SOLVING PARAMETRIC VARIATIONAL INEQUALITIES VIA FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Variational inequalities (VIs) provide a unified framework for modeling optimization, equilibrium, and multi-agent decision problems. In many applications, the VI depends on exogenous parameters and must be solved repeatedly across varying instances, giving rise to parametric variational inequalities (PVIs). When solution sets are non-singleton, a single prediction can discard alternatives useful for downstream criteria such as welfare or robustness. To address this, we propose a conditional flow-matching approach that learns distributions over PVI solutions, enabling solution sampling and post-hoc selection for unseen instances. Theoretically, we establish VI-specific guarantees that translate flow-matching error into Wasserstein distributional, feasibility, and stationarity errors, together with finite-sample high-probability bounds that account for approximation, optimization, and generalization errors. Empirically, we evaluate the framework on three complementary benchmarks: Braess traffic equilibria, where it recovers diversity along a non-singleton equilibrium set; nonconvex AC optimal power flow, where it exhibits near-theory-consistent feasibility-error scaling as an out-of-assumption stress test and competitive time–accuracy tradeoffs; and non-monotone bimatrix games, where best-of-M and welfare analyses evaluate flexible equilibrium selection.

1 INTRODUCTION

Variational inequalities (VIs) provide a unified framework for constrained optimization, equilibrium computation, and multi-agent decision making (Kinderlehrer & Stampacchia, 2000; Facchinei & Pang, 2003). Given a feasible set $\mathcal{X}$ and an operator F, a VI seeks $x^\star \in \mathcal{X}$ such that $F(x^\star)^\top(y-x^\star) \geq 0$ for all $y \in \mathcal{X}$, encompassing optimality and equilibrium conditions in optimization, games, control, and economics (Boyd & Vandenberghe, 2004; Bertsekas, 1999; Nagurney, 2013). Classical algorithms such as extragradient and proximal-point methods are well understood under monotonicity and related regularity conditions, but general non-monotone VIs remain challenging (Korpelevich, 1976; Rockafellar, 1976; Bauschke & Combettes, 2017; Beck, 2017).

Many applications require solving a *family* of VIs parameterized by exogenous inputs, such as travel demands, network capacities, or power-system operating conditions (Dafermos, 1980; 1988; Nagurney, 2013). These *parametric variational inequalities* (PVIs) (Kyparisis, 1992) arise naturally when related problem instances must be solved repeatedly. PVIs pose two challenges that motivate our approach. First, a VI may admit multiple solutions, whereas classical solvers typically return one solution per run. Multiple candidates can be important for downstream criteria such as social welfare in games or upper-level performance in bilevel problems (Harsanyi & Selten, 1988; Wiesemann et al., 2013; Besancon et al., 2024; Chen et al., 2026). Second, conventional solvers generally incur fresh online computation whenever a new instance is encountered, so solving many related instances repeatedly pays this per-instance cost. Learning-based approaches instead shift part of this computation offline, exploiting shared structure across instances to accelerate solving at deployment time (Amos, 2023; Sambharya et al., 2024; Fioretto et al., 2020).

We address both challenges through *solution-distribution learning*. Using flow matching (FM) (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2023), we learn a parameter-conditioned distribution over PVI solutions from solution samples generated offline by a solver or data-labeling process, following the supervised paradigm (Amos, 2023; Fioretto et al., 2020; Sambharya et al., 2024; Sharony et al., 2024; Liang & Chen, 2024). At inference time, a new instance

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

is solved by sampling one or more candidate solutions, enabling amortized inference while retaining multiple alternatives when the solution set is non-singleton. Our theoretical analysis connects flow-matching error to Wasserstein distributional error and, through PVI regularity, to approximate feasibility and gap-based VI stationarity. We further bridge finite-sample empirical training to these population-level guarantees, accounting for model approximation, optimization, and generalization errors. We evaluate the framework on three complementary benchmarks. A Braess traffic-equilibrium network with a non-singleton equilibrium set tests recovery of equilibrium diversity. Nonconvex AC optimal power flow (ACOPF) provides a theory-facing scaling check together with time–accuracy tradeoffs against classical and learning-based baselines. Non-monotone bimatrix games evaluate best-of-M and welfare-based post-hoc selection across game families of varying size.

