SIAM J. MATH. DATA SCI.
Vol. 6, No. 3, pp. 627–648

Three-Operator Splitting for Learning to Predict Equilibria in Convex Games*

D. McKenzie†, H. Heaton‡, Q. Li§, S. Wu Fung†, S. Osher¶, and W. Yin§

Abstract. Systems of competing agents can often be modeled as games. Assuming rationality, the most likely outcomes are given by an *equilibrium*, e.g., a Nash equilibrium. In many practical settings, games are influenced by *context*, i.e., additional data beyond the control of any agent (e.g., weather for traffic and fiscal policy for market economies). Often, the exact game mechanics are unknown, yet vast amounts of historical data consisting of (context, equilibrium) pairs are available, raising the possibility of *learning* a solver that predicts the equilibria *given only the context*. We introduce Nash fixed-point networks (N-FPNs), a class of neural networks that naturally output equilibria. Crucially, N-FPNs employ a constraint decoupling scheme to handle complicated agent action sets while avoiding expensive projections. Empirically, we find that N-FPNs are compatible with the recently developed Jacobian-free backpropagation technique for training implicit networks, making them significantly faster and easier to train than prior models. Our experiments show that N-FPNs are capable of scaling to problems orders of magnitude larger than existing learned game solvers. All code is available online.

Key words. end-to-end learning, variational inequalities, game theory, operator splitting, machine learning

MSC codes. 68Q25, 68R10, 68U05

DOI. 10.1137/22M1544531

1. Introduction.

Many recent works in deep learning highlight the power of using end-to-end learning in conjunction with known analytic models and constraints [11, 56, 19, 40, 35, 24, 15, 29]. This best-of-both-worlds approach fuses the flexibility of learning-based approaches with the interpretability of models derived by domain experts. We further this line of research by proposing a practical framework for learning to predict the outcomes of contextual (i.e., parameterized) games from historical data while respecting constraints on players' actions. Many social systems can aptly be analyzed as games, including market economies [4], traffic

*Received by the editors January 6, 2023; accepted for publication (in revised form) April 22, 2024; published electronically July 11, 2024.
https://doi.org/10.1137/22M1544531
Funding: The first, second, third, fourth and fifth authors were supported by AFOSR MURI FA9550-18-1-0502 and ONR grants N00014-18-1-2527, N00014-20-1-2093, and N00014-20-1-2787. The second author's work was also supported by the National Science Foundation (NSF) Graduate Research Fellowship under grant DGE-1650604. The fourth author was also supported in part by National Science Foundation award DMS-2309810 and DMS-2110745. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

†Department of Mathematics, Colorado School of Mines, Golden, CO 80401 USA (dmckenzie@mines.edu, swufung@mines.edu).
‡Typal Academy, Richland, WA 99352 USA (research@typal.academy).
§Decision Intelligence Lab, DAMO Academy, Alibaba US, Bellevue, WA 98004 USA (liqiuweiss@gmail.com, wotao.yin@gmail.com).
¶Department of Mathematics, University of California Los Angeles, Los Angeles, CA 90095 USA (sjo@math.ucla.edu).

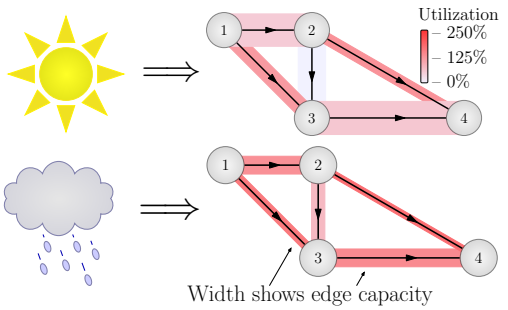

Figure 1. *Proposed N-FPNs can predict traffic flow (specifically, the utilization of each road segment) given only contextual information (e.g., weather).*

routing [65], even penalty kicks in soccer [5]. We consider games with costs parameterized by a context variable d beyond the control of any player. As in the multiarmed bandit literature, we call such games *contextual* [61]. For example, in traffic routing, d may encode factors such as weather and local sporting events or tolls influencing drivers' commutes; see Figure 1.

Game-theoretic analyses frequently assume players' cost functions are known a priori and seek to predict how players will act, typically by computing a Nash equilibrium $x_d^\star$ [50]. Informally, a Nash equilibrium is a choice of strategy for each player such that no player can improve their outcomes via unilateral deviation. However, in practice, the cost functions are frequently unknown. Here, we consider the problem of predicting equilibria given only contextual information, without knowing players' cost functions. We do so by learning an approximation to the game gradient (see (2.2)), so, in this sense, our work is closely related to the literature on inverse game theory. For technical reasons, we focus on games in which each player's cost function is strongly convex. This property is sometimes referred to as "diagonal strict convexity" [57]. We note that this class of games includes many routing games [58]. Furthermore, in many cases, it is possible—even desirable—to add a regularizer to each cost function such that it becomes strongly convex [48]; see Remark 3.2 for further discussion.

For noncontextual games, many prior works (see section 6) seek to use (noisy) observations of the Nash equilibrium to learn the cost functions. Our approach is different. We propose a new framework: Nash fixed-point networks (N-FPNs). Each N-FPN is trained on historical data pairs $(d, x_d^\star)$ to "predict the appropriate game from context and then output game equilibria" by tuning an operator so that its fixed points coincide with Nash equilibria. N-FPN inferences are computed by repeated application of the operator until a fixed point condition is satisfied. Thus, by construction, N-FPNs are implicit networks—neural networks evaluated using an arbitrary number of layers [68, 6, 23, 21]—and the operator weights can be efficiently trained using Jacobian-free backpropagation (JFB) [21]. Importantly, the N-FPN architecture incorporates a constraint decoupling scheme derived from an application of three-operator splitting [18]. This decoupling allows N-FPNs to avoid costly projections onto agents' action sets, the computational bottleneck of prior works [40, 41, 39]. This innovation allows N-FPNs to scale to large games or to games with action sets significantly more complicated than the probability simplex.

One might enquire as to the *expressiveness* of N-FPNs. That is, can a given contextual game be arbitrarily well-approximated by an N-FPN? We answer this question in the affirmative, at least for contextual games possessing the diagonal strict convexity property alluded to above.

Finally, we demonstrate how N-FPNs can be used to predict other closely related kinds of equilibria, particularly quantal response equilibria [46] and Wardrop equilibria [65]. We complement our theoretical insights with numerical experiments demonstrating the efficacy and scalability of the N-FPN framework. We end by discussing how N-FPNs might be applied to other phenomena modeled by variational inequalities.

Contributions. We provide a *scalable* data-driven framework for efficiently predicting equilibria in systems modeled as contextual games. Specifically, we do the following.

- Provide a general, expressible, and end-to-end trained model predicting Nash equilibria.
- Give a scheme for decoupling constraints for efficient forward and backward propagation.
- Prove that N-FPNs are universal approximators for a certain class of contextual games.
- Demonstrate empirically the scalability of N-FPNs to large-scale problems.

2. Preliminaries.

We begin with a brief review of relevant game theory. After establishing notation, we provide a set of assumptions under which the mapping $d \mapsto x_d^\star$ is "well-behaved." We then describe variational inequalities and how Nash equilibria can be characterized using fixed-point equations.

2.1. Games and equilibria.

Let $\mathcal{X}$ be a finite-dimensional Hilbert space. A K-player normal form contextual game is defined by action sets[1] $\mathcal{V}_k$ and cost functions $u_k : \mathcal{X} \times \mathcal{D} \to \mathbb{R}$ for $k \in [K]$, where the constraint profile is $\mathcal{C} \triangleq \mathcal{V}_1 \times \cdots \times \mathcal{V}_K$ and $\mathcal{D}$ denotes the set of contexts (i.e., data space). The kth player's action x_k is constrained to the action set $\mathcal{V}_k$, yielding an action profile $x = (x_1, \ldots, x_K) \in \mathcal{C} \subseteq \mathcal{X}$. Actions of all players other than k are $x_{-k} = (x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_K)$. Each rational player aims to minimize their cost function u_k by controlling only x_k while explicitly knowing that u_k is impacted by other players' actions x_{-k}. An action profile $x_d^\star$ is a Nash equilibrium (NE) provided that, for all $x_k \in \mathcal{V}_k$ and $k \in [K]$,

$$u_k(x_k, x_{d,-k}^\star; d) \geq u_k(x_{d,k}^\star, x_{d,-k}^\star; d). \tag{2.1}$$

In words, $x_d^\star$ is an NE if no player can decrease their cost by unilaterally deviating from $x_d^\star$. Throughout, we make the following assumptions:

(A1) $\mathcal{C} \subset \mathcal{X}$ is closed and convex.
(A2) The cost functions $u_k(x; d)$ are continuously differentiable with respect to x.
(A3) For all x, each $\nabla_k u_k(x; \cdot)$ is Lipschitz.
(A4) Each cost function $u_k(x_k, x_{-k}; d)$ is α-strongly convex with respect to x_k.
(A5) The set of contextual data $\mathcal{D}$ is compact.

When Assumption (A2) holds, we define the *game gradient* by

$$F(x; d) \triangleq \left[\nabla_{x_1} u_1(x; d)^\top \cdots \nabla_{x_K} u_K(x; d)^\top\right]^\top. \tag{2.2}$$

2.2. Variational inequalities.

This subsection briefly outlines variational inequalities and their connection to games.

[1] These are also known as *decision sets* and/or *strategy sets*.

Definition 2.1. *For $\alpha > 0$, a mapping $F\colon \mathcal{X} \times \mathcal{D} \to \mathcal{X}$ is α-cocoercive*[2] *if*

$$\langle F(x;d) - F(y;d), x - y\rangle \geq \alpha \|F(x;d) - F(y;d)\|^2 \text{ for all } x, y \in \mathcal{X}, d \in \mathcal{D} \tag{2.3}$$

and α-strongly monotone if

$$\langle F(x;d) - F(y;d), x - y\rangle \geq \alpha \|x - y\|^2 \text{ for all } x, y \in \mathcal{X}, d \in \mathcal{D}. \tag{2.4}$$

If (2.4) holds for $\alpha = 0$, then $F(\cdot\,; d)$ is monotone.

Definition 2.2. *Given $d \in \mathcal{D}$, a point $x_d^\star \in \mathcal{C}$ is a variational inequality (VI) solution provided that*

$$\langle F(x_d^\star; d), x - x_d^\star\rangle \geq 0, \quad \text{for all } \ x \in \mathcal{C}. \tag{VI}$$

The solution set for (VI) is denoted by $\mathrm{VI}(F(\cdot\,; d), \mathcal{C})$.

