2-\sigma^2) - \sigma^2\log\big(\frac{a^2}{\sigma^2}\big)\Big) \tag{B.376}$$

Note that $f_{cc'}^{w_2,a}$ must be convex along the subspace with $w_2 = 0$ as well, which implies that

$$g(a||\sigma) = 1/2\Big((a^2-\sigma^2) - \sigma^2\log\big(\frac{a^2}{\sigma^2}\big)\Big) \tag{B.377}$$

is convex as well. This function is of individual interest because it may serve as a preferred alternative to KL-divergence. □

B.13 Progressive Learning of LQ-GAN

Here, we consider the stochastic setting where the GAN is trained using samples from $p(y)$ and $p(z)$. There are two ways to learn both the mean and variance of a distribution using $F_{cc}^{w_2,a}$. One is to first learn the mean to a high degree of accuracy, then stop learning the mean and start learning the variance. The other is to keep learning the mean with an appropriate weighting of the two systems to maintain stability. We discuss the former option first.

Proposition 30. *Assume all $y \sim p(y)$ lie in $[y_{low}, y_{hi}]$. After $k > \left(\frac{y_{hi}-y_{low}}{-|\mu|+\sqrt{\mu^2+d\sigma^2}}\right)^2 \log[\frac{\sqrt{2}}{\delta^{1/2}}]$ iterations, with probability, $1-\delta$, the (w_1, b)-subsystem can be "shut-off" and the (w_2, a)-subsystem safely "turned-on" resulting in a $1/2$-strongly-monotone $F_{cc'}^{w_2,a}$.*

Proof. We begin by observing the symmetrized Jacobian of $F_{cc'}^{w_2,a}$:

$$J_{cc'}^{w_2,a} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{a^2-b^2+\mu^2+\sigma^2}{2a^2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{G}{2a^2}+\frac{1}{2} \end{bmatrix}, \tag{B.378}$$

where $G = \mu^2+\sigma^2-b^2$. In order for $F_{cc'}^{w_2,a}$ to be strongly monotone, we require $G \geq 0$. In other words, the square of the generator's estimate of the mean, b_k, learned from training the (w_1, b)-subsystem needs to be less than or equal to $\mu^2+\sigma^2$.

Assume we are using $F_{cc'}^{w_1,b}$ with step size $\rho_k = \frac{1}{k+1}$ to train the (w_1, b)-subsystem. Note that this was shown equivalent to the standard running mean in Proposition 13. Therefore, $b_k = Z = \frac{1}{K}\sum_{i=1}^{k} y_i$. Also, $\mathbb{E}[Z] = \mu$. Then, using Hoeffding's inequality, we find

$$Pr(|Z - \mathbb{E}[Z]| \geq t) \leq 2e^{-\frac{2kt^2}{(y_{hi}-y_{low})^2}} \tag{B.379}$$

$$\Rightarrow Pr(|b_k - \mu| < t) \geq 1 - 2e^{-\frac{2kt^2}{(y_{hi}-y_{low})^2}} = 1-\delta \tag{B.380}$$

Assume $|b_k - \mu| < t$ and introduce a scalar: $0 < d < 1$. Remember, we require $b_k^2 < \mu^2 + \sigma^2$. And we know $\mu - t < b_k < \mu + t$ which implies

$$b_k^2 < \mu^2 + \underbrace{t^2 + 2|\mu|t}_{=d\sigma^2} < \mu^2 + \sigma^2 \tag{B.381}$$

$$\Rightarrow 0 = t^2 + 2|\mu|t - d\sigma^2, t > 0 \tag{B.382}$$

This expression has two roots for t, one positive and one negative. $|b_k - \mu|$ can only be upper bounded by a positive number, so we select the positive root.

$$t_{roots} = \frac{-2|\mu| \pm \sqrt{4\mu^2 + d4\sigma^2}}{2} \tag{B.383}$$

$$= -|\mu| \pm \sqrt{\mu^2 + d\sigma^2} \tag{B.384}$$

$$t_+ = -|\mu| + \sqrt{\mu^2 + d\sigma^2} \tag{B.385}$$

Plugging t_+ back into equation (B.381) for t, we find that

$$G = \mu^2 + \sigma^2 - b_k^2 > (1 - d)\sigma^2. \tag{B.386}$$

Rearranging (B.380) and plugging in t, we can derive the number of iterations required:

$$k > \Big(\frac{y_{hi} - y_{low}}{-|\mu| + \sqrt{\mu^2 + d\sigma^2}}\Big)^2 \log\big[\frac{\sqrt{2}}{\delta^{1/2}}\big]. \tag{B.387}$$

If we assume $p(y) \sim \mathcal{N}(\mu, \sigma^2)$ and use a Chernoff bound, we find

$$Pr(|b_k - \mu| < t) \geq 1 - 2e^{-\frac{kt^2}{\sigma^2}} = 1 - \delta \tag{B.388}$$

$$k > \Big(\frac{\sigma}{-|\mu| + \sqrt{\mu^2 + d\sigma^2}}\Big)^2 \log\big[\frac{2}{\delta}\big]. \tag{B.389}$$

The number of samples needed to maintain stability of the system grows as the true mean μ deviates from zero. This is not an artifact of the concentration inequalities (it

occurs with both), but of the parameterization of the LQ-GAN—the samples are not mean centered before being passed to the quadratic discriminator, i.e., $w_2 y^2$ rather than $w_2(y-\mu)^2$. This may explain why batch norm is so helpful (almost required) in stabilizing training.

□

Proposition 31. *Assume all $y \sim p(y)$ lie in $[y_{low}, y_{hi}]$. After $k > \left(\frac{y_{hi}-y_{low}}{-|\mu|+\sqrt{\mu^2+d\sigma^2}}\right)^2 \log[\frac{\sqrt{2}}{\delta^{1/2}}]$ iterations, with probability, $1-\delta$, the (w_1, b)-subsystem can be up-weighted and the (w_2, a)-subsystem "turned-on", resulting in a strictly-monotone LQ-GAN.*

Proof. As before, assume we are running $F_{cc}^{w_1,b}$ on the (w_1, b)-subsystem and $F_{cc'}^{w_2,a}$ on the (w_2, a)-subsystem. Also, multiply $F_{cc}^{w_1,b}$ by $e > 0$, i.e., increase the learning rate by e or divide the learning rate of $F_{cc'}^{w_2,a}$ by e. The full symmetrized Jacobian of this system is:

$$J_{cc'} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & \frac{a^2-b^2+\mu^2+\sigma^2}{2a^2} & \frac{b}{2a} \\ 0 & 0 & \frac{b}{2a} & e \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & \frac{G}{2a^2}+\frac{1}{2} & \frac{b}{2a} \\ 0 & 0 & \frac{b}{2a} & e \end{bmatrix} \tag{B.390}$$

The upper left 2×2 block of this matrix is positive definite. In order to show the whole matrix is positive definite, it suffices to prove the lower right block is positive definite. The trace and determinant of that block are

$$Tr_{ab} = 1/2 + e + \frac{G}{2a^2} \tag{B.391}$$

$$Det_{ab} = \frac{2e(a^2+G)-b^2}{4a^2}. \tag{B.392}$$

where $G = \mu^2 + \sigma^2 - b^2$ as before. We need $G \geq 0$ for $Tr_{ab} > 0$ (for $\lim_{a\to 0+}$) and $2eG \geq b^2$ for $Det > 0$. As before, Hoeffding's inequality says k iterations are required

for an accurate estimate of the mean (see Equation (B.387)). And as before, we find that $G = (1-d)\sigma^2$. We will focus on the determinant condition here. Let

$$G = (1-d)\sigma^2 \geq \frac{b^2}{2e} \tag{B.393}$$

$$\Rightarrow e \geq \frac{b^2}{2(1-d)\sigma^2} \tag{B.394}$$

$$\Rightarrow e \geq \frac{\mu^2 + d\sigma^2}{2(1-d)\sigma^2} \tag{B.395}$$

$$\text{or } \Rightarrow d \leq 1 - \frac{b^2}{2e\sigma^2} \tag{B.396}$$

$$\Rightarrow d \leq 1 - \frac{\mu^2 + d\sigma^2}{2e\sigma^2}. \tag{B.397}$$

More simply, let $d = 1/2$. Then set $e > \frac{\mu_{\max}^2}{\sigma_{\min}^2} + \frac{1}{2}$. This ensures the trace and determinant are both strictly positive which implies that the resulting system is at least strictly monotone.