Contributions. (i) We formulate PVI solving as solution-distribution learning with flow matching, covering non-singleton solution sets and equilibrium problems that need not admit an underlying scalar objective. (ii) We establish VI-specific guarantees translating flow-matching error into Wasserstein distributional error, feasibility error, and gap-based stationarity error, and connect these guarantees to finite-sample empirical training through high-probability bounds accounting for model approximation, optimization, and generalization errors. (iii) We demonstrate the framework across three complementary regimes: non-singleton traffic equilibria, nonconvex ACOPF, and non-monotone games, evaluating solution diversity, theory-facing error scaling and time–accuracy tradeoffs, and best-of-M/welfare-based post-hoc selection, respectively.

2 Related Work

Variational inequalities and parametric analysis. VIs are a classical framework for optimization and equilibrium problems, with standard algorithmic foundations in extragradient, proximal-point, and monotone-operator methods (Kinderlehrer & Stampacchia, 2000; Facchinei & Pang, 2003; Korpelevich, 1976; Rockafellar, 1976). In parametric settings, the sensitivity and stability of VI solution maps have long been studied in variational analysis and traffic equilibrium (Dafermos, 1988; Kyparisis, 1992; Dontchev & Rockafellar, 2009). Recent work on optimization with PVI constraints further emphasizes regularity, stationarity, and implicit-gradient computation (Chen et al., 2026). Complementing this line of research, we study how to learn a distribution of PVI solutions.

Learning-based optimization, generative models, and flow matching. Learning-based methods can accelerate repeated optimization through predictions or learned update rules (Amos, 2023). Most approaches are point-prediction methods, which can be restrictive for PVIs with multiple solutions. Recent diffusion- and flow-based models instead represent rich conditional distributions over solutions (Janner et al., 2022; Li et al., 2024; Liang & Chen, 2024). Among these, Liang & Chen (2024) is closest to our distributional viewpoint, using flow matching to learn solution distributions for parametric optimization problems. Our setting differs in two respects. First, a VI operator need not be the gradient of any scalar objective, so objective suboptimality is not generally available as a correctness criterion; our analysis instead controls feasibility and stationarity directly. Second, their population-level flow guarantee does not by itself characterize the data-dependent model returned by finite-sample empirical training. We provide a learning-theoretic bridge from finite-sample empirical training to population flow error and, ultimately, to downstream PVI guarantees. Appendix E gives additional technical discussion. Flow matching is the modeling primitive because it offers a scalable conditional transport framework together with useful error analyses and constrained-domain extensions (Lipman et al., 2023; Benton et al., 2024; Xie et al., 2024; Li et al., 2026).

Equilibrium computation in games and networks. Our experiments also connect to equilibrium computation in games and networked systems. First-order learning dynamics can exhibit recurrent or cyclic behavior in games without suitable structure (Mertikopoulos et al., 2018), while optimistic variants recover convergence under additional conditions (Mertikopoulos et al., 2019; Daskalakis & Panageas, 2018). We consider an alternative method that moves part of the computation offline: our empirical claim is not that the flow-based method universally replaces classical VI algorithms, but that after training, fast repeated sampling can be competitive and can explore diverse solutions through sample selection.

3 Problem Setup and Flow-Based Method

Parametric variational inequalities. Let $w \in \mathcal{W} \subseteq \mathbb{R}^p$ denote a VI instance parameter. For each w, consider an operator $F_w : \mathbb{R}^n \to \mathbb{R}^n$ and feasible set $\mathcal{X}_w \subseteq \mathbb{R}^n$. The corresponding PVI seeks $x^\star \in \mathcal{X}_w$ such that $F_w(x^\star)^\top (y - x^\star) \geq 0$ for all $y \in \mathcal{X}_w$. We denote its solution set by $\mathcal{S}(w) \triangleq \{x \in \mathcal{X}_w : F_w(x)^\top (y - x) \geq 0,\ \forall y \in \mathcal{X}_w\}$. If F_w is continuous and $\mathcal{X}_w$ is nonempty, convex, and compact, then $\mathcal{S}(w)$ is nonempty and compact (Facchinei & Pang, 2003), but need not be a singleton.