NEs may be characterized using VIs [20]; namely,

$$x_d^\star \text{ is an NE} \iff x_d^\star \in \mathrm{VI}(F(\cdot\,; d), \mathcal{C}). \tag{2.5}$$

That is, $x_d^\star$ is an NE if no unilateral change improves any single cost and a VI solution if no feasible update improves the sum of all costs. By (2.5), these views are equivalent.

2.3. Implicit neural networks. Commonplace feedforward neural networks are a composition of parameterized functions $T^\ell_{\Theta_\ell}(\cdot)$ (called layers) that take data d as input and return a prediction y. Formally, given d, a network $\mathcal{N}_\Theta$ computes each inference y via

$$\begin{aligned} y = \mathcal{N}_\Theta(x) &= x^{L+1}, \\ \text{where } x^1 = d \text{ and } x^{\ell+1} &= T^\ell_{\Theta_\ell}(x^\ell) \ \text{ for all } \ell \in [L]. \end{aligned} \tag{2.6}$$

Instead of an explicit cascade of distinct compositions, *implicit neural networks* $\mathcal{N}_\Theta$ use a single mapping T_Θ, and the output $\mathcal{N}_\Theta(d)$ is defined implicitly[3] by an equation, e.g.,

$$\mathcal{N}_\Theta(d) \triangleq x_d^\circ, \quad \text{where} \quad x_d^\circ = T_\Theta(x_d^\circ; d). \tag{2.7}$$

Equation (2.7) can be solved via a number of methods, e.g., fixed-point iteration: $x^{k+1} = T_\Theta(x^k; d)$. Implicit neural networks recently received much attention because they admit a memory efficient backprop [6, 7, 23, 21, 22]. By construction, the output of $\mathcal{N}_\Theta(d)$ is a fixed point. Thus, several recent works explore using implicit networks in supervised learning problems, where the target to be predicted can naturally be interpreted as a fixed point [28, 24, 31, 27, 47].

3. Well-behaved equilibria. We verify that Assumptions (A1), (A2), (A3), (A4), and (A5) are sufficient to guarantee that $x_d^\star$ depends smoothly on d. This is crucial for showing that an N-FPN can approximate the relationship between d and $x_d^\star$ (see Theorem 4.1).

[2] This is also known as α-inverse strongly monotone.

[3] We reserve the notation $x_d^\star$ for denoting equilibria, fixed points, or VI solutions associated to the true game we wish to approximate. We use x_d° for denoting equilibria, fixed points, or VI solutions associated to the approximating neural network.

Theorem 3.1. *If Assumptions* (A1), (A2), (A3), (A4), *and* (A5) *hold, then*
(1) *there is a unique NE* $x_d^\star$ *for all* $d \in \mathcal{D}$;
(2) *the map* $d \mapsto x_d^\star$ *is Lipschitz continuous.*

Proof. Assumption (A4) implies the game gradient $F(\cdot; d)$ is α-strongly monotone. Hence, by [57, Theorem 2] the NE $x_d^\star$ is unique; see also [20, Theorem 2.2.3]. This proves part 1. For part 2, first observe that (A3) implies that F is Lipschitz continuous with respect to d in addition to being α-strongly monotone. [17, Theorem 2.1] then shows that, around any fixed $\bar{d} \in \mathcal{D}$, the map $d \mapsto x_d^\star$ is locally Lipschitz; i.e., there exists a constant $L_{\bar{d}}$ and an open neighborhood $N_{\bar{d}} \subset \mathcal{D}$ of $\bar{d}$ upon which $d \mapsto x_d^\star$ is $L_{\bar{d}}$-Lipschitz continuous. Because $\mathcal{D}$ is compact (Assumption (A5)), a standard covering argument converts this local Lipschitz property to a global Lipschitz property. ∎

Remark 3.2. Assumption (A4) is fairly restrictive but in line with prior work [40, 39, 1, 11, 54, 69]. For games where the u_k are not strongly convex, one can add a regularizer: $\tilde{u}_k(x) = u_k(x) + h(x)$. As an illustrative example, consider the case where each player's action set is the probability simplex

$$\mathcal{V}_k = \Delta_n := \left\{ x \in \mathbb{R}^n : \sum_j x_{k,j} = 1 \text{ and } x_{k,j} \geq 0 \right\}. \tag{3.1}$$

Adding an entropic regularizer—i.e., $h(x) = \sum_j x_{k,j} \log(x_{k,j})$ as in [40]—affords an elegant interpretation of the NE of the resulting regularized game as the quantal response equilibrium (QRE) [46] of the original game. QRE are a useful solution concept for boundedly rational agents (e.g., humans). They describe situations where agents are likely to select the best action but may also select a suboptimal action with nonzero probability. However, choosing such an h means that $\|\nabla_k \tilde{u}_k\| \to \infty$ as x_k approaches the boundary of $\mathcal{V}_k$, which may be undesirable. Formally, one may resolve this by using a "smoothed" entropic regularizer $h(x) = \sum_j x_{k,j} \log(x_{k,j} + \epsilon)$, which does satisfy Assumption (A3), as discussed in [42]. In practice, this seems unnecessary; see section 7.1. Alternatively, one may use an $h(x)$ that does not diverge as x_k approaches the boundary of $\mathcal{V}_k$, such as a quadratic penalty. We refer the reader to [48] for further discussion on the choice of $h(x)$ and the interpretation thereof.

4. Proposed method: N-FPNs.

Recall that our goal is to train a predictor capable of approximating $x_d^\star$ given only d. We assume a fixed, yet unknown, contextual game that induces a probability distribution μ on $\mathcal{D} \times \mathcal{X}$ relating d and $x_d^\star$. As a predictor, we propose to use an N-FPN $\mathcal{N}_\Theta$, defined abstractly as the solution to a parameterized variational inequality

$$\mathcal{N}_\Theta(d) \triangleq \mathrm{VI}(F_\Theta(\cdot; d), \mathcal{C}). \tag{4.1}$$

Below, we discuss how an N-FPN can be viewed concretely as an implicit neural network. In our context, the set $\mathcal{C}$ is a product of action sets $\mathcal{V}_k$ and $F_\Theta(\cdot;\cdot)$ is a neural network with weights Θ.

Fixing a smooth loss function $\ell : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, in principle, one selects a predictor (i.e., a choice of weights Θ) via minimizing the population risk:

$$\min_\Theta \mathbb{E}_{(d, x_d^\star) \sim \mu} \left[\ell(\mathcal{N}_\Theta(d), x_d^\star) \right]. \tag{4.2}$$

In practice, one minimizes the *empirical risk*, given a training dataset $\{(d_i, x^\star_{d_i})\}_{i=1}^N \sim_\mu \mathcal{D}\times\mathcal{X}$, instead [64]:

$$\min_\Theta \sum_{i=1}^N \ell(\mathcal{N}_\Theta(d_i), x^\star_{d_i}). \tag{4.3}$$

A similar approach was proposed in [39]; we discuss how our approach improves upon theirs in section 6. First, we provide a novel theorem guaranteeing that the proposed design has sufficient capacity to accurately approximate the mapping $d \mapsto x^\star_d$ for games of interest.

Theorem 4.1 (universal approximation). *If Assumptions* (A1), (A2), (A3), (A4), *and* (A5) *hold, then, for all* $\varepsilon > 0$, *there exists* $F_\Theta(\cdot;\cdot)$ *such that* $\max_{d\in\mathcal{D}} \|x^\star_d - \mathcal{N}_\Theta(d)\|_2 \le \varepsilon$.

A proof of Theorem 4.1 can be found in the supplemental material (supplement.pdf [local/web 437KB]). Since N-FPNs are universal approximators in theory, two practical questions arise:

1) For a given d, how are inferences of $\mathcal{N}_\Theta(d)$ computed?
2) How are weights Θ tuned using training data $\{d, x^\star_d\}$?

We address each inquiry in turn. As is well-known [20], for all $\alpha > 0$,

$$x^\circ_d \in \mathrm{VI}(F_\Theta(\cdot\,; d), \mathcal{C}) \quad \Longleftrightarrow \quad x^\circ_d = P_\mathcal{C}(x^\circ_d - \alpha F_\Theta(x^\circ_d; d)), \tag{4.4}$$

where $P_\mathcal{C}$ denotes the projection onto the set $\mathcal{C}$; i.e., $P_\mathcal{C}(x) \triangleq \mathrm{argmin}_{y\in\mathcal{C}} \|y - x\|^2$. When the operator $P_\mathcal{C} \circ (\mathrm{I} - \alpha F_\Theta)$ on the right-hand side of (4.4) is tractable and well-behaved, inferences of $\mathcal{N}_\Theta(d)$ can be computed via a fixed point iteration, as in [39]. Unfortunately, for some $\mathcal{C}$, computing $P_\mathcal{C}$ and $\mathrm{d}P_\mathcal{C}/\mathrm{d}z$ requires a number of operations scaling *cubicly* with the dimension of $\mathcal{C}$ [2], rendering this approach intractable even for moderately sized problems.

Our key insight is that there are multiple ways to turn (4.1) into a fixed point problem. Specifically, we propose a fixed point formulation that, while superficially more complicated, avoids expensive projections and is easy to backpropagate through. The key ingredient (see Lemma 4.2) is an application of three-operator splitting [18], which replaces $P_\mathcal{C}$ with projection operators possessing simple and explicit projection formulas. Similar ideas can be found in [18, 51], but, to the best of our knowledge, this splitting has not yet appeared in the implicit neural network literature.

We present this architecture concretely as Algorithm 4.1. With a slight abuse of terminology, we refer to this architecture also as an N-FPN. Although we find Algorithm 4.1 to be most practical, we note other operator-based methods (e.g., the alternating direction method of multipliers (ADMM) and primal-dual hybrid gradient (PDHG)) can be used within the N-FPN framework via equivalences of different fixed-point formulations of the VI.

The proposed fixed point operator T_Θ below in (4.5) is computationally cheaper to evaluate than that in (4.4) when the projections $P_{\mathcal{C}_1}$ and $P_{\mathcal{C}_2}$ are computationally cheaper than $P_\mathcal{C}$. For example, suppose that $\mathcal{C}$ is a polytope written in general form: $\mathcal{C} = \{x : Ax = b \text{ and } x \ge 0\}$. Here, computing $P_\mathcal{C}(x)$ amounts to solving the quadratic program $\min_{y\in\mathcal{C}} \|x - y\|_2^2$. However, we may instead take $\mathcal{C}_1 = \{x : Ax = b\}$ and $\mathcal{C}_2 = \{x : x \ge 0\}$, both of which can[4] enjoy straightforward closed-form projection operators $P_{\mathcal{C}_1}$ and $P_{\mathcal{C}_2}$. Also, taking $\mathcal{C}_2 = \mathcal{C}$ and $\mathcal{C}_1 = \mathcal{X}$

[4]This depends on some properties of A (e.g., rank).