We can show that this system is not strongly-monotone by upper bounding the minimum eigenvalue. To ease the analysis, let $H = 2eG - b^2$ and note that $H < 2e\sigma^2$ (see Equation (B.393)), i.e., H is finite. This allows us to upper bound the determinant, in turn, upper bounding the minimum eigenvalue. The determinant simplifies to

$$Det_{ab} = \frac{e}{2} + \frac{H}{4a^2}. \tag{B.398}$$

The minimum eigenvalue is upper bounded as follows:

$$\lambda_{min} = \frac{1}{2}\Big(Tr - \sqrt{Tr^2 - 4Det}\Big) \tag{B.399}$$

$$= \frac{1}{2}\Big(1/2 + e + \frac{G}{2a^2} - \sqrt{(1/2 + e + \frac{G}{2a^2})^2 - 2e - \frac{H}{a^2}}\Big) \tag{B.400}$$

$$\lim_{a \to 0^+} \lambda_{min} = \frac{1}{2}\Big(1/2 + e + \frac{G}{2a^2} - \sqrt{(1/2 + e + \frac{G}{2a^2})^2}\Big) = 0 \tag{B.401}$$

As the system continues learning a more accurate mean (iterations, k, is increasing), d is effectively decreasing towards zero. In the limit $\lim_{d\to 0+} e \geq \frac{\mu^2}{2\sigma^2}$.

Given, $[y_{low}, y_{hi}]$, we can set $\mu_{max} = \max(|y_{low}|, |y_{hi}|)$. Also, note that if the distribution is known to support ϵ balls at the ends of the specified interval, $[y_{low}, y_{hi}]$, with some nonzero probabilities, P_{low} and P_{hi}, then we can lower bound the variance as well. Specifically, let $P_{low} = \frac{\epsilon}{2}\big(p(y_{low}) + p(y_{low} + \epsilon)\big)$ and $P_{hi} = \frac{\epsilon}{2}\big(p(y_{hi}) + p(y_{hi} - \epsilon)\big)$. Then

$$\sigma^2 = \mathbb{E}[(y-\mu)^2] = \int_{y_{low}}^{y_{hi}} p(y)(y-\mu)^2 dy \tag{B.402}$$

$$\geq \int_{y_{low}}^{y_{low}+\epsilon} p(y)(y-\mu)^2 dy + \int_{y_{hi}+\epsilon}^{y_{hi}} p(y)(y-\mu)^2 dy \tag{B.403}$$

$$= \frac{\epsilon}{2}\big(p(y_{low}) + p(y_{low}+\epsilon)\big)(y_{low}-\mu)^2 \tag{B.404}$$

$$+ \frac{\epsilon}{2}\big(p(y_{hi}) + p(y_{hi}-\epsilon)\big)(y_{hi}-\mu)^2 + \mathcal{O}(\epsilon^2) \tag{B.405}$$

$$\approx P_{low}(y_{low}-\mu)^2 + P_{hi}(y_{hi}-\mu)^2 \tag{B.406}$$

$$\geq P_{low}P_{hi}(y_{hi}-y_{low})^2 = \sigma^2_{\min}. \tag{B.407}$$

□

B.14 Analysis of the (W_2, A)-Subsystem for the N-d LQ-GAN

Let A be a lower triangular matrix with positive diagonal—A represents the generator's guess at the square root of Σ.

Proposition 32. *The 2-D LQ-GAN is not quasimonotone for F_{cc} or F_{eg} with or without scaling.*

Proof. We will show that this system fails condition (A). Please refer to the Mathematica notebook for our derivations of these results.

Define the following skew symmetric matrix.

$$K = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \tag{B.408}$$

Let $v_{cc} = KF_{cc}$ and $v_{eg} = KF_{eg}$. Similarly, with scaling, let $v_{cc'} = KF_{cc'}$ and $v_{eg'} = KF_{eg'}$. Let

$$\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 100 \end{bmatrix} \tag{B.409}$$

$$x = \begin{bmatrix} W11 \\ W12 \\ W22 \\ A11 \\ A22 \\ A21 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0.1 \\ 0.1 \end{bmatrix} \tag{B.410}$$

Then

$$v_{cc}^{\top} J_{cc}(x) v_{cc}^{\top}\Big|_x = -189684 < 0 \tag{B.411}$$

$$v_{eg}^{\top} J_{eg}(x) v_{eg}^{\top}\Big|_x = -189684 < 0 \tag{B.412}$$

$$v_{cc'}^{\top} J_{cc'}(x) v_{cc'}^{\top}\Big|_x = -2.95426 \cdot 10^9 < 0 \tag{B.413}$$

$$v_{eg'}^{\top} J_{eg'}(x) v_{eg'}^{\top}\Big|_x = -2.95426 \cdot 10^9 < 0 \tag{B.414}$$

This implies that neither system is quasimonotone (with, cc'/eg', or without, cc/eg, scaling). □

Proposition 33. *The 2-D LQ-GAN with W_{11} and A_{11} already learned, i.e., $W_{11} = 0$ and $A_{11} = A_{11}^*$, is not quasimonotone for F_{cc} or F_{eg}.*

Proof. We will show that this system fails condition (A). Please refer to the Mathematica notebook for our derivations of these results.

Define the following skew symmetric matrix.

$$K = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{B.415}$$

Let $v_{cc} = KF_{cc}$ and $v_{eg} = KF_{eg}$. Let

$$\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 100 \end{bmatrix} \tag{B.416}$$

$$x = \begin{bmatrix} W12 \\ W22 \\ A22 \\ A21 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0.1 \\ 0.1 \end{bmatrix} \tag{B.417}$$

Then

$$v_{cc}^{\top} J_{cc}(x) v_{cc}^{\top}\Big|_x = -189684 < 0 \tag{B.418}$$

$$v_{eg}^{\top} J_{eg}(x) v_{eg}^{\top}\Big|_x = -189684 < 0 \tag{B.419}$$

This implies that neither system is quasimonotone. □

Proposition 34. *The 3-D LQ-GAN with the diagonal of A already learned, i.e., $A_{ii} = A_{ii}^*$, is not quasimonotone for F_{cc} or F_{eg} with or without scaling.*

Proof. We will show that this system fails condition (A). Please refer to the Mathematica notebook for our derivations of these results.

Define the following skew symmetric matrix.

$$K = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \tag{B.420}$$

Let $v_{cc} = KF_{cc}$ and $v_{eg} = KF_{eg}$. Let

$$\Sigma = \begin{bmatrix} 0.2 & 0.15 & 0.5 \\ 0.15 & 0.9 & 0.8 \\ 0.5 & 0.8 & 2 \end{bmatrix} \tag{B.421}$$

$$x = \begin{bmatrix} W12 \\ W13 \\ W23 \\ A21 \\ A31 \\ A32 \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 0.1 \\ 0.2 \\ -0.5 \end{bmatrix} \tag{B.422}$$

Then

$$v_{cc}^{\top} J_{cc}(x) v_{cc}^{\top}\Big|_x = -1024.26 < 0 \tag{B.423}$$

$$v_{eg}^{\top} J_{eg}(x) v_{eg}^{\top}\Big|_x = -242766 < 0 \tag{B.424}$$

This implies that neither system is quasimonotone. □

Proposition 35. *The N-d LQ-GAN with all but a single row of A fixed is strictly-monotone for F_{cc}, F_{eg}, and F_{con}.*

Proof. First, note that the Cholesky decomposition of Σ, denoted by A^*, obeys the follow equation:

$$0 = \Sigma_{ij} - \sum_{d=1}^{i} A_{id}^* A_{jd}^* \tag{B.425}$$

where $i < j$. Σ is symmetric, so Σ_{ji} can be recovered as Σ_{ij}. This allows us to remove 1 degree of freedom from the system by defining the diagonal term in a single row of A in terms of the other entries in the row:

$$A_{ii} = \sqrt{\Sigma_{ii} - \sum_{d=1}^{i-1} A_{id}^2} \tag{B.426}$$

where as before A_{ii} must be greater than zero. We assume that Σ_{ii} has already been learned by *Crossing-the-Curl* as described in the main body. The condition $A_{ii} > 0$ can be ensured by constraining $\sum_{d=1}^{i-1} A_{id}^2 \leq \Sigma_{ii} - \epsilon$ with $\epsilon \ll 1$—this can be achieved with a simple ball projection.

Consider learning a single row of A, specifically A_{Ni} with $i < N$; A_{NN} is recovered as discussed above and $A_{N,i>N} = 0$ by definition of the Cholesky decomposition. We will also set all $W_{2ij} = W_{2ji}$ equal to zero except where i xor j equals N. This has the effect of fixing parts of the system irrelevant for solving the Nth row of A. For ease of exposition, we will drop the "2" subscript of W_2 in what follows.