Algorithm 4.1. N-FPN (abstract form).

$\mathcal{N}_\Theta(d)$:
  $z^1 \leftarrow \tilde{z}$, $z^0 \leftarrow \tilde{z}$, $n \leftarrow 1$
  while $\|z^n - z^{n-1}\| > \varepsilon$ **or** $n = 1$
    $x^{n+1} \leftarrow P_{\mathcal{C}^1}(z^n)$
    $y^{n+1} \leftarrow P_{\mathcal{C}^2}(2x^{n+1} - z^n - \gamma F_\Theta(x^{n+1}; d))$
    $z^{n+1} \leftarrow z^n - x^{n+1} + y^{n+1}$
    $n \leftarrow n + 1$
  return $P_{\mathcal{C}^1}(z^n)$

Algorithm 4.2. N-FPN—projected gradient (special case).

$\mathcal{N}_\Theta(d)$: ◁ Input data are d
  $x^1 \leftarrow \tilde{x}$, $n \leftarrow 2$, ◁ Initializations
  $x^2 \leftarrow P_{\mathcal{C}}(x^1 - F_\Theta(x^1; d))$ ◁ Apply T update
  while $\|x^n - x^{n-1}\| > \varepsilon$ ◁ Loop to converge
    $x^{n+1} \leftarrow P_{\mathcal{C}}(x^n - F_\Theta(x^n; d))$ ◁ Apply T update
    $n \leftarrow n + 1$ ◁ Iterate counter
  return x^n ◁ Output inference

(i.e., the whole space) reduces (4.6) to (4.4). For completeness, we present this special case of an N-FPN as Algorithm 4.2 becausethis is more comparable to the approaches proposed in prior work [40, 39].

Below, we provide a lemma justifying the decoupling of constraints in the action set $\mathcal{C}$. Here, we make use of polyhedral sets;[5] however, this result also holds in a more general setting utilizing relative interiors of $\mathcal{C}^1$ and $\mathcal{C}^2$. By $\delta_{\mathcal{C}} : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$, we denote the indicator function defined such that $\delta_{\mathcal{C}}(x) = 0$ in $\mathcal{C}$ and $+\infty$, elsewhere. The subgradient of the indicator function (also known as the normal cone of $\mathcal{C}$) is denoted by $\partial\delta_{\mathcal{C}}$.

Lemma 4.2. *Fix $\gamma > 0$. Suppose that $\mathcal{C} = \mathcal{C}_1 \cap \mathcal{C}_2$ for convex $\mathcal{C}_1$ and $\mathcal{C}_2$. If both $\mathcal{C}_i$ are polyhedral or have relative interiors with a point in common and the VI has a unique solution, then*

$$T_\Theta(x; d) \triangleq x - P_{\mathcal{C}^1}(x) + P_{\mathcal{C}^2}\left(2P_{\mathcal{C}^1}(x) - x - \gamma F_\Theta(P_{\mathcal{C}^1}(x); d)\right) \tag{4.5}$$

yields the equivalence

$$\mathcal{N}_\Theta(d) = x_d^\circ \iff x_d^\circ = P_{\mathcal{C}^1}(z_d^\circ) \text{ where } z_d^\circ = T_\Theta(z_d^\circ; d). \tag{4.6}$$

Proof. We begin with the well-known equivalence relation [20]

$$x_d^\circ \in \mathrm{VI}(F_\Theta(\cdot\,; d), \mathcal{C}) \iff 0 \in F_\Theta(x_d^\circ\,; d) + \partial\delta_{\mathcal{C}}(x_d^\circ). \tag{4.7}$$

[5] A set is polyhedral if it is of the form $\{x : \langle x, a^i \rangle \le b_i \text{ for } i \in [p]\}$ for $p \in \mathbb{N}$.

Because $\mathcal{C}^1$ and $\mathcal{C}^2$ are either polyhedral sets or share a common relative interior, we may apply [55, Theorem 23.8.1] to assert that

$$\partial\delta_{\mathcal{C}} = \partial\delta_{\mathcal{C}^1} + \partial\delta_{\mathcal{C}^2}. \tag{4.8}$$

Consider three maximal[6] monotone operators A, B, and C, with C single-valued. For $\gamma > 0$, let $J_{\gamma A}$ and $R_{\gamma A}$ be the resolvent of γA and reflected resolvent of γA, respectively; i.e.,

$$J_{\gamma A} \triangleq (\mathrm{I} + \gamma A)^{-1} \quad \text{and} \quad R_{\gamma A} \triangleq 2J_{\gamma A} - \mathrm{I}. \tag{4.9}$$

In particular, the resolvent of $\partial\delta_{\mathcal{C}^i}$ is precisely the projection operator $P_{\mathcal{C}^i}$ [10, Example 23.4]. Using three-operator splitting (e.g., see [18, Lemma 2.2] and [59]), we obtain the equivalence

$$0 \in (A + B + C)(x) \iff x = J_{\gamma B}(z), \tag{4.10}$$

where

$$z = z - J_{\gamma B}(z) + J_{\gamma A}(R_{\gamma B} - \gamma C J_{\gamma B})(z). \tag{4.11}$$

Setting $A = \partial\delta_{\mathcal{C}^2}$, $B = \partial\delta_{\mathcal{C}^1}$, and $C = F_\Theta$, (4.11) reduces to

$$0 \in F_\Theta(x_d^\circ; d) + \partial\delta_{\mathcal{C}^1}(x_d^\circ) + \partial\delta_{\mathcal{C}^2}(x_d^\circ) \iff x_d^\circ = P_{\mathcal{C}_1}(z_d^\circ), \text{ where } z_d^\circ = T_\Theta(z_d^\circ; d). \tag{4.12}$$

Combining (4.7), (4.8), and (4.12) yields the desired result. ∎

4.1. Forward propagation. Given the operator $T_\Theta(\cdot; d)$, there are many algorithms for determining its fixed point x_d°. Prior works [40, 39] use Newton-style methods, which are fast for small-scale and sufficiently smooth problems. But they may scale poorly to high dimensions (i.e., large $\dim(\mathcal{C})$). We employ Krasnosel'skiĭ–Mann (KM) iteration, which is the abstraction of splitting algorithms with low per-iteration computational and memory footprint. This is entirely analogous to the trade-off between first-order (e.g., gradient descent, proximal-gradient) and second-order methods (e.g., Newton) in high-dimensional optimization; see [59] for further discussion. The next theorem provides a sufficient condition under which KM iteration converges. Because the proof is standard, we relegate it to the supplemental material (supplement.pdf [local/web 437KB]).

Theorem 4.3. *Suppose that $\mathcal{C}^1$ and $\mathcal{C}^2$ are as in Lemma 4.2 and F_Θ is α-cocoercive. If a sequence $\{z^k\}$ is generated via $z^{k+1} = T_\Theta(z^k; d)$ for T_Θ in (4.5) with $\gamma = \alpha$ and $\{z : z = T_\Theta(z; d)\} \neq \varnothing$, then $P_{\mathcal{C}_1}(z^k) \to x_d^\circ = \mathcal{N}_\Theta(d)$. Moreover, the computational complexity to obtain an estimate x^k with fixed-point residual norm no more than $\epsilon > 0$ is $\mathcal{O}\left(\dim(\mathcal{C})^2/\epsilon^2\right)$.*

We simplify the iterate updates for T_Θ in (4.5) by introducing auxiliary sequences $\{x^k\}$ and $\{y^k\}$; see Algorithm 4.1.

Remark 4.4. There are several ways to design the architecture of F_Θ so that it is guaranteed to be cocoercive, regardless of Θ. For example, there are the following:

[6] A monotone operator M is *maximal* if there is no other monotone operator S such that $\mathrm{Gra}(M) \subset \mathrm{Gra}(S)$ properly [59]. This is a technical assumption that holds for all cases of our interest.

1. One easily verifies that, if $F_\Theta(\cdot; d)$ is α-strongly monotone and L-Lipschitz, then it is α/L^2-cocoercive [45]. Spectral normalization [49] can be used to ensure that $F_\Theta(\cdot; d)$ is 1-Lipschitz for most architectural choices. If a linear (in x) $F_\Theta(\cdot; d)$ suffices, one may use the parameterization suggested in [68]:

$$F_\Theta(x; d) = \left(\alpha I + A^\top A + B^\top - B\right) x + Q_{\Theta'}(d) \tag{4.13}$$

with $\alpha \in (0, 1)$ to guarantee that F_Θ is α-strongly monotone, in addition to spectral normalization. Here, $Q_{\Theta'}$ is any neural network mapping context to latent space and A, B may depend on d. We implement this in subsection 7.1 and observe that it performs well. If a more sophisticated F_Θ is required, one could use [52] to parameterize a nonlinear monotone operator $\tilde{F}_\Theta$, whence $F_\Theta = \alpha I + \tilde{F}_\Theta$ is α-strongly monotone. We caution that the parameterization given in [52] is indirect—$\tilde{F}_\Theta$ is given as the resolvent of a nonexpansive operator Q_Θ—and so is unlikely to work well with three-operator splitting.
2. By the Baillon–Haddad theorem [8, 9], if $f_\Theta(x; d)$ is a convex and L-Lipschitz–differentiable $\mathbb{R}$-valued function, then $F_\Theta(x; d) = \nabla_x f_\Theta(x; d)$ is $1/L$-cocoercive. Using the architecture proposed in [3] guarantees that $f_\Theta(\cdot; d)$ is convex (in x), and spectral normalization may again be applied to $F_\Theta(x; d)$ to ensure 1-Lipschitz differentiability. However, it is not clear that $F_\Theta(x; d)$ constructed in this manner will be easy to train. Indeed, [60, section 2] suggests it is better to parameterize $F_\Theta(x; d)$ directly and not as the gradient of some function for an analogous problem in diffusion-based generative modeling.

4.2. Backpropagation. In order to solve (4.3) using gradient-based methods such as stochastic gradient descent or ADAM [33], one needs to compute the gradient $\mathrm{d}\ell/\mathrm{d}\Theta$. To circumvent backpropagating through each forward step, $\mathrm{d}\ell/\mathrm{d}\Theta$ may be expressed by[7]

$$\frac{\mathrm{d}\ell}{\mathrm{d}\Theta} = \frac{\mathrm{d}\ell}{\mathrm{d}x}\frac{\mathrm{d}\mathcal{N}_\Theta}{\mathrm{d}\Theta} = \frac{\mathrm{d}\ell}{\mathrm{d}x}\frac{\mathrm{d}P_{\mathcal{C}^1}(z_d^\circ)}{\mathrm{d}z}\frac{\mathrm{d}z_d^\circ}{\mathrm{d}\Theta}. \tag{4.14}$$

Starting with the definition of z_d° as a fixed point

$$z_d^\circ = T_\Theta(z_d^\circ; d) \tag{4.15}$$

and appealing to the implicit function theorem [37], we obtain the Jacobian-based equation

$$\frac{\mathrm{d}z_d^\circ}{\mathrm{d}\Theta} = \mathcal{J}_\Theta^{-1}\frac{\partial T_\Theta}{\partial \Theta} \quad \text{with} \quad \mathcal{J}_\Theta \triangleq \mathrm{Id} - \frac{\mathrm{d}T_\Theta}{\mathrm{d}z}. \tag{4.16}$$

Solving (4.16) (assuming that $\mathcal{J}_\Theta$ is invertible; see Remark 4.5) is computationally intensive for large-scale games. Instead, we employ JFB, which consists of replacing $\mathcal{J}_\Theta^{-1}$ in (4.16) with the identity matrix and using

$$p_\Theta := \frac{\mathrm{d}\ell}{\mathrm{d}x}\frac{\mathrm{d}P_{\mathcal{C}^1}(z_d^\circ)}{\mathrm{d}z}\frac{\partial T_\Theta}{\partial \Theta} \tag{4.17}$$

[7] All arguments are implicit and use N-FPNs defined by (4.6).

in lieu of $\mathrm{d}\ell/\mathrm{d}\Theta$. This substitution yields a preconditioned gradient and is effective for training in image classification [21] and data-driven CT reconstructions [28]. Importantly, using JFB only requires backpropagating through a single application of T_Θ (i.e., the final forward step) in order to compute $\partial T_\Theta/\partial\Theta$.

Remark 4.5. One sufficient condition commonly used to guarantee the invertibility of $\mathcal{J}_\Theta$ is to assume that T_Θ is *contractive*, although this condition rarely holds in practice and implicit networks empirically perform well without a firm guarantee of invertibility [6, 7]. Contractivity of T_Θ is also necessary to guarantee that p_Θ is a descent direction (see [21, Theorem 3.1]), although again, this appears unnecessary in practice [21, 53, 36, 72]. We note that T_Θ is averaged if F_Θ is cocoercive. The use of JFB for averaged operators is an ongoing topic of interest; see [47].

5. Further constraint decoupling.

As discussed above, the architecture expressed in Algorithm 4.1 provides a massive computational speed-up over prior architectures when $\mathcal{C}=\mathcal{C}_1\cap\mathcal{C}_2$ and $P_{\mathcal{C}_1}$ and $P_{\mathcal{C}_2}$ admit explicit and computationally cheap expressions, e.g., when $\mathcal{C}$ is a polytope. Yet, in many practical problems, $\mathcal{C}$ has a more complicated structure. For example, it may be the intersection of a large number of sets (i.e., $\mathcal{C}=\mathcal{C}_1\cap\cdots\cap\mathcal{C}_K$) or the Minkowski sum of intersections of simple sets (i.e., $\mathcal{C}=\mathcal{C}_1+\cdots+\mathcal{C}_K$, where $\mathcal{C}_k=\mathcal{C}_k^1\cap\mathcal{C}_k^2$). We generalize our decoupling scheme by passing to a product space. With this extended decoupling, we propose an N-FPN architecture with efficient forward propagation (i.e., evaluation of $\mathcal{N}_\Theta$) and backward propagation (to tune weights Θ) using only the projection operators $P_{\mathcal{C}_k}$ (for the K-intersection case) or $P_{\mathcal{C}_k^i}$ (for the Minkowski sum case). We discuss the Minkowski sum case here and defer the K-intersection case to the supplemental material (supplement.pdf [local/web 437KB]).

5.1. Minkowski sum.

This subsection provides a decoupling scheme for constraints structured as a Minkowski sum;[8] i.e.,

$$\mathcal{C}\triangleq\mathcal{C}_1+\cdots+\mathcal{C}_K, \tag{5.1}$$

where $\mathcal{C}_k\subset\mathcal{X}$ and $\mathcal{C}_k=\mathcal{C}_k^1\cap\mathcal{C}_k^2$ for all $k\in[K]$. The core idea is to avoid attempting to directly project onto $\mathcal{C}$ and instead perform simple projections onto each set $\mathcal{C}_k^i$, assuming that the projection onto $\mathcal{C}_k^i$ admits an explicit formula. First, define the product space

$$\overline{\mathcal{X}}\triangleq\underbrace{\mathcal{X}\times\mathcal{X}\times\cdots\times\mathcal{X}}_{K\text{ times}}. \tag{5.2}$$

For notational clarity, we denote elements of $\overline{\mathcal{X}}$ by overlines so that each element $\overline{x}\in\overline{\mathcal{X}}$ is of the form $\overline{x}=(\overline{x}_1,\dots,\overline{x}_K)$ with $\overline{x}_k\in\mathcal{X}$ for all $k\in[K]$. Because $\mathcal{X}$ is a Hilbert space, $\overline{\mathcal{X}}$ is naturally endowed with a scalar product $\langle\cdot,\cdot\rangle_{\overline{\mathcal{X}}}$ defined by

$$\langle\overline{x},\overline{y}\rangle_{\overline{\mathcal{X}}}\triangleq\sum_{k=1}^{K}\langle\overline{x}_k,\overline{y}_k\rangle. \tag{5.3}$$

[8] This arises in the modeling Wardrop equilibria in traffic routing problems.

Between $\mathcal{X}$ and the product space $\overline{\mathcal{X}}$, we define two maps $Q^-\colon \overline{\mathcal{X}} \to \mathcal{X}$ and $Q^+\colon \mathcal{X} \to \overline{\mathcal{X}}$:

$$Q^-(\overline{x}) \triangleq \sum_{k=1}^{K} \overline{x}_k \quad \text{and} \quad Q^+(x) \triangleq (\underbrace{x, x, \dots, x}_{K \text{ copies}}). \tag{5.4}$$

In words, $Q^-(\overline{x})$ maps down to $\mathcal{X}$ by adding together the blocks of $\overline{x}$ and $Q^+(x)$ maps up to $\overline{\mathcal{X}}$ by making K copies of x, thus motivating the use of "+" and "−" signs. Define the Cartesian product

$$\mathcal{A} \triangleq \mathcal{C}_1 \times \cdots \times \mathcal{C}_K \subseteq \overline{\mathcal{X}}, \tag{5.5}$$

and note that $Q^-(\mathcal{A}) = \mathcal{C}$. To further decouple each set $\mathcal{C}_k$, also define the Cartesian products

$$\mathcal{A}^i \triangleq \mathcal{C}_1^i \times \cdots \times \mathcal{C}_K^i \quad \text{for all } i \in [2] \tag{5.6}$$

so that $\mathcal{A} = \mathcal{A}^1 \cap \mathcal{A}^2$. Note that the projection onto $\mathcal{A}^i$ can be computed componentwise; namely,

$$P_{\mathcal{A}^i}(\overline{x}) = \big(P_{\mathcal{C}_1^i}(\overline{x}_1), \dots, P_{\mathcal{C}_K^i}(\overline{x}_K)\big) \quad \text{for all } i \in [2]. \tag{5.7}$$

We now rephrase Algorithm 4.1, applied to a VI in the product space $\mathrm{VI}(Q^+ \circ F \circ Q^-, \mathcal{A})$, into Algorithm 5.1 using $\mathcal{A}^i$ in lieu of $\mathcal{C}^i$. Here, F represents a neural network $F_\Theta(\cdot; d)$ with weights Θ; for notational clarity, we omit the arguments and subscript. The use of Algorithm 5.1 is justified by the following two lemmas. The first shows that the product space operator is monotone whenever F is. The second shows that the solution sets to the two VIs coincide after applying Q^- to map down from $\overline{\mathcal{X}}$ to $\mathcal{X}$.

Lemma 5.1. *If $F\colon \mathcal{X} \to \mathcal{X}$ is α-cocoercive, then $Q^+ \circ F \circ Q^-$ on $\overline{\mathcal{X}}$ is (α/K)-cocoercive.*

Proof. Fix any $\overline{x}, \overline{y} \in \overline{\mathcal{X}}$, and set $R_{\overline{x}} \triangleq (F \circ Q^-)(\overline{x})$ and $R_{\overline{y}} \triangleq (F \circ Q^-)(\overline{y})$. Then, observe that

$$\big\langle Q^+(R_{\overline{x}}) - Q^+(R_{\overline{y}}), \overline{x} - \overline{y} \big\rangle_{\overline{\mathcal{X}}} = \sum_{k=1}^{K} \langle R_{\overline{x}} - R_{\overline{y}}, \overline{x}_k - \overline{y}_k \rangle \tag{5.8a}$$

$$= \big\langle R_{\overline{x}} - R_{\overline{y}}, Q^-(\overline{x}) - Q^-(\overline{y}) \big\rangle. \tag{5.8b}$$

Substituting in the definition of $R_{\overline{x}}$ and $R_{\overline{y}}$ reveals that

$$\big\langle Q^+(R_{\overline{x}}) - Q^+(R_{\overline{y}}), \overline{x} - \overline{y} \big\rangle_{\overline{\mathcal{X}}} = \big\langle F(Q^-(\overline{x})) - F(Q^-(\overline{y})), Q^-(\overline{x}) - Q^-(\overline{y}) \big\rangle \tag{5.9a}$$

$$\geq \alpha \|F(Q^-(\overline{x})) - F(Q^-(\overline{y}))\|^2 \tag{5.9b}$$

$$= \frac{\alpha}{K} \|Q^+ \circ F \circ Q^-(\overline{x}) - Q^+ \circ F \circ Q^-(\overline{y})\|_{\overline{\mathcal{X}}}^2, \tag{5.9c}$$

where the final equality follows from the definition of the norm on $\overline{\mathcal{X}}$. Because (5.9) holds for arbitrary $\overline{x}, \overline{y} \in \overline{\mathcal{X}}$, the result follows. ∎

Proposition 5.2. *For $F\colon \mathcal{X} \to \mathcal{X}$, $\overline{x}^\circ \in \mathrm{VI}(Q^+ \circ F \circ Q^-, \mathcal{A})$ if and only if $Q^-(\overline{x}^\circ) \in \mathrm{VI}(F, \mathcal{C})$.*

Algorithm 5.1. N-FPN (Minkowski sum constraints $\mathcal{C} = \mathcal{C}_1 + \cdots + \mathcal{C}_K$).