We will begin by writing down the map for the entire system and then simplifying using the constraints and assumptions discussed above:

$$F_{W_2} = AA^\top - \Sigma \tag{B.427}$$

$$= \begin{bmatrix} A_{11}^2 & A_{11}A_{21} & A_{11}A_{31} & \cdots \\ A_{11}A_{21} & A_{21}^2 + A_{22}^2 & A_{21}A_{31} + A_{22}A_{32} & \cdots \\ A_{11}A_{31} & A_{21}A_{31} + A_{22}A_{32} & A_{31}^2 + A_{32}^2 + A_{33}^2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} - \begin{bmatrix} S_{11} & S_{12} & S_{13} & \cdots \\ S_{12} & S_{22} & S_{23} & \cdots \\ S_{13} & S_{23} & S_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{B.428}$$

$$F_A = -2W_2A \tag{B.429}$$

$$= -2 \begin{bmatrix} A_{11}W_{11} + A_{21}W_{12} + A_{31}W_{13} + \cdots & A_{22}W_{12} + A_{32}W_{13} + \cdots & A_{33}W_{13} + \cdots & \cdots \\ A_{11}W_{12} + A_{21}W_{22} + A_{31}W_{23} + \cdots & A_{22}W_{22} + A_{32}W_{23} + \cdots & A_{33}W_{23} + \cdots & \cdots \\ A_{11}W_{13} + A_{21}W_{23} + A_{31}W_{33} + \cdots & A_{22}W_{23} + A_{32}W_{33} + \cdots & A_{33}W_{33} + \cdots & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{B.430}$$

We are only interested in learning the Nth row of A. Take $N = 3$ for example. Notice that the 3rd row of A, $A_{3:}$, only contains the following W_2 terms: W_{13}, W_{23}. The rest are set to zero as mentioned earlier. The reason for this will become apparent soon. We fix all other entries to zero to highlight the relevant subsystem below:

$$F_{W_2} = AA^\top - \Sigma \tag{B.431}$$

$$= \begin{bmatrix} 0 & 0 & A_{11}A_{31} - S_{13} & \cdots \\ 0 & 0 & A_{21}A_{31} + A_{22}A_{32} - S_{23} & \cdots \\ A_{11}A_{31} - S_{13} & A_{21}A_{31} + A_{22}A_{32} - S_{23} & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{B.432}$$

$$F_{W_{i<N}} = 2\big(\sum_{d \le i} A_{id}A_{Nd} - S_{iN}\big) \tag{B.433}$$

$$F_A = -2W_2A \tag{B.434}$$

$$= -2 \begin{bmatrix} 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots \\ A_{11}W_{13} + A_{21}W_{23} & A_{22}W_{23} & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{B.435}$$

$$F_{A_{N>i}} = -2\big(\sum_{d<N} A_{di}W_{dN}\big). \tag{B.436}$$

Notice that the map F_{W_2} is zero only if Equation (B.425) is satisfied for Σ_{iN} and $W_{dN} = 0$ for all $d < N$. Therefore, setting all other entries of W_2 as prescribed simplified the system, while maintaining the correct fixed point.

In order to determine the monotonicity of this system, we need to compute the Jacobian of $F = [F_{W_2}; F_A]$:

$$J = \begin{bmatrix} \frac{\partial F_{W_{i<d}}}{\partial W_{k<d}} & \frac{\partial F_{W_{i<d}}}{\partial A_{d>k}} \\ \frac{\partial F_{A_{d>i}}}{\partial W_{k<d}} & \frac{\partial F_{A_{d>i}}}{\partial A_{d>k}} \end{bmatrix} \tag{B.437}$$

$$= -2 \begin{bmatrix} (d-1) \times 0 & -A_{i \geq k} \\ A_{k \geq i} & (d-1) \times 0 \end{bmatrix} \tag{B.438}$$

$$= -2 \begin{bmatrix} 0 & 0 & -A_{11} & 0 \\ 0 & 0 & -A_{21} & -A_{22} \\ A_{11} & A_{21} & 0 & 0 \\ 0 & A_{22} & 0 & 0 \end{bmatrix} \text{ for } N = 3 \tag{B.439}$$

$$= -2 \begin{bmatrix} 0 & -A_{:d-1} \\ A_{:d-1}^\top & 0 \end{bmatrix} \tag{B.440}$$

which is skew-symmetric and constant with respect to the variables being learned: $W_{2,i<N}$ and $A_{N>i}$. Therefore, $J + J^\top = 0$ is PSD, which implies F is monotone. The fact that J is constant along with Proposition 11 imply that $F_{cc} = F_{eg} = F_{con} = -JF$ are also monotone:

$$F_{cc} = F_{eg} = F_{con} = 2 \begin{bmatrix} -A_{:d-1} F_{A_{d>i}} \\ A_{:d-1}^\top F_{W_{i<d}} \end{bmatrix}. \tag{B.441}$$

Note that the component of F_{cc} corresponding to the dynamics of A, is independent of W_2. This means the dynamics are now decoupled from W_2 and can be run separately. By inspecting the symmetrized Jacobian of F_{cc} we can show that it is a block matrix composed of positive definite matrices:

$$J_{sym} = \frac{1}{4}(J - J^\top)^\top (J - J^\top) \tag{B.442}$$

$$= J^\top J = -JJ \tag{B.443}$$

$$= \begin{bmatrix} A_{:d-1}A_{:d-1}^\top & 0 \\ 0 & A_{:d-1}^\top A_{:d-1} \end{bmatrix}. \tag{B.444}$$

$A_{:d-1}A_{:d-1}^\top$ is positive definite because A is constrained to be of Cholesky form. Moreover, the eigenvalues of $A_{:d-1}^\top A_{:d-1}$ are the same as $A_{:d-1}A_{:d-1}^\top$, therefore both blocks are positive definite. This implies the entire matrix J_{sym} is positive definite which means $F_{cc} = F_{eg} = F_{con}$ are strictly monotone. Note that we do not require $A_{:d-1} = A_{:d-1}^*$ for strict monotonicity. In practice, the system will actually be both strongly-monotone and smooth. This is because A is constrained with a projection onto a ball and the diagonal of A is restricted to be larger than ϵ. These two conditions guarantee a nonzero, finite minimum and maximum value for the eigenvalues of $A_{:d-1}A_{:d-1}^\top$—the minimum corresponds to strong-monotonicity and the maximum corresponds to smoothness. □

Unlike the (w_2, a)-subsystem where monotonicity depends on the accuracy of the learned mean, this system is monotone as long as $A_{:d-1}$ is PSD which is guaranteed from the form we have prescribed to A. This result suggests learning the rows of A in succession, and each subsystem is guaranteed to be strictly monotone. Note that the variance, i.e., diagonal of Σ, will be slightly off the true value if the mean, μ, is not first learned perfectly. The learned A will then be slightly off the true A^* and errors will compound, but still not affect monotonicity. The subsystems corresponding to each row of A can be revisited to learn the entries of A more accurately. Permuting the dimensions of x such that the dimensions corresponding to highest variance are learned first may ensure subsystems with maximal *strong*-monotonicity. We leave a detailed examination to future research.

B.15 An $\mathcal{O}(N/k)$ Algorithm for LQ-GAN

Here we present pseudocode for solving the stochastic LQ-GAN. The maps corresponding to learning the mean and variance by *Crossing-the-Curl* are both strongly convex and can therefore be solved with a simple projected gradient method. We argued in the previous section that the map associated with learning the covariance terms is strongly-monotone and smooth, not only strictly monotone. In practice, we found that a projected Extragradient algorithm [55] gave better results. The full procedure is outlined in Algorithm 5. Replace sample estimates with the true μ and Σ for the deterministic LQ-GAN.