$\mathcal{N}_\Theta(d)$: ◁ Input data are d
  $n \leftarrow 1$ ◁ Initialize counter
  for $k = 1, 2, \ldots, K$
    $\overline{z}_k^1 \leftarrow \hat{z}$ ◁ Initialize iterates to $\hat{z} \in \mathcal{X}$
  while $\sum_{k=1}^K \|\overline{z}_k^n - \overline{z}_k^{n-1}\| > \varepsilon$ **or** $n = 1$ ◁ Loop until convergence at fixed point
    for $k = 1, 2, \ldots, K$ ◁ Loop over constraints $\mathcal{C}_k^1$
      $\overline{x}_k^{n+1} \leftarrow P_{\mathcal{C}_k^1}(\overline{z}_k^n)$ ◁ Project onto constraint set
    $v^{n+1} \leftarrow \sum_{k=1}^K \overline{x}_k^{n+1}$ ◁ Combine projections
    for $k = 1, 2, \ldots, K$ ◁ Loop over constraints $\mathcal{C}_k^2$
      $\overline{y}_k^{n+1} \leftarrow P_{\mathcal{C}_k^2}(2\overline{x}_k^{n+1} - \overline{z}_k^n - \alpha F_\Theta(v^{n+1}; d))$ ◁ Blockwise project reflected gradients
      $\overline{z}_k^{n+1} \leftarrow \overline{z}_k^n - \overline{x}_k^{n+1} + \overline{y}_k^{n+1}$ ◁ Apply blockwise updates
    $n \leftarrow n + 1$ ◁ Increment counter
return v^n ◁ Output inference

Proof. Fix $\overline{y} \in \mathcal{A}$ and $\overline{x}^\circ \in \mathrm{VI}(Q^+ \circ F \circ Q^-, \mathcal{A})$. Similarly to the proof of Lemma 5.1, observe that

$$\left\langle (Q^+ \circ F \circ Q^-)(\overline{x}^\circ), \overline{y} - \overline{x}^\circ \right\rangle_{\overline{\mathcal{X}}} = \sum_{k=1}^K \left\langle (F \circ Q^-)(\overline{x}^\circ), \overline{y}_k - \overline{x}_k^\circ \right\rangle \tag{5.10a}$$

$$= \left\langle F(Q^-(\overline{x}^\circ)), Q^-(\overline{y}) - Q^-(\overline{x}^\circ) \right\rangle. \tag{5.10b}$$

Because $Q^-(\mathcal{A}) = \mathcal{C}$, it follows that $x^\circ \triangleq Q^-(\overline{x}^\circ) \in \mathcal{C}$ and $w \triangleq Q^-(\overline{y}) \in \mathcal{C}$. Consequently,

$$0 \le \left\langle (Q^+ \circ F \circ Q^-)(\overline{x}^\circ), \overline{y} - \overline{x}^\circ \right\rangle_{\overline{\mathcal{X}}} = \left\langle F(x^\circ), w - x^\circ \right\rangle. \tag{5.11}$$

Because $\overline{y}$ was arbitrarily chosen, (5.11) holds for all $w \in \mathcal{C}$, and thus, $Q^-(\overline{x}^\circ) \in \mathrm{VI}(F, \mathcal{C})$.

Conversely, fix $\overline{y} \in \mathcal{A}$ and $\overline{x}^\circ \in \overline{\mathcal{X}}$ such that $Q^-(\overline{x}^\circ) \in \mathrm{VI}(F, \mathcal{C})$. Then, $Q^-(\overline{y}) \in \mathcal{C}$ and

$$0 \le \left\langle F(Q^-(\overline{x}^\circ)), Q^-(\overline{y}) - Q^-(\overline{x}^\circ) \right\rangle \tag{5.12a}$$

$$= \sum_{k=1}^K \left\langle F(Q^-(\overline{x}^\circ)), \overline{y}_k - \overline{x}_k^\circ \right\rangle \tag{5.12b}$$

$$= \left\langle (Q^+ \circ F \circ Q^-)(\overline{x}^\circ)), \overline{y} - \overline{x}^\circ \right\rangle_{\overline{\mathcal{X}}}. \tag{5.12c}$$

Together, the inequality (5.12) and the fact that $\overline{y} \in \mathcal{A}$ was arbitrarily chosen imply that $\overline{x}^\circ \in \mathrm{VI}(Q^+ \circ F \circ Q^-, \mathcal{A})$. This completes the proof. ∎

6. Related works. There are two distinct learning problems for games. The first considers repeated rounds of the same game and operates from the agent's perspective. The agents are assumed to have imperfect knowledge of the game, and the goal is to learn the optimal strategy (i.e., the NE or a coarse correlated equilibrium) given only the cost incurred in each round. This problem is not investigated in this work, and we refer the reader to [25, 63, 61] for further details.

The second problem supposes historical observations of agents' behavior are available to an external observer. For example, [54, 34, 1] posit a simple functional form of the agent's cost functions u_k that depend linearly on a set of unknown parameters. Assuming that a set of noisy observations of the equilibrium[9] $x^\star$ is observed, a regression problem can be formulated and solved to obtain an estimate of these parameters. These works do not consider cost functions depending on the context d. A similar approach is pursued in [66], except, instead of attempting to estimate the unknown parameters in the agent's cost functions, an equilibrium x° is predicted, which explains the agent's behavior for all possible values of these unknown parameters. Small-scale traffic routing problems are considered. [11, 70, 69, 71] are important precursors to our work. Similar to us, they view the primary object of study as a variational inequality with unknown F. Given noisy observations of solutions to this variational inequality, [11] proposes a nonparametric, kernel-based method for approximating F. This method is applied in [70, 69, 71] to noncontextual traffic routing problems on road networks discussed in section 7.2. While it is conceivable that this method could be extended to contextual games, to the best of the author's knowledge, this has not yet been done.

Several recent works [40, 41, 39] consider data consisting of pairs of contexts d and equilibria $x_d^\star$ of the contextual game parameterized by d and employ techniques from contemporary deep learning. Crucially, [40] is the first paper to propose a differentiable game solver—which we refer to as Payoff-Net—allowing for end-to-end training of a neural network that predicts $x_d^\star$ given d. Abstractly, the output of Payoff-Net is defined as the NE of the game

$$\min_{x_1\in\Delta_n}\min_{x_2\in\Delta_n} x_1^\top B_\Theta(d)x_2 - \sum_j x_{1,j}\log(x_{1,j}) + \sum_j x_{2,j}\log(x_{2,j}), \tag{6.1}$$

where $B_\Theta(d)$ is a neural network whose output is an antisymmetric $n\times n$ matrix, while Δ_n is the n-probability simplex (see Remark 3.2 for further discussion on the role of the entropic regularizers). The KKT conditions for (6.1) are

$$\begin{aligned} B_\Theta(d)x_2 + \log(x_1) + 1 + \mu 1 &= 0,\\ B_\Theta(d)^\top x_1 - \log(x_2) - 1 + \nu 1 &= 0,\\ 1^\top x_1 &= 1,\\ 1^\top x_2 &= 1, \end{aligned} \tag{6.2}$$

where 1 (respectively, 0) represents the all-ones (respectively, all-zeros) vector of appropriate dimension and log is applied elementwise. The forward pass of Payoff-Net applies Newton's method to (6.2), at a cost of $\mathcal{O}(n^3)$ per iteration (see [2] for further discussion on this complexity). Differentiating (6.2) with respect to B_Θ yields a linear system that may be solved for $\frac{\mathrm{d}x_d^\star}{\mathrm{d}B_\Theta}$ at a cost of $\mathcal{O}(n^3)$. This is done on the backward pass of Payoff-Net. From $\frac{\mathrm{d}x_d^\star}{\mathrm{d}B_\Theta}$, one may compute $\frac{\mathrm{d}x_d^\star}{\mathrm{d}\Theta}$ via the chain rule. We highlight that, by construction, Payoff-Net can only be applied to two-player, zero-sum games with $\mathcal{C}=\Delta_n\times\Delta_n$. In [41], this approach was modified, leading to a faster backpropagation algorithm, but only for two-player, zero-sum games with

[9] Some of the aforementioned work considers equilibria other than Nash; e.g., [66] considers a correlated equilibrium while [1] considers generalized NEs, leading to additional technical challenges.

$\mathcal{C} = \Delta_n \times \Delta_n$ that admit a compact extensive form representation. In [39], a differentiable variational inequality layer (VI-Layer), similar to (4.1), is proposed. Using the equivalence (4.4), they convert the problem of training this VI-Layer to that of tuning a parameterized operator $F_\Theta(\cdot;\cdot)$ such that

$$x_d^\star \approx P_{\mathcal{C}}(x_d^\star - F_\Theta(x_d^\star; d)).$$

This idea is a significant step forward because it extends the approach of [40] to unregularized games with arbitrary $\mathcal{C}$ and an arbitrary number of players. It also extends [11] by connecting their approach with the techniques of deep learning. However, because [39] does not use constraint decoupling (see Lemma 4.2 and section 5), they are forced to use an iterative $\mathcal{O}(\dim(\mathcal{C})^3)$ algorithm [2] to compute $P_{\mathcal{C}}$ (resp., $\mathrm{d}P_{\mathcal{C}}/\mathrm{d}z$) in every forward (resp., backward) pass, as compared to the $\mathcal{O}(\dim(\mathcal{C})^2)$ cost of N-FPN. When $F_\Theta(\cdot;\cdot)$ is a multilayer neural network, tuning Θ might require millions of forward and backward passes. Thus, their approach is impractical for games with even moderately large $\mathcal{C}$ (see section 3.3 of [2]). Since the arXiv version of this work [30] appeared, the use of N-FPNs for contextual traffic routing has been furthered by [43], where origin-destination (OD)–pair (see subsection 7.2)–specific contextual dependencies are considered.

Our N-FPN architecture, particularly the use of operator-splitting techniques, leverages insights from projection methods, which, in Euclidean spaces, date back to the 1930s [16, 32]. Projection methods are well-suited to large-scale problems because they are built from projections onto individual sets, which are often easy to compute; see [13, 14] and the references therein. Finally, we note that the learning problem (4.3) is an example of a mathematical program with equilibrium constraints [44]. In this context, the difficulty of "differentiating through" the fixed point z_d° (see (4.16)) is well-known, and we refer the reader to [38] for further discussion on computing this derivative, as well as an alternative approach for doing so.

7. Numerical examples.

We show the efficacy of N-FPNs on two classes of contextual games: matrix games and traffic routing.

7.1. Contextual matrix games.

In [40], the Payoff-Net architecture is used for a contextual "rock-paper-scissors" game. This is a (symmetric) matrix game where both players have action sets of dimension 3. We extend this experiment to higher-dimensional action sets. Note that [40] considers *entropy-regularized* cost functions[10]

$$\begin{aligned} u_1(x;d) &= x_2^\top B(d) x_1 + \sum_i x_{1,i} \log(x_{1,i}), \\ u_2(x;d) &= -x_2^\top B(d) x_1 + \sum_i x_{2,i} \log(x_{2,i}) \end{aligned} \tag{7.1}$$

for antisymmetric contextual cost matrix $B(d) \in \mathbb{R}^{a \times a}$, thus guaranteeing that the game satisfies Assumptions (A1), (A2), (A3), (A4), and (A5), particularly (A4). We do the same here. Each player's set of mixed strategies is the probability simplex Δ_a so that $\mathcal{C} = \Delta_a \times \Delta_a$. We vary a in multiples of 10 from 20 to 120. For each a, we generate a training dataset

[10] Equivalently, they determine the QRE, not the NE; see Remark 3.2.

$\{(d^i, x^\star_{d^i})\}_{i=1}^{2000}$ and train a Payoff-Net, an N-FPN constrained to be cocoercive using (4.13), and an unconstrained N-FPN with comparable numbers of parameters for 100 epochs or until the test loss is below 10^{-5}, whichever comes first. See the supplemental material (supplement.pdf [local/web 437KB]) for further architectural details. The results are presented in Figure 2.

Payoff-Net achieves the target test loss in much fewer epochs than (either version of) the N-FPN. We attribute this to the use of Newton's method on the forward pass (which approximates the NE to higher precision), as well as the use of the true gradient on the backward pass. However, the time Payoff-Net requires to complete an epoch grows exponentially with the size of $\mathcal{C}$ (see Figure 2). Hence, for larger $\mathcal{C}$, it is one to two orders of magnitude faster to train an N-FPN to the desired test loss.

For illustration, we simulate play in the unregularized (i.e., without the entropic term in (7.1)) matrix game between two agents over a test set of contexts d. The first agent has full access to $B(d)$ and plays according to the computed NE. Four options are used for the second agent:

- An N-FPN agent, who plays the strategy provided by a trained (unconstrained) N-FPN given d.
- A Payoff-Net agent, who plays the strategy provided by a trained Payoff-Net given d.
- A data-agnostic agent, who plays the uniform strategy (i.e., each action is selected with equal probability) regardless of d.
- An optimal agent, who has full access to $B(d)$ and plays according to the computed NE.

We plot the absolute value of the mean cost averaged over all 1000 trials per d and all d for a given action set size a. The results are illustrated in Figure 3. Because this game is zero-sum, the expected mean cost is zero. Over all a, the N-FPN outperforms Payoff-Net. We attribute this to the fact that Payoff-Net explicitly incorporates the entropic regularizer into its architecture (see section 6), whereas the unconstrained N-FPN does not.

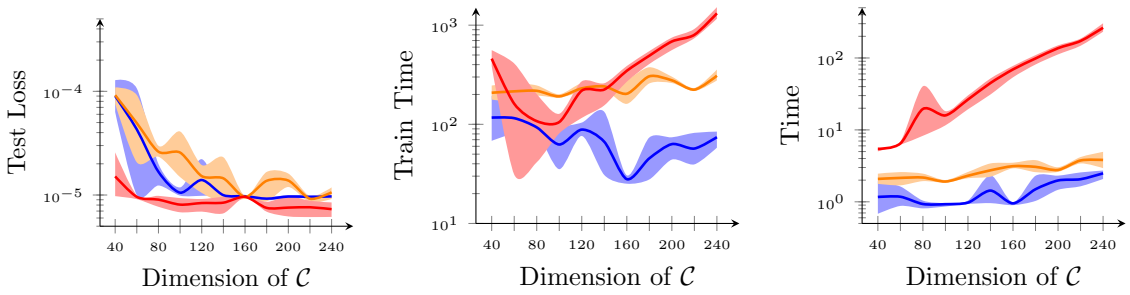

Figure 2. *Final test loss (left), total training time (center), and mean training time per epoch (right) for Payoff-Net (shown in red), a cocoercive N-FPN (shown in orange), and an unconstrained N-FPN (shown in blue). Each network is trained for* 100 *epochs or until a test loss less than* 10^{-5} *is achieved. The final test loss decreases as a function of* a*, which is expected since the number of parameters increases with* a*. Note that, in this experiment, the form of an N-FPN without three-operator splitting (i.e., Algorithm 4.2) is used, and so, the speed-up in train time observed is attributable to the fact that the N-FPN uses fixed-point iteration for forward propagation and JFB for backward propagation, while Payoff-Net uses Newton's method on the forward pass and solves (4.16) on the backward pass.*

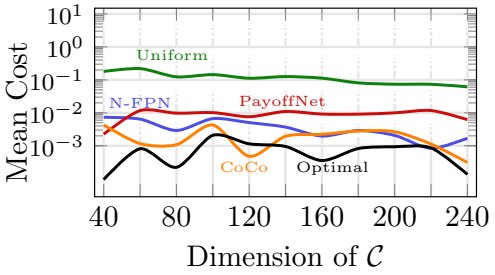

Figure 3. *Simulated play for matrix games of increasing size. Here, "N-FPN" refers to the unconstrained variant, while "CoCo" refers to the cocoercive variant of N-FPN.*

7.2. Contextual traffic routing.

Setup. Consider a road network represented by a directed graph with vertices V and arcs E. Let $N \in \mathbb{R}^{|V|\times|E|}$ denote the vertex–arc incidence matrix defined by

$$N_{ij} \triangleq \begin{cases} +1 & \text{if } (i,j) \in E, \\ -1 & \text{if } (j,i) \in E, \\ 0, & \text{otherwise.} \end{cases} \tag{7.2}$$

For example, for the simple road network shown in Figure 4, the incidence matrix is

$$N = \begin{bmatrix} -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 \\ 1 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{7.3}$$

An OD-pair is a triple (v_1, v_2, q) with $v_i \in V$ and $q \in \mathbb{R}_{>0}$, encoding the constraint of routing q units of traffic from v_1 to v_2. Each OD-pair is encoded by a vector $b \in \mathbb{R}^{|V|}$ with $b_{v_1} = -q$, $b_{v_2} = q$ and all other entries zero. A valid *traffic flow* $x \in \mathbb{R}^{|E|}$ for an OD-pair has nonnegative entries satisfying the flow equation $Nx = b$. The eth entry x_e represents the traffic density along the eth arc. The flow equation ensures that the number of cars entering an intersection equals the number leaving, except for a net movement of q units of traffic from v_1 to v_2. For K OD-pairs, a valid traffic flow x is the sum of traffic flows for each OD-pair, which is in the Minkowski sum

$$\mathcal{C} = \sum_{k=1}^{K} \mathcal{C}_k \triangleq \underbrace{\{x : Nx = b_k\}}_{\mathcal{C}_k^1} \bigcap \underbrace{\{x : x \geq 0\}}_{\mathcal{C}_k^2}. \tag{7.4}$$

A *contextual travel time function* $t_e(x_e; d)$ is associated with each arc, where d encodes contextual data. This function increases monotonically with x_e, reflecting the fact that increased congestion leads to longer travel times. The context d encodes exogenous factors—weather, construction, and so on. Here, the equilibrium of interest is, roughly speaking, a flow configuration $x_d^\star$, where the travel time between each OD-pair is as short as possible when taking into account congestion effects [12]. This is known as a *Wardrop equilibrium* (also called the *user equilibrium*) [65], a special case of NEs where $F = [t_1(x_1; d)^\top \cdots t_{|E|}(x_{|E|}; d)^\top]^\top$. In certain cases, a Wardrop equilibrium is the limit of a sequence of NEs as the number of drivers goes to infinity [26].

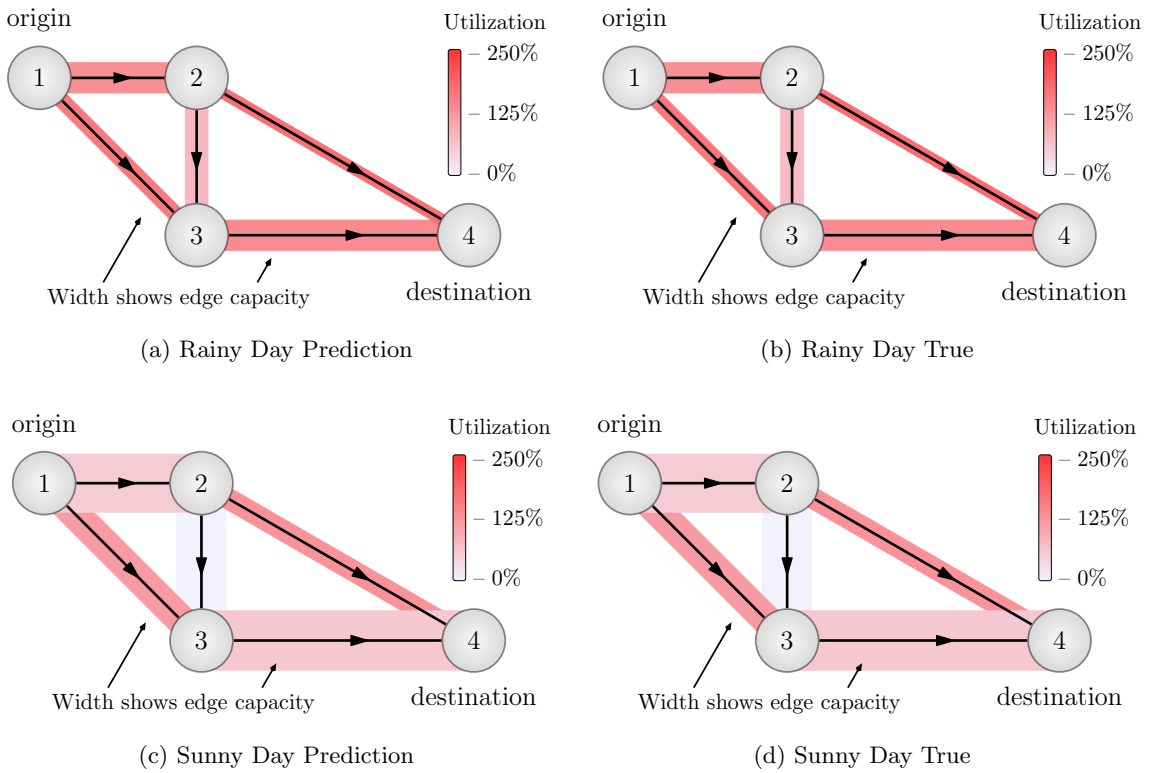

Figure 4. (a) *True traffic flow for "rainy" context.* (b) *Predicted traffic by $\mathcal{N}_\Theta$ for "rainy" context.* (c) *True traffic flow for "sunny" context.* (d) *Predicted traffic by $\mathcal{N}_\Theta$ for "sunny" context.*