Algorithm 8 *Crossing-the-Curl* for LQ-GAN

Input: Sampling distribution $p(y)$, max iterations K, batch size B, lower bound on variance $\sigma_{\min}$

(1) Learn Mean

$\mu_0 = [0, \dots, 0]^\top$

for all $k = 1, 2, \dots, K$ **do**

$\hat{\mu} = \frac{1}{B}\sum_{s=1}^{B}(y_s \sim p(y))$

$\mu_k = \frac{k}{k+1}\mu_{k-1} + \frac{1}{k+1}\hat{\mu}$, i.e., $\mu_k = \mu_{k-1} - \rho_k F^b_{cc}$ with step size $\rho_k = \frac{1}{k+1}$

end for

(2) Learn Variance

$\sigma_0 = [1, \dots, 1]^\top$

for all $k = 1, 2, \dots, K$ **do**

$\hat{\sigma}^2 = \frac{1}{B}\sum_{s=1}^{B}[(y_s \sim p(y)) - \mu_K]^2$

$F^a_{cc'} = (\sigma_k^2 - \hat{\sigma}^2)/(2\sigma_k)$

$\sigma_k = \texttt{clip}(\sigma_{k-1} - \frac{1}{k+1}F^a_{cc'}, \sigma_{\min}, \infty)$

end for

(3) Learn Covariance

$A_0 = LT(I_N)$, i.e., lower triangular part of identity matrix

$A_{0,11} = \sigma_{K,1}$

for all $d = 2, \dots, N$ **do**

for all $k = 1, 2, \dots, K$ **do**

$y_s \sim p(y),\ s = 1, \dots, B$

$\hat{\Sigma} = \frac{1}{B}\sum_{s=1}^{B}(y_s - \mu_K)^\top(y_s - \mu_K)$

$F_{W_{i<d}} = 2\big(\sum_{j\le i} A_{k-1,ij}A_{k-1,dj} - \hat{\Sigma}_{id}\big)$

$F^A_{cc} = A^\top_{k-1,:d-1}F_{W_{i<d}}$ where $A_{k-1,:d-1}$ refers to the top left $d-1 \times d-1$ block of A_{k-1}

$\hat{A}_{k,d:} = A_{k-1,d:} - \frac{1}{k+1}F^A_{cc}$ where $A_{k-1,d:}$ refers to the dth row of A_k excluding the diagonal

if $\sum_j \hat{A}^2_{k,dj} > \sigma^2_{K,d} - \sigma^2_{\min}$ **then**

$\hat{A}_{k,dj} = \hat{A}_{k,dj} \cdot \sigma_{K,d}/\sqrt{\sum_j \hat{A}^2_{k,dj} + \sigma^2_{\min}}$

end if

$F_{W_{i<d}} = 2\big(\sum_{j\le i} A_{k-1,ij}\hat{A}_{k,dj} - \hat{\Sigma}_{id}\big)$

$F^A_{cc} = A^\top_{k-1,:d-1}F_{W_{i<d}}$ where $A_{k-1,:d-1}$ refers to the top left $d-1 \times d-1$ block of A_{k-1}

$A_{k,d:} = A_{k-1,d:} - \frac{1}{k+1}F^A_{cc}$ where $A_{k-1,d:}$ refers to the dth row of A_k excluding the diagonal

if $\sum_j A^2_{k,dj} > \sigma^2_{K,d} - \sigma^2_{\min}$ **then**

$A_{k,dj} = A_{k,dj} \cdot \sigma_{K,d}/\sqrt{\sum_j A^2_{k,dj} + \sigma^2_{\min}}$

end if

end for

$A_{K,dd} = \sqrt{\sigma^2_{K,d} - \sum_j A^2_{K,dj}}$

end for

B.15.1 Convergence Rate

As mentioned above, the maps for learning the mean and variance are both strongly convex which implies a $\mathcal{O}(1/k)$ stochastic convergence rate for each, the sum of which is still $\mathcal{O}(1/k)$.

In practice, the maps for learning each row of A are strongly-monotone and smooth (see last paragraph of proof of Proposition B.14) which implies a $\mathcal{O}(1/k)$ stochastic convergence rate for each as well. Because this technique consists of $N+1$ steps for learning the full N-d LQ-GAN, it requires $\hat{k} = Nk$ iterations which, in total, implies a $\mathcal{O}(N/k)$ stochastic convergence rate.

Hidden within this analysis is the fact that each iteration of learning the mean and variance is $\mathcal{O}(N)$ in terms of time-complexity and each iteration for learning each row of A is $\mathcal{O}(N^2)$, therefore this entire procedure is $\mathcal{O}(N^3/k)$ in terms of FLOPS. This is expected as the complexity of a Cholesky decomposition to compute $A = \Sigma^{1/2}$ is also $\mathcal{O}(N^3)$. Note that unlike the complexity of computing F each iteration which can be mitigated with parallel computation, the sequential nature of the stagewise procedure cannot be amortized which is why we report a $\mathcal{O}(N/k)$ convergence rate and not $\mathcal{O}(1/k)$.

Another subtle point is that the LQ-GAN is locally monotone about the equilibrium. Recall from Theorem D.1 on p.26 in the work of Nagarajan and Kolter [2017] that the Jacobian at the equilibrium is of the following form (remember our definition for the Jacobian is the negative of theirs):

$$J = \begin{bmatrix} J_{DD} & J_{DG} \\ -J_{DG}^\top & 0 \end{bmatrix} \tag{B.445}$$

where J_{DD} is positive definite. The symmetrized Jacobian is then

$$J_{sym} = \frac{1}{2}(J + J^\top) = \begin{bmatrix} J_{DD} & 0 \\ 0 & 0 \end{bmatrix} \succeq 0. \tag{B.446}$$

This implies F is monotone where $F = [\nabla V_{A,b}; -\nabla V_{W_2,w_1}]$. Therefore, we can use stagewise procedure in Algorithm 5 to converge to a local neighborhood about the equilibrium, constrain the system to this neighborhood with a projection (which will guarantee smoothness of the map), and then continue with an Extragradient method applied to the full system. The local convergence rate will still be $\mathcal{O}(1/k)$ with $\mathcal{O}(N^3)$ iteration complexity due to the matrix multiplications required in computing F (see Proposition 9).

B.16 Deep Learning Specifications and Results

We also experimented on common neural-net driven tasks. We tested F_{lin} with $(\alpha, \beta, \gamma) = (1, 10, 10^{-4})$ on a mixture of Gaussians and $(\alpha, \beta, \gamma) = (1, 10, 0.1)$ on CIFAR10 against F_{con}, i.e., $(\alpha, \beta, \gamma) = (1, 10, 0)$. Introducing a small $-JF$ term can help accelerate training (see Figure B.1).

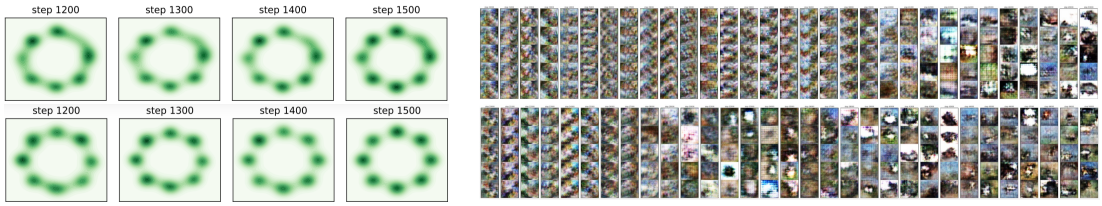


Figure B.1: F_{con} (top) vs F_{lin} (bottom) on a mixture of Gaussians (left) and CIFAR10 (right). Each column of images corresponds to an epoch with epochs increasing left to right.

B.16.1 Images at End of Training for Mixture of Gaussians

See Figure B.2.

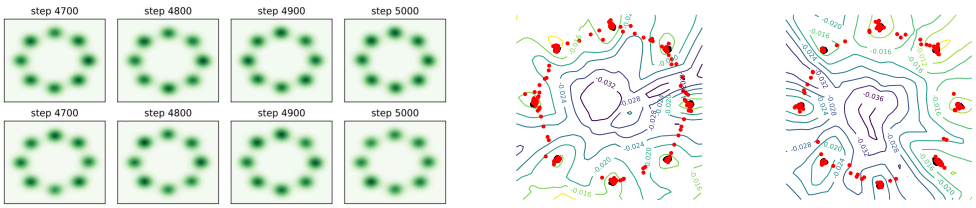


Figure B.2: F_{con} (top row) vs F_{lin} (bottom row) on a mixture of Gaussians. Contour plots of discriminator along with samples in red shown for F_{con} (left) and F_{lin} (right).

B.16.2 Mixture of Gaussians Network Architectures

Both the generator and discriminator are fully connected neural networks. The relevant hyperparameters for setting up the GAN are itemized below.

- batch size 512
- divergence Wasserstein
- disc optim Adam
- disc learning rate 0.001
- disc n hidden 16
- disc n layer 4
- disc nonlinearity ReLU
- gen optim Adam
- gen learning rate 0.001
- gen n hidden 16
- gen n layer 4
- gen nonlinearity ReLU

- betas [0.5, 0.999]
- epsilon 1e-08
- max iter 5001
- z dim 16
- x dim 2

F_{con} was used with $\beta = 1.0$ and F_{lin} was used with $(\alpha, \beta, \gamma) = (1.0, 1.0, 0.001)$.

B.16.3 Images at End of Training for CIFAR10

See Figure B.3.

Figure B.3: F_{con} (top row) vs F_{lin} (bottom row) on CIFAR10. Images generated at final iteration shown for F_{con} (left) and F_{lin} (right).