TRAFIX scores. Accuracy of traffic routing predictions are measured by a TRAFIX score. This score forms an intuitive alternative to mean squared error. An error tolerance $\varepsilon > 0$ is chosen (n.b. $\varepsilon = 5 \times 10^{-3}$ in our experiments). For an estimate x of $x^\star$, the TRAFIX score with parameter ε is the percentage of edges for which x has relative error (with tolerance[11] $\tau > 0$) less than ε; i.e.,

$$\text{(rel. error of edge } e) \triangleq \frac{|x_e - x_e^\star|}{|x_e^\star| + \tau},$$
$$\text{TRAFIX}(x, x^\star; \varepsilon, \tau) \triangleq \frac{(\#\text{ edges with rel. error} < \varepsilon)}{(\#\text{ edges})} \times 100\%.$$

Our plots and Tables 1, 2, and 3 show the expected TRAFIX scores over the distributions of testing data.

Datasets and Training. We are unaware of any prior datasets for contextual traffic routing, and so, we construct our own. First, we construct a toy example based on the "Braess paradox" network studied in [39], illustrated in Figure 4. Here, $d \in \mathbb{R}^5$; see supplementary materials (supplement.pdf [local/web 437KB]) for further details.

[11] The parameter τ is added to handle the case when the eth component of $x^\star$ is zero; i.e., $x_e^\star = 0$.

Table 1

Comparison of different equilibria prediction methods. Analytic modeling algorithms yield game equilibria that are not data-driven. Traditional feedforward networks are data-driven and easy to train but are incapable of outputting a game equilibrium. Existing game-based implicit models are nontrivial to train (backpropagate) and require intricate forward propagation.

Attribute	Analytic	Feedforward	[39], [40]	Proposed N-FPNs
Output is equilibria	✓		✓	✓
Data-driven		✓	✓	✓
Constraint Decoupling	✓			✓
Simple backprop	NA	✓		✓

Table 2

Datasets used. First and second columns show the number of edges, nodes, and OD-pairs for corresponding dataset. Second column shows the number of tunable parameters. Further details may be found in supplementary materials (supplement.pdf [local/web 437KB]).

Dataset	Edges/nodes	OD-pairs	# params
Sioux Falls	76/24	528	46K
Eastern Massachusetts	258/74	1113	99K
Berlin-Friedrichshain	523/224	506	179K
Berlin-Tiergarten	766/361	644	253K
Anaheim	914/416	1406	307K
Chicago-Sketch	2950/933	93513	457K

Table 3

Results of traffic routing experiments. To benchmark our results, we provide comparison with a traditional neural network architecture. To make a fair comparison, we use the same architecture for F_Θ in N-FPNs and the feedforward neural network.

	MSE		TRAFIX	
Dataset	N-FPN	Feedforward	N-FPN	Feedforward
Sioux Falls	1.9×10^{-3}	5.4×10^{-3}	94.42%	70.16%
Eastern Massachusetts	4.7×10^{-4}	4.1×10^{-3}	97.94%	92.70%
Berlin-Friedrichshain	5.3×10^{-4}	9.3×10^{-4}	97.42%	97.94%
Berlin-Tiergarten	7.6×10^{-4}	5.5×10^{-4}	95.95%	97.03%
Anaheim	2.4×10^{-3}	5.1×10^{-2}	95.28%	58.57%
Chicago-Sketch	2.5×10^{-3}	3.1×10^{-3}	98.81%	97.12%

We also constructed contextual traffic routing datasets based on road networks of real-world cities curated by the Transportation Networks for Research Project [62]. We did so by fixing a choice of $t_e(x; d)$ for each arc e, randomly generating a large set of contexts $d \in [0, 1]^{10}$, and then, for each d, finding a solution $x_d^\circ \in \mathrm{VI}(F(\cdot\,; d), \mathcal{C})$. Table 2 shows a description of the traffic networks datasets, including the numbers of edges, nodes, and OD-pairs. Further details are in the supplemental materials (supplement.pdf [local/web 437KB]). We emphasize that, for these contextual games, the structure of $\mathcal{C}$ is complex; it is a Minkowski sum of hundreds of high-dimensional polytopes (recall (7.4)). We train an N-FPN using the constraint decoupling described in section 5 for forward propagation (see Algorithm 5.1) to predict $x_d^\star$ from d for

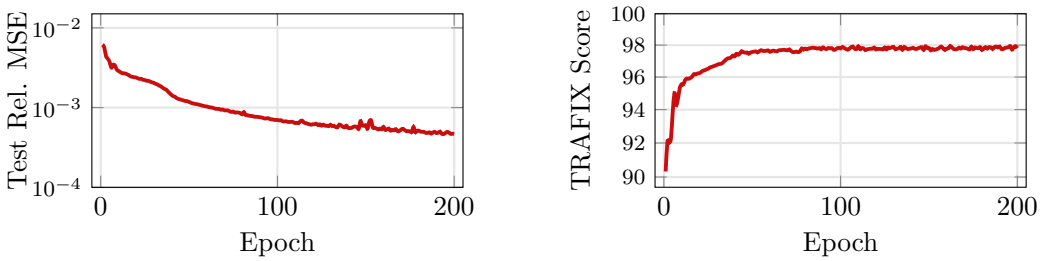

Figure 5. *Plots for N-FPN performance on Eastern Massachusetts testing data. The first plot shows convergence of expected relative mean squared error on testing data after each training epoch, and the second plot shows the expected TRAFIX score on testing data after each training epoch.*

each dataset with architectures as described in the appendix. Additional training details are in the appendix. For comparison, we also train a traditional feedforward neural network. We use the same architecture used to parameterize the game gradient in N-FPNs and use the same number of epochs during training. We perform a logarithmic search when tuning the learning rate.

Results. As illustrated in Figure 4, the N-FPN almost perfectly predicts the resulting Wardrop equilibrium given only the context d. The results for the real-world networks are shown in the final two columns of Table 2. The convergence during training of the relative Mean Squared Error (MSE) and TRAFIX score on the Eastern Massachusetts testing dataset is shown in Figure 5. Additional plots can be found in the supplementary materials (supplement.pdf [local/web 437KB]).

8. Conclusions.

The fusion of big data and optimization algorithms offers potential for predicting equilibria in systems with many interacting agents. The proposed N-FPNs form a scalable data-driven framework for efficiently predicting equilibria for such systems that can be modeled as contextual games. The N-FPN architecture yields equilibria outputs that satisfy constraints while also being trained end-to-end. Moreover, the provided constraint decoupling schemes enable simple forward and backward propagation using explicit formulae for each projection. The efficacy of N-FPNs is illustrated on large-scale traffic routing problems using a contextual traffic routing benchmark dataset and TRAFIX scoring system. Although we focus here on games, we note that N-FPNs are equally applicable to any system modeled using a variational inequality (or, equivalently, a linear complementarity problem), for example, convex optimization [2, 67] or physical simulation [19]. Future work shall focus on end-to-end learning for these domains using N-FPN.

REFERENCES

[1] S. Allen, J. P. Dickerson, and S. A. Gabriel, *Using Inverse Optimization to Learn Cost Functions in Generalized Nash Games*, preprint, arXiv:2102.12415, 2021.

[2] B. Amos and J. Z. Kolter, *OptNet: Differentiable optimization as a layer in neural networks*, in International Conference on Machine Learning, PMLR, 2017, pp. 136–145.

[3] B. Amos, L. Xu, and J. Z. Kolter, *Input convex neural networks*, in International Conference on Machine Learning, PMLR, 2017, pp. 146–155.

[4] K. J. Arrow and G. Debreu, *Existence of an equilibrium for a competitive economy*, Econometrica, 22 (1954), pp. 265–290.