B.16.4 CIFAR10 Network Architectures

Both the generator and discriminator are convolutional neural networks; we copied the architectures used in [74]. The generator consists of a linear layer, followed by 4 deconvolution layers (5×5 kernel, 2×2 stride, leaky ReLU, 64 hidden channels), followed by a final linear layer with a tanh nonlinearity. The discriminator consists of 4 convolution layers (5×5 kernel, 2×2 stride, leaky ReLU, 64 hidden channels) followed by a linear layer. The relevant hyperparameters for setting up the GAN are itemized below.

- batch size 64
- divergence JS
- disc optim RMSprop
- disc learning rate 0.0001
- gen optim RMSprop
- gen learning rate 0.0001
- betas [0.5, 0.999]
- epsilon 1e-08
- max iter 150001
- z dim 256
- x dim 1024

F_{con} was used with $\beta = 10.0$ and F_{lin} was used with $(\alpha, \beta, \gamma) = (1.0, 10.0, 0.0001)$.

APPENDIX C

GENERATIVE MULTI-ADVERSARIAL NETWORKS

This appendix supplements Chapter 4 with additional experiments and descriptions of their architectures.

C.0.1 Accelerated Convergence and Reduced Variance

See Figures C.1, C.2, C.3, and C.4.

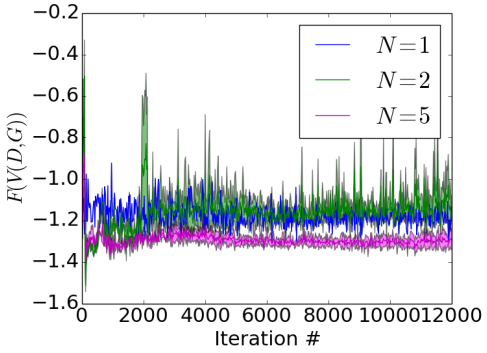


Figure C.1: Generator objective, F, averaged over 5 training runs on CelebA. Increasing N (# of D) accelerates convergence of F to steady state (solid line) and reduces its variance, σ^2 (filled shadow $\pm 1\sigma$). Figure C.2 provides alternative evidence of GMAN-0's accelerated convergence.

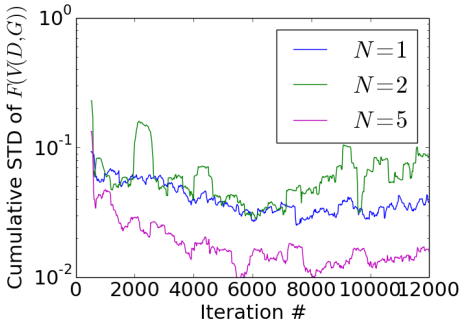


Figure C.2: *Stdev*, σ, of the generator objective over a sliding window of 500 iterations. Lower values indicate a more steady-state. GMAN-0 with $N = 5$ achieves steady-state at ≈2x speed of GAN ($N = 1$). Note Figure C.1's filled shadows reveal *stdev* of F over runs, while this plot shows *stdev* over time.

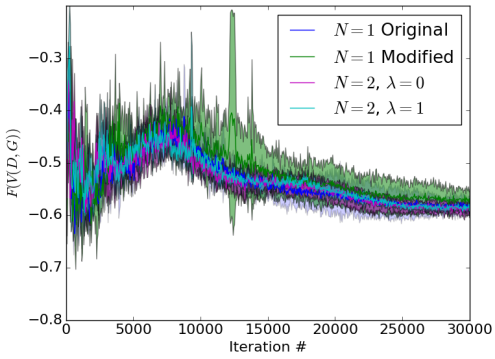


Figure C.3: Generator objective, F, averaged over 5 training runs on CIFAR-10. Increasing N (# of D) accelerates convergence of F to steady state (solid line) and reduces its variance, σ^2 (filled shadow $\pm 1\sigma$). Figure C.4 provides alternative evidence of GMAN-0's accelerated convergence.

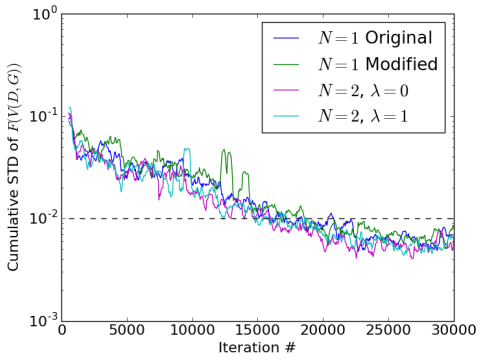


Figure C.4: *Stdev*, σ, of the generator objective over a sliding window of 500 iterations. Lower values indicate a more steady-state. GMAN-0 with $N = 5$ achieves steady-state at ≈2x speed of GAN ($N = 1$). Note Figure C.3's filled shadows reveal *stdev* of F over runs, while this plot shows *stdev* over time.

C.0.2 Additional GMAM Tables

See Tables C.1, C.2, C.3, C.4, C.5. Increasing the number of discriminators from 2 to 5 on CIFAR-10 significantly improves scores over the standard GAN both in terms of the GMAM metric and Inception scores.

	Score	Variant	GMAN*	GMAN-1	GAN	GMAN-0	GMAN-max	mod-GAN
Better→	**0.184**	GMAN*	-	−0.007	−0.040	−0.020	−0.028	−0.089
	0.067	GMAN-1	0.007	-	−0.008	−0.008	−0.021	−0.037
	0.030	GAN	0.040	0.008	-	0.002	−0.018	−0.058
	0.005	GMAN-0	0.020	0.008	0.002	-	−0.013	−0.018
	−0.091	GMAN-max	0.028	0.021	0.018	0.013	-	−0.011
	−0.213	mod-GAN	0.089	0.037	0.058	0.018	0.011	-

Table C.1: Pairwise GMAM metric means for select models on MNIST. For each column, a positive GMAM indicates better performance relative to the row opponent; negative implies worse. Scores are obtained by summing each column.

	Score	Variant	GMAN-0	GMAN-1	GMAN*	mod-GAN
Better→	**0.172**	GMAN-0	-	−0.022	−0.062	−0.088
	0.050	GMAN-1	0.022	-	0.006	−0.078
	−0.055	GMAN*	0.062	−0.006	-	−0.001
	−0.167	mod-GAN	0.088	0.078	0.001	-

Table C.2: Pairwise GMAM metric means for select models on CIFAR-10. For each column, a positive GMAM indicates better performance relative to the row opponent; negative implies worse. Scores are obtained by summing each column. GMAN variants were trained with **two** discriminators.

	GMAN-0	GMAN-1	mod-GAN	GMAN*
Score	**5.878** ± 0.193	5.765 ± 0.168	5.738 ± 0.176	5.539 ± 0.099

Table C.3: Inception score means with standard deviations for select models on CIFAR-10. Higher scores are better. GMAN variants were trained with **two** discriminators.

	Score	Variant	GMAN-0	GMAN*	GMAN-1	mod-GAN
Better→	**0.180**	GMAN-0	-	−0.008	−0.041	−0.132
	0.122	GMAN*	0.008	-	−0.038	−0.092
	0.010	GMAN-1	0.041	0.038	-	−0.089
	−0.313	mod-GAN	0.132	0.092	0.089	-

Table C.4: Pairwise GMAM metric means for select models on CIFAR-10. For each column, a positive GMAM indicates better performance relative to the row opponent; negative implies worse. Scores are obtained by summing each column. GMAN variants were trained with **five** discriminators.

	GMAN-1	GMAN-0	GMAN*	mod-GAN
Score	**6.001** ± 0.194	5.957 ± 0.135	5.955 ± 0.153	5.738 ± 0.176

Table C.5: Inception score means with standard deviations for select models on CIFAR-10. Higher scores are better. GMAN variants were trained with **five** discriminators.

C.0.3 Generated Images

See Figures C.5 and C.6.

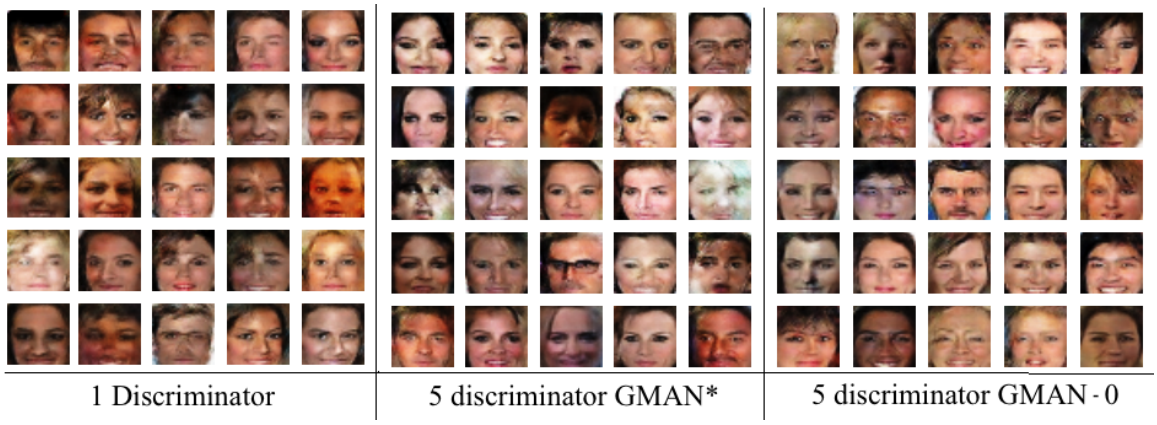


Figure C.5: Sample of images generated on CelebA cropped dataset.

C.1 Related Work

A GAN framework with two discriminators appeared in the work of Yoo et al. [2016], however, it is applicable only in a semi-supervised case where a label can be assigned to subsets of the dataset (e.g., $\mathcal{X} = \{\mathcal{X}_1 = \text{Domain } 1, \mathcal{X}_2 = \text{Domain } 2, \ldots\}$). In contrast, our framework applies to an unsupervised scenario where an obvious partition of the dataset is unknown. Furthermore, extending GMAN to the semi-supervised domain-adaptation scenario would suggest multiple discriminators per domain, therefore our line of research is strictly orthogonal to that of their multi-domain discriminator approach. Also, note that assigning a discriminator to each domain is akin to prescribing a new discriminator to each value of a conditional variable in conditional GANs [77]. In this case, we interpret GMAN as introducing multiple conditional discriminators and not a discriminator for each of the possibly exponentially many conditional labels.

In Section 4.4.3, we describe an approach to customize adversarial training to better suit the development of the generator. An approach with similar conceptual

Figure C.6: Sample of images generated by GMAN-0 on CIFAR dataset.

underpinnings was described in the work of Ravanbakhsh et al. [2016], however, similar to the above, it is only admissible in a semi-supervised scenario whereas our applies to the unsupervised case.

C.2 *Softmax* Representability

Let $softmax(V_i) = \hat{V} \in [\min_{V_i}, \max_{V_i}]$. Also let $a = \arg\min_i V_i$, $b = \arg\max_i V_i$, and $\mathcal{V}(t) = V((1-t)D_a + tD_b)$ so that $\mathcal{V}(0) = V_a$ and $\mathcal{V}(1) = V_b$. The *softmax* and minimax objective $V(D_i, G)$ are both continuous in their inputs, so by the *intermediate value theorem*, we have that $\exists\, \hat{t} \in [0,1]$ *s.t.* $\mathcal{V}(\hat{t}) = \hat{V}$, which implies $\exists\, \hat{D} \in \mathcal{D}$ *s.t.* $V(\hat{D}, G) = \hat{V}$. This result implies that the *softmax* (and any other continuous substitute) can be interpreted as returning $V(\hat{D}, G)$ for some $\hat{D}$ selected by computing an another, unknown function over the space of the discriminators. Note that this result holds even if $\hat{D}$ is not representable by the architecture chosen for the dicriminator's neural network.

C.3 Unconstrained Optimization

To convert GMAN* minimax formulation to an unconstrained minimax formulation, we introduce an auxiliary variable, Λ, define $\lambda(\Lambda) = \log(1 + e^{\Lambda})$, and let the generator minimize over $\Lambda \in \mathbb{R}$ instead.

C.4 Boosting with *AdaBoost.OL*

Note that the online AdaBoost algorithm [14] does not require knowledge of the weak learner's slight edge over random guessing ($P(\text{correct prediction}) = 0.5 + \gamma \in (0, 0.5]$), and in fact, allows $\gamma < 0$. This is theoretically crucial because our weak learners are deep nets with unknown, possibly negative, γ's.

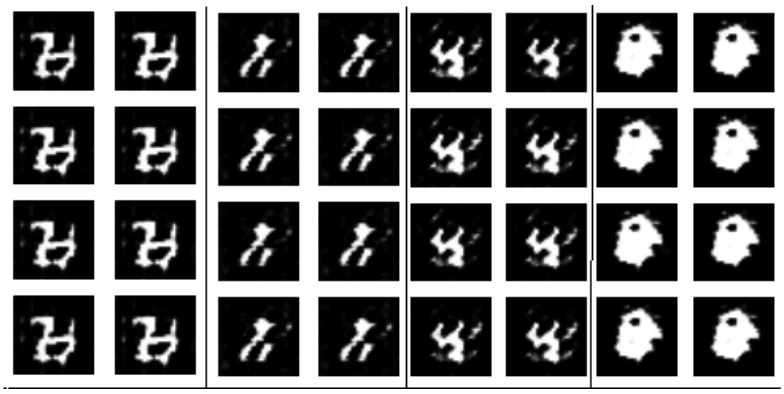


Figure C.7: Example of images generated across four independent runs on MNIST with boosting.

C.5 Experimental Setup

All the experiments were conducted using architecture similar to DCGAN [92]. We use convolutional transpose layers [121] for the generator G and strided convolutions for the discriminator D except for the input of the generator and the last layer of the discriminator.

We use the single step gradient method as in the work of Nowozin et al. [2016].

Batch normalization [48] was used in each of the generator layers. The different discriminators were trained with varying dropout rates from $[0.3, 0.7]$.

Variations in the discriminators were effected in two ways. We varied the architecture by varying the number of filters in the discriminator layers (reduced by factors of 2, 4 and so on), as well as varying dropout rates. Secondly we also decorrelated the samples that the disriminators were training on by splitting the minibatch across the discriminators.

Specifics for the MNIST architecture and training are:

- Generator latent variables $z \sim \mathcal{U}\left(-1, 1\right)^{100}$
- Generator convolution transpose layers as follows:

  $(4, 4, 128)\,, (8, 8, 64)\,, (16, 16, 32)\,, (32, 32, 1)$

- Base Discriminator architecture: $(32, 32, 1), (16, 16, 32), (8, 8, 64), (4, 4, 128)$.
- Variants have either convolution 3 $(4, 4, 128)$ removed or all the filter sizes are divided by 2 or 4. That is, $(32, 32, 1), (16, 16, 16), (8, 8, 32), (4, 4, 64)$ or $(32, 32, 1), (16, 16, 8), (8, 8, 16), (4, 4, 32)$.
- ReLu activations for all the hidden units. Tanh activation at the output units of the generator. Sigmoid at the output of the Discriminator.
- Optimization was done using Adam [59] with a learning rate of 2×10^{-4} and $\beta_1 = 0.5$.
- MNIST was trained for 20 epochs with a minibatch of size 100.
- CelebA and CIFAR were trained over 24000 iterations with a minibatch of size 100 each iteration.

The code was written in Tensorflow [1] and run on Nvidia GTX 980 GPUs.

APPENDIX D

ANALYZING NON-MONOTONE GAMES

This appendix supplements Chapter 5.

D.1 BoA Algorithm Pseudocode

We present the boundary of attraction algorithm from the work of Armiyoon and Wu [2014] for convenience.

Algorithm 9 Boundaries of Attraction (BoA) Algorithm

INPUT: VI$(F, \mathcal{X})$.

Initialize grid X over state space $\mathcal{X}$
Initialize $P(x)$, $x \in X$ to uniform distribution
Initialize hash D
repeat
  Sample x_0 from $P(x)$
  Compute Lyapunov exponent (LE) for x_0
  Compute LEs for neighbors of x_0
  if $\exists i, j$ s.t. $LE(x_i) \neq LE(x_j)$ **then**
    Save (x_i, x_j) to $D[LE(x_i)]$
    Save (x_j, x_i) to $D[LE(x_j)]$
  end if
  Update $P(x)$ according to heuristic
until Frequency of boundary detection $<$ threshold

D.2 Polynomial Coefficients for Demand Function Q_{ij}

We list the coefficients for β defined for the demand function Q_{ij} below in Table D.1.

β_0	β_1	β_2	β_3	β_4	β_5
8	-20	26	-19	7	-1

Table D.1: The polynomial function coefficients, β, for $t^c = 1$. Viable coefficients can be derived for any $t^c \in [1, 3.8]$ (see supplementary Mathematica file for derivation). Outside of that range, the demand function begins to lose properties such as monotonicity and/or the existence of the elastic/inelastic region.

D.3 Individual Cloud Profit Functions Non-Concave

The individual cloud profit functions may be non-concave. Consider $H_{11} = 10, H_{12} = 1, \alpha_{11} = 1, \alpha_{12} = 1/10, c_1 = 1, d_1 = 1, d_r = 1, \sum_{i' \neq i} p_{i'} = 2$. Then

$$\pi_1(p_1, d_1) = (10e^{-(\frac{p_1^2}{p_1+2})^2} + e^{-(\frac{1}{10}\frac{p_1^2}{p_1+2})^2})(p_1 - 1). \tag{D.1}$$

Figure D.1 shows the function.