[5] O. H. Azar and M. Bar-Eli, *Do soccer players play the mixed-strategy Nash equilibrium?*, Appl. Economics, 43 (2011), pp. 3591–3601.
[6] S. Bai, J. Z. Kolter, and V. Koltun, *Deep equilibrium models*, in Advances in Neural Information Processing Systems, 2019, pp. 690–701.
[7] S. Bai, V. Koltun, and J. Z. Kolter, *Multiscale deep equilibrium models*, Adv. Neural Inf. Process. Syst., 33 (2020), pp. 5238–5250.
[8] J.-B. Baillon and G. Haddad, *Quelques propriétés des opérateurs angle-bornés et n-cycliquement monotones*, Israel J. Math., 26 (1977), pp. 137–150.
[9] H. H. Bauschke and P. L. Combettes, *The Baillon–Haddad theorem revisited*, J. Convex Anal., 17 (2010), pp. 781–787.
[10] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2nd ed., Springer, Cham, 2017.
[11] D. Bertsimas, V. Gupta, and I. C. Paschalidis, *Data-driven estimation in equilibrium using inverse optimization*, Math. Program., 153 (2015), pp. 595–633.
[12] G. Carlier and F. Santambrogio, *A continuous theory of traffic congestion and Wardrop equilibria*, J. Math. Sci., 181 (2012), pp. 792–804.
[13] Y. Censor and A. Cegielski, *Projection methods: An annotated bibliography of books and reviews*, Optimization, 64 (2015), pp. 2343–2358, https://doi.org/10.1080/02331934.2014.957701.
[14] Y. Censor, W. Chen, P. L. Combettes, R. Davidi, and G. T. Herman, *On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints*, Comput. Optim. Appl., 51 (2012), pp. 1065–1088.
[15] T. Chen, X. Chen, W. Chen, H. Heaton, J. Liu, Z. Wang, and W. Yin, *Learning to Optimize: A Primer and a Benchmark*, preprint, arXiv:2103.12828, 2021.
[16] G. Cimmino, *Cacolo approssimato per le soluzioni dei systemi di equazioni lineari*, La Ricerca Scientifica (Roma), 1 (1938), pp. 326–333.
[17] S. Dafermos, *Sensitivity analysis in variational inequalities*, Math. Oper. Res., 13 (1988), pp. 421–434.
[18] D. Davis and W. Yin, *A three-operator splitting scheme and its optimization applications*, Set-Valued Var. Anal., 25 (2017), pp. 829–858.
[19] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter, *End-to-end differentiable physics for learning and control*, Adv. Neural Inf. Process. Syst., 31 (2018), pp. 7178–7189.
[20] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer, New York, 2007.
[21] S. W. Fung, H. Heaton, Q. Li, D. McKenzie, S. Osher, and W. Yin, *JFB: Jacobian-Free Backpropagation for implicit networks*, in Proceedings of the AAAI Conference on Artificial Intelligence, 36 (2022), pp. 6648–6656.
[22] Z. Geng, X.-Y. Zhang, S. Bai, Y. Wang, and Z. Lin, *On training implicit models*, Adv. Neural Inf. Process. Syst., 34 (2021), pp. 24247–24260.
[23] L. E. Ghaoui, F. Gu, B. Travacca, A. Askari, and A. Y. Tsai, *Implicit Deep Learning*, preprint, arXiv:1908.06315, 2019.
[24] D. Gilton, G. Ongie, and R. Willett, *Deep Equilibrium Architectures for Inverse Problems in Imaging*, preprint, arXiv:2102.07944, 2021.
[25] J. Hannan, *Approximation to Bayes risk in repeated play*, Contrib. Theory Games, 21 (1957), 97.
[26] A. Haurie and P. Marcotte, *On the relationship between Nash–Cournot and Wardrop equilibria*, Networks, 15 (1985), pp. 295–308.
[27] H. Heaton and S. W. Fung, *Explainable AI via Learning to Optimize*, preprint, arXiv:2204.14174, 2022.
[28] H. Heaton, S. W. Fung, A. Gibali, and W. Yin, *Feasibility-Based Fixed Point Networks*, preprint, arXiv:2104.14090, 2021.
[29] H. Heaton, S. W. Fung, A. T. Lin, S. Osher, and W. Yin, *Wasserstein-based projections with applications to inverse problems*, SIAM J. Math. Data Sci., 4 (2022), pp. 581–603.
[30] H. Heaton, D. McKenzie, Q. Li, S. W. Fung, S. Osher, and W. Yin, *Learn to Predict Equilibria via Fixed Point Networks*, preprint, arXiv:2106.00906v1, 2021.

[31] H. W. Heaton, *Learning to Optimize with Guarantees*, Ph.D. thesis, University of California, Los Angeles, 2021.
[32] S. Karczmarz, *Angenäherte auflösung von systemen linearer Gleichungen*, Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat., (1937), pp. 355–357.
[33] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, International Conference on Learning Representations (ICLR) (Poster), (2015).
[34] I. C. Konstantakopoulos, L. J. Ratliff, M. Jin, C. Spanos, and S. S. Sastry, *Smart building energy efficiency via social game: A robust utility learning framework for closing–the–loop*, in 2016 1st International Workshop on Science of Smart City Operations and Platforms Engineering (SCOPE) in Partnership with Global City Teams Challenge (GCTC) (SCOPE-GCTC), IEEE, 2016, pp. 1–6.
[35] J. Kotary, F. Fioretto, P. Van Hentenryck, and B. Wilder, *End-to-End Constrained Optimization Learning: A Survey*, preprint, arXiv:2103.16378, 2021.
[36] Y. Koyama, N. Murata, S. Uhlich, G. Fabbro, S. Takahashi, and Y. Mitsufuji, *Music source separation with deep equilibrium models*, in ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 296–300.
[37] S. G. Krantz and H. R. Parks, *The Implicit Function Theorem: History, Theory, and Applications*, Springer, New York, 2012.
[38] J. Li, J. Yu, B. Liu, Z. Wang, and Y. M. Nie, *Achieving Hierarchy-Free Approximation for Bilevel Programs with Equilibrium Constraints*, preprint, arXiv:2302.09734, 2023.
[39] J. Li, J. Yu, Y. Nie, and Z. Wang, *End-to-end learning and intervention in games*, Adv. Neural Inf. Process. Syst., 33 (2020), pp. 16653–16665.
[40] C. K. Ling, F. Fang, and J. Z. Kolter, *What Game Are We Playing? End-to-End Learning in Normal and Extensive Form Games*, preprint, arXiv:1805.02777, 2018.
[41] C. K. Ling, F. Fang, and J. Z. Kolter, *Large scale learning of agent rationality in two-player zero-sum games*, in Proceedings of the AAAI Conference on Artificial Intelligence 33, 2019, pp. 6104–6111.
[42] B. Liu, J. Li, Z. Yang, H.-T. Wai, M. Hong, Y. Nie, and Z. Wang, *Inducing equilibria via incentives: Simultaneous design-and-play ensures global convergence*, Adv. Neural Inf. Process. Syst., 35 (2022), pp. 29001–29013.
[43] Z. Liu, Y. Yin, F. Bai, and D. K. Grimm, *End-to-end learning of user equilibrium with implicit neural networks*, Transp. Res. Part C Emerg. Technol., 150 (2023), 104085.
[44] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, 1996.
[45] P. Marcotte and J. H. Wu, *On the convergence of projection methods: Application to the decomposition of affine variational inequalities*, J. Optim. Theory Appl., 85 (1995), pp. 347–362.
[46] R. D. McKelvey and T. R. Palfrey, *Quantal response equilibria for normal form games*, Game. Econ. Behav., 10 (1995), pp. 6–38.
[47] D. McKenzie, S. W. Fung, and H. Heaton, *Faster Predict-and-Optimize with Three-Operator Splitting*, preprint, arXiv:2301.13395v1, 2023.
[48] P. Mertikopoulos and W. H. Sandholm, *Learning in games via reinforcement and regularization*, Math. Oper. Res., 41 (2016), pp. 1297–1324.
[49] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, *Spectral Normalization for Generative Adversarial Networks*, preprint, arXiv:1802.05957, 2018.
[50] J. F. Nash, *Equilibrium points in n-person games*, Proc. Natl. Acad. Sci., 36 (1950), pp. 48–49.
[51] F. Pedregosa and G. Gidel, *Adaptive three operator splitting*, in International Conference on Machine Learning, PMLR, 2018, pp. 4085–4094.
[52] J.-C. Pesquet, A. Repetti, M. Terris, and Y. Wiaux, *Learning maximally monotone operators for image recovery*, SIAM J. Imaging Sci., 14 (2021), pp. 1206–1237.
[53] Z. Ramzi, F. Mannel, S. Bai, J.-L. Starck, P. Ciuciu, and T. Moreau, *SHINE: Sharing the inverse estimate from the forward pass for bi-level optimization and implicit models*, in ICLR 2022-International Conference on Learning Representations, 2022.
[54] L. J. Ratliff, M. Jin, I. C. Konstantakopoulos, C. Spanos, and S. S. Sastry, *Social game for building energy efficiency*, in Incentive design, in 2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), IEEE, 2014, pp. 1011–1018.
[55] R. T. Rockafellar, *Convex Analysis*, Princeton Math. Ser. 36, Princeton University Press, Princeton, NJ, 1970.

[56] Y. Romano, M. Elad, and P. Milanfar, *The little engine that could: Regularization by denoising (RED)*, SIAM J. Imaging Sci., 10 (2017), pp. 1804–1844.
[57] J. B. Rosen, *Existence and uniqueness of equilibrium points for concave n-person games*, Econometrica, (1965), pp. 520–534.
[58] T. Roughgarden, *Routing games*, Algorithmic Game Theory, 18 (2007), pp. 459–484.
[59] E. Ryu and W. Yin, *Large-Scale Convex Optimization: Algorithm Designs via Monotone Operators*, Cambridge University Press, Cambridge, UK, 2022.
[60] T. Salimans and J. Ho, *Should EBMS model the energy or the score?*, in Energy Based Models Workshop-ICLR 2021, 2021.
[61] P. G. Sessa, I. Bogunovic, A. Krause, and M. Kamgarpour, *Contextual games: Multi-agent learning with side information*, Adv. Neural Inf. Process. Syst., 33 (2020), pp. 21912–21922.
[62] B. Stabler, H. Bar-Gera, and E. Sall, *Transportation Networks for Research*, 2016, https://github.com/bstabler/TransportationNetworks. Accessed: 2021-05-24.
[63] G. Stoltz and G. Lugosi, *Learning correlated equilibria in games with compact sets of strategies*, Game. Econ. Behav., 59 (2007), pp. 187–208.
[64] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1999.
[65] J. G. Wardrop, *Some theoretical aspects of road traffic research*, Proc. Inst. Civ. Eng., 1 (1952), pp. 325–362.
[66] K. Waugh, B. D. Ziebart, and J. A. Bagnell, *Computational rationalization: The inverse equilibrium problem*, in Proceedings of the 28th International Conference on International Conference on Machine Learning, 2011, pp. 1169–1176.
[67] B. Wilder, B. Dilkina, and M. Tambe, *Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization*, in Proceedings of the AAAI Conference on Artificial Intelligence 33, 2019, pp. 1658–1665.
[68] E. Winston and J. Z. Kolter, *Monotone operator equilibrium networks*, in Advances in Neural Information Processing Systems 33, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds., Curran Associates, 2020, pp. 10718–10728, https://proceedings.neurips.cc/paper/2020/file/798d1c2813cbdf8bcdb388db0e32d496-Paper.pdf.
[69] J. Zhang and I. C. Paschalidis, *Data-driven estimation of travel latency cost functions via inverse optimization in multi-class transportation networks*, in 2017 IEEE 56th Annual Conference on Decision and Control (CDC), IEEE, 2017, pp. 6295–6300.
[70] J. Zhang, S. Pourazarm, C. G. Cassandras, and I. C. Paschalidis, *The price of anarchy in transportation networks by estimating user cost functions from actual traffic data*, in 2016 IEEE 55th Conference on Decision and Control (CDC), IEEE, 2016, pp. 789–794.
[71] J. Zhang, S. Pourazarm, C. G. Cassandras, and I. C. Paschalidis, *The price of anarchy in transportation networks: Data-driven evaluation and reduction strategies*, Proc. IEEE, 106 (2018), pp. 538–553.
[72] Y. Zhang, D. W. Zhang, S. Lacoste-Julien, G. J. Burghouts, and C. G. Snoek, *Multiset-equivariant set prediction with approximate implicit differentiation*, in International Conference on Learning Representations, 2022.