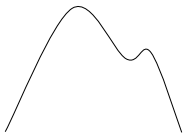


Figure D.1: Individual profit functions may be non-concave.

D.4 Model Parameters for Scenario 1

Below, we have listed the parameters that define the market for Scenario 1.

c_1	c_2	c_3	c_4
1.05	1.10	0.95	1.15

Table D.2: Cloud cost function coefficients.

α_{ij}	Cloud 1	Cloud 2	Cloud 3	Cloud 4
Client 1	0.27	0.27	0.27	0.38
Client 2	0.34	0.34	0.34	0.31
Client 3	0.33	0.33	0.33	0.26
Client 4	0.25	0.40	0.40	0.34

Table D.3: Client preferences.

H_{ij}	Cloud 1	Cloud 2	Cloud 3	Cloud 4
Client 1	11	11	11	11
Client 2	9	9	9	9
Client 3	6	6	6	6
Client 4	12	12	12	12

Table D.4: Client scale factors.

D.5 Model Parameters for Scenario 2

In scenario 2, the first client refocuses their loyalty towards the newly introduced green tech cloud, cloud 5. All other parameters remain the same.

α_{ij}	Cloud 1	Cloud 2	Cloud 3	Cloud 4
Client 1	**0.38**	**0.38**	**0.38**	**0.27**
Client 2	0.34	0.34	0.34	0.31
Client 3	0.33	0.33	0.33	0.26
Client 4	0.25	0.40	0.40	0.34

Table D.5: Business preferences.

D.6 Model Parameters for BoA Demonstration

These are the parameters that define the market for the BoA demonstration. Everything remains the same from scenario 1; the only changes come with the addition of cloud 5.

c_1	c_2	c_3	c_4	$\boldsymbol{c_5}$
1.05	1.10	0.95	1.15	**1.20**

Table D.6: Cloud cost function coefficients.

α_{ij}	Cloud 1	Cloud 2	Cloud 3	Cloud 4	**Cloud 5**
Client 1	0.27	0.27	0.27	0.38	**0.38**
Client 2	0.34	0.34	0.34	0.31	**0.31**
Client 3	0.33	0.33	0.33	0.26	**0.26**
Client 4	0.25	0.40	0.40	0.34	**0.34**

Table D.7: Client preferences.

H_{ij}	Cloud 1	Cloud 2	Cloud 3	Cloud 4	**Cloud 5**
Client 1	11	11	11	11	**11**
Client 2	9	9	9	9	**9**
Client 3	6	6	6	6	**6**
Client 4	12	12	12	12	**12**

Table D.8: Client scale factors.

BIBLIOGRAPHY

[1] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Michele Aghassi, Dimitris Bertsimas, and Georgia Perakis. Solving asymmetric variational inequalities via convex optimization. *Operations Research Letters*, 34(5):481–490, 2006.

[3] A. A. Ahmadi, A. Olshevsky, P. A. Parrilo, and J. N. Tsitsiklis. Np-hardness of deciding convexity of quartic polynomials and related problems. *Mathematical Programming*, 2013.

[4] Felipe Alvarez. On the minimizing property of a second order dissipative system in hilbert spaces. *SIAM Journal on Control and Optimization*, 38(4):1102–1119, 2000.

[5] S. I. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 1998.

[6] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[7] Ali Reza Armiyoon and Christine Qiong Wu. An innovative approach for identifying boundaries of a basin of attraction for a dynamical system using Monte Carlo techniques and Lyapunov exponents. In *53rd IEEE Conference on Decision and Control, CDC 2014, Los Angeles, CA, USA, December 15-17, 2014*, pages 6299–6304. IEEE, 2014. ISBN 978-1-4799-7746-8. doi: 10.1109/CDC.2014.7040376. URL `http://dx.doi.org/10.1109/CDC.2014.7040376`.

[8] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang. Generalization and equilibrium in generative adversarial nets (gans). *arXiv preprint arXiv:1703.00573*, 2017.

[9] Muhammad Aslam Noor. Generalized set-valued variational inequalities. *Le Matematiche*, 52(1):3–24, 1998.

[10] Kamyar Azizzadenesheli, Brandon Yang, Weitang Liu, Emma Brunskill, Zachary C Lipton, and Animashree Anandkumar. Sample-efficient deep rl with generative adversarial tree search. *arXiv preprint arXiv:1806.05780*, 2018.

[11] David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. *arXiv preprint arXiv:1802.05642*, 2018.

[12] David Balduzzi, Karl Tuyls, Julien Perolat, and Thore Graepel. Re-evaluating evaluation. *arXiv preprint arXiv:1806.02643*, 2018.

[13] T. Basar and G. J. Olsder. *Dynamic noncooperative game theory*. SIAM, 1999.

[14] Alina Beygelzimer, Satyen Kale, and Haipeng Luo. Optimal and adaptive algorithms for online boosting. *arXiv preprint arXiv:1502.02651*, 2015.

[15] V. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

[16] V. S. Borkar and S. P. Meyn. The ode method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 2000.

[17] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[18] X. Cai, G. Gu, and B. He. On the o(1/t) convergence rate of the projection and contraction methods for variational inequalities with lipschitz continuous monotone operators. *Computational Optimization and Applications*, 2014.

[19] Ennio Cavazzuti, Massimo Pappalardo, and Mauro Passacantando. Nash equilibria, variational inequalities, and dynamical systems. *Journal of optimization theory and applications*, 114(3):491–506, 2002.

[20] Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–10. IEEE, 2018.

[21] Antoine-Augustin Cournot. *Recherches sur les principes mathématiques de la théorie des richesses par Augustin Cournot*. chez L. Hachette, 1838.

[22] J. P. Crouzeix and J. A. Ferland. Criteria for differentiable generalized monotone maps. *Mathematical Programming*, 1996.

[23] S. Dafermos. Traffic equilibria and variational inequalities. *Transportation Science*, 14:42–54, 1980.

[24] Stella Dafermos. An iterative scheme for variational inequalities. *Mathematical Programming*, 26(1):40–47, 1983.

[25] C. D. Dang and G. Lan. On the convergence properties of non-euclidean extragradient methods for variational inequalities with generalized monotone operators. *Computational Optimization and Applications*, 2015.

[26] L. de Oliveira, M. Paganini, and B. Nachman. Learning particle physics by example: location-aware generative adversarial networks for physics synthesis. *Computing and Software for Big Science*, 2017.

[27] SV Denisov, VV Semenov, and LM Chabak. Convergence of the modified extragradient method for variational inequalities with non-lipschitz operators. *Cybernetics and Systems Analysis*, 51(5):757–765, 2015.

[28] Emily Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with context-conditional generative adversarial networks. *arXiv preprint arXiv:1611.06430*, 2016.

[29] René Descartes. *La géométrie de René Descartes*. A. Hermann, 1886.

[30] H. Drucker and Y. Le Cun. Improving generalization performance using double backpropagation. *IEEE Transactions on Neural Networks*, 1992.

[31] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[32] J. Dutta. When is a gap function good for error bounds? Technical report, Optimization Online, May 2012.

[33] Eyal Even-Dar, Yishay Mansour, and Uri Nadav. On the convergence of regret minimization dynamics in concave games. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 523–532. ACM, 2009.

[34] F. Facchinei and J. Pang. *Finite-Dimensional Variational Inequalities and Complimentarity Problems*. Springer, 2003.

[35] Michael Fairbank and Eduardo Alonso. The divergence of reinforcement learning al-gorithms with value-iteration and function ap-proximation. *arXiv preprint arXiv:1107.4606*, 2011.

[36] S. Feizi, C. Suh, F. Xia, and D. Tse. Understanding gans: the lqg setting. *arXiv preprint arXiv:1710.10793*, 2017.

[37] Dylan J Foster, Thodoris Lykouris, Karthik Sridharan, Eva Tardos, et al. Learning in games: Robustness of fast convergence. In *Advances in Neural Information Processing Systems*, pages 4727–4735, 2016.

[38] T. L. Friesz. *Dynamic optimization and differential games*. Springer Science & Business Media, 2010.

[39] Gauthier Gidel, Hugo Berard, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial nets. *arXiv preprint arXiv:1802.10551*, 2018.

[40] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[41] Geoffrey J Gordon, Amy Greenwald, and Casey Marks. No-regret learning in convex games. In *Proceedings of the 25th international conference on Machine learning*, pages 360–367. ACM, 2008.

[42] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, 2017.

[43] P. Hartman and G. Stampacchia. On some nonlinear elliptic differential functional equations. *Acta Mathematica*, 115:271–310, 1966.

[44] D. J. Higham, A. R. Humphries, and R. J. Wain. Phase space error control for dynamical systems. *SIAM Journal on Scientific Computing*, 2000.

[45] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *arXiv preprint arXiv:1606.03476*, 2016.

[46] Xiaolin Hu and Jun Wang. Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network. *IEEE Transactions on Neural Networks*, 17(6):1487–1499, 2006.

[47] Daniel Jiwoong Im, Chris Dongjoo Kim, Hui Jiang, and Roland Memisevic. Generating images with recurrent adversarial networks. *arXiv preprint arXiv:1602.05110*, 2016.

[48] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[49] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[50] A. N. Iusem, A. Jofré, R. I. Oliveira, and P. Thompson. Extragradient method with variance reduction for stochastic variational inequalities. *SIAM Journal on Optimization*, 2017.

[51] Tyler Jarvis and James Tanton. The hairy ball theorem via sperner's lemma. *The American Mathematical Monthly*, 111(7):599–603, 2004.

[52] Ramesh Johari and John N Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, 29(3):407–435, 2004.

[53] A. Juditsky, A. Nemirovski, and C. Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 2011.

[54] Artur Kadurin, Alexander Aliper, Andrey Kazennov, Polina Mamoshina, Quentin Vanhaelen, Kuzma Khrabrov, and Alex Zhavoronkov. The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget*, 8(7):10883, 2017.

[55] A. Kannan and U. V. Shanbhag. Optimal stochastic extragradient schemes for pseudomonotone stochastic variational inequality problems and their variants. *arXiv preprint arXiv:1410.1628*, 2017.

[56] Aswin Kannan and Uday V Shanbhag. The pseudomonotone stochastic variational inequality problem: Analytical statements and stochastic extragradient schemes. In *American Control Conference (ACC), 2014*, pages 2930–2935. IEEE, 2014.

[57] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[58] H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, New Jersey, 1996.

[59] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[60] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[61] G. Korpelevich. The extragradient method for finding saddle points and other problems. 1977.

[62] GM Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

[63] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Master's Thesis*, 2009.

[64] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The mnist database of handwritten digits, 1998.

[65] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017.

[66] Alexanckr Levin. An analytical method of estimating the domain of attraction for polynomial differential equations. *Automatic Control, IEEE Transactions on*, 39(12):2471–2475, 1994. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=362845`.

[67] Dong Li, Anna Nagurney, and Min Yu. Consumer learning of product quality with time delay: Insights from spatial price equilibrium models with differentiated products. *Omega*, 81:150–168, 2018.

[68] Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Finite-sample analysis of proximal gradient td algorithms. In *UAI*, pages 504–513. Citeseer, 2015.

[69] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[70] Eliana Lorch. Visualizing deep network training trajectories with pca. In *The 33rd International Conference on Machine Learning JMLR volume*, volume 48, 2016.

[71] Thodoris Lykouris, Vasilis Syrgkanis, and Éva Tardos. Learning and efficiency in games with dynamic population. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 120–129. Society for Industrial and Applied Mathematics, 2016.

[72] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017.

[73] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[74] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. *arXiv preprint arXiv:1705.10461*, 2017.

[75] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine Learning*, pages 3478–3487, 2018.

[76] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[77] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[78] Y. Mroueh and T. Sercu. Fisher gan. In *Advances in Neural Information Processing Systems*, 2017.

[79] Y. Mroueh, T. Sercu, and V. Goel. Mcgan: Mean and covariance feature matching gan. *arXiv preprint arXiv:1702.08398*, 2017.

[80] Vaishnavh Nagarajan and J Zico Kolter. Gradient descent gan optimization is locally stable. In *Advances in Neural Information Processing Systems*, pages 5591–5600, 2017.

[81] A. Nagurney and D. Zhang. *Projected Dynamical Systems and Variational Inequalities with Applications*. Kluwer Academic Press, 1996.

[82] Anna Nagurney and Tilman Wolf. A Cournot–Nash–Bertrand game theory model of a service-oriented internet with price and quality competition among network transport providers. *Computational Management Science*, 11(4):475–502, 2014.

[83] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[84] Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[85] M Aslam Noor. Extragradient methods for pseudomonotone variational inequalities. *Journal of Optimization Theory and Applications*, 117(3):475–488, 2003.

[86] Muhammad Aslam Noor. Modified projection method for pseudomonotone variational inequalities. *Applied Mathematics Letters*, 15(3):315–320, 2002.

[87] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *arXiv preprint arXiv:1606.00709*, 2016.

[88] W. O. Paradis and D. D. Perlmutter. Tracking function approach to practical stability and ultimate boundedness. *AIChE Journal*, 12(1):130–136, 1966. ISSN 1547-5905. doi: 10.1002/aic.690120125. URL `http://dx.doi.org/10.1002/aic.690120125`.

[89] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[90] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[91] Mikhail I Rabinovich, Ramón Huerta, Pablo Varona, and Valentin S Afraimovich. Transient cognitive dynamics, metastability, and decision making. *PLoS computational biology*, 4(5):e1000072, 2008.

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

[93] Siamak Ravanbakhsh, Francois Lanusse, Rachel Mandelbaum, Jeff Schneider, and Barnabas Poczos. Enabling dark energy science with deep generative models of galaxy images. *arXiv preprint arXiv:1609.05796*, 2016.

[94] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. 2018.

[95] G Romano, L Rosati, F Marotti de Sciarra, and P Bisegna. A potential theory for monotone multi-valued operators. *Quarterly of applied mathematics*, 4(4): 613–631, 1993.

[96] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems*, pages 2018–2028, 2017.

[97] Tim Roughgarden. Intrinsic robustness of the price of anarchy. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 513–522. ACM, 2009.

[98] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

[99] Marco Sandri. Numerical calculation of Lyapunov exponents. *Mathematica Journal*, 6(3):78–84, 1996.

[100] S Schaible and Dinh The Luc. Generalized monotone nonsmooth maps. *Journal of Convex Analysis*, 3:195–206, 1996.

[101] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017.

[102] G. Scutari, D. P. Palomar, F. Facchinei, and J. Pang. Convex optimization, game theory, and variational inequality theory. *IEEE Signal Processing Magazine*, 2010.

[103] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

[104] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. Westview press, 2014.

[105] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC Press, 2018.

[106] Richard S Sutton, Hamid R Maei, and Csaba Szepesvári. A convergent $o(n)$ temporal-difference algorithm for off-policy learning with linear function approximation. In *Advances in neural information processing systems*, pages 1609–1616, 2009.

[107] Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000. ACM, 2009.

[108] Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. In *Advances in Neural Information Processing Systems*, pages 2989–2997, 2015.

[109] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844v3*, 2016.

[110] P. Thomas. Genga: A generalization of natural gradient ascent with positive and negative convergence results. In *International Conference on Machine Learning*, 2014.

[111] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[112] Federica Tinti. Numerical solution for pseudomonotone variational inequality problems by extragradient methods. In *Variational analysis and applications*, pages 1101–1128. Springer, 2005.

[113] Giulio Tononi, David Balduzzi, and MS Gazzaniga. Toward a theory of consciousness. In *The Cognitive Neurosciences*, pages 1201–1220. MIT Press, 2009.

[114] Masatoshi Uehara, Issei Sato, Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. Generative adversarial nets from a density ratio estimation perspective. *arXiv preprint arXiv:1610.02920*, 2016.

[115] Yue Wang, Alexandra Meliou, and Gerome Miklau. A consumer-centric market for database computation in the cloud. Technical report, University of Massachusetts, 2015.

[116] Hassler Whitney. *Geometric integration theory*, volume III. Courier Corporation, 2012.

[117] Alan Wolf, Jack B. Swift, Harry L. Swinney, and John A. Vastano. Determining lyapunov exponents from a time series. *Physica*, pages 285–317, 1985.

[118] Raymond A Yeh, Chen Chen, Teck-Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *CVPR*, volume 2, page 4, 2017.

[119] Donggeun Yoo, Namil Kim, Sunggyun Park, Anthony S Paek, and In So Kweon. Pixel-level domain transfer. *arXiv preprint arXiv:1603.07442*, 2016.

[120] F. Yousefian, A. Nedić, and U. V. Shanbhag. Optimal robust smoothing extragradient algorithms for stochastic variational inequality problems. In *IEEE 53rd Annual Conference on Decision and Control (CDC)*, 2014.

[121] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2528–2535. IEEE, 2010.

[122] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint*, 2017.

[123] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

[124] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.

[125] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*. School of Computer Science, Carnegie Mellon University, 2003.