Distributional formulation. We learn a solution distribution $p(\cdot \mid w)$ supported on $\mathcal{S}(w)$, accessed through finitely many offline-generated solution samples $X_1 \sim p(\cdot \mid w)$, as in supervised amortized optimization (Fioretto et al., 2020; Sambharya et al., 2024; Liang & Chen, 2024; Sharony et al., 2024). Once learned, a new PVI instance is solved by sampling from this conditional distribution, optionally followed by post-hoc candidate selection.

Flow matching preliminaries. Flow matching constructs a continuous-time transport from a simple base distribution p_0 to a target distribution p_1. To connect with our method, we set $p_1(\cdot \mid w) = p(\cdot \mid w)$. Such a transport can be represented by a time-dependent velocity field $u_t(\cdot\,; w) : \mathbb{R}^n \to \mathbb{R}^n$ whose deterministic flow transports $p_0(\cdot)$ to $p_1(\cdot \mid w)$. More precisely, for each initialization x_0, let $(X_t^{x_0})_{t\in[0,1]}$ solve $\mathrm{d}X_t^{x_0}/\mathrm{d}t = u_t(X_t^{x_0}; w)$, where $X_0^{x_0} = x_0$. If $X_0 \sim p_0$ and $X_t = X_t^{X_0}$, then the law $p_t(\cdot \mid w)$ of X_t evolves according to the continuity equation (Lipman et al., 2023): $\partial_t p_t(x \mid w) + \nabla_x \cdot \big(p_t(x \mid w) u_t(x; w)\big) = 0$, where $p_0(\cdot \mid w) = p_0(\cdot)$ and $p_1(\cdot \mid w) = p(\cdot \mid w)$. Accordingly, our learning task is to approximate this unknown velocity field $u_t(\cdot\,; w)$ from samples. We approximate u_t by a parameterized vector field $u_t^\theta(\cdot\,; w)$. Given the same initialization $\hat{X}_0 = X_0 \sim p_0$, the learned dynamics are defined by the ordinary differential equation (ODE)

$$\mathrm{d}\hat{X}_t/\mathrm{d}t = u_t^\theta(\hat{X}_t; w), \qquad t \in [0, 1]. \tag{1}$$

The induced learned flow map is denoted by $\hat{\Phi}_t^\theta(\cdot\,; w)$, so $\hat{X}_t = \hat{\Phi}_t^\theta(\hat{X}_0; w)$ and the terminal (learned) distribution is $\hat{p}^\theta(\cdot \mid w) \triangleq \hat{\Phi}_1^\theta(\cdot\,; w)_{\#} p_0$, where $\#$ denotes the push-forward operator (Bogachev, 2007). We use X_t for paths generated by u_t and $\hat{X}_t$ for paths generated by u_t^θ.

The ideal flow matching objective learns the unknown velocity field u_t by regressing the model velocity u_t^θ against it along the marginal path p_t (Lipman et al., 2023):

$$\mathcal{L}_{\mathrm{FM}}(\theta; w) \triangleq \mathbb{E}_{t\sim \mathrm{Unif}[0,1],\, X_t \sim p_t(\cdot|w)} \left[\|u_t^\theta(X_t; w) - u_t(X_t; w)\|_2^2\right]. \tag{2}$$

Directly optimizing (2) is impractical because the marginals p_t and the velocity u_t are unknown. Conditional flow matching (CFM) avoids this difficulty by replacing the intractable marginal supervision with a conditional transport path, while preserving the same minimizers of (2) under standard regularity conditions (Lipman et al., 2023). For a fixed w, draw $X_0 \sim p_0(\cdot)$ and $X_1 \sim p(\cdot \mid w)$, and define the *interpolation variable* $Z_t \triangleq (1-t)X_0 + tX_1$. Its *conditional velocity* is $u_{t|1}(Z_t; w \mid X_1) \triangleq X_1 - X_0$ for $t \in [0, 1]$. The resulting CFM loss is

$$\mathcal{L}_{\mathrm{CFM}}(\theta; w) \triangleq \mathbb{E}_{t\sim \mathrm{Unif}[0,1],\, X_0\sim p_0,\, X_1 \sim p(\cdot|w)} \left[\|u_t^\theta(Z_t; w) - u_{t|1}(Z_t; w \mid X_1)\|_2^2\right]. \tag{3}$$

Intuitively, (3) regresses the model vector field onto the conditional velocity along the chosen path, and minimizing this loss learns a vector field whose induced flow pushes p_0 toward the target p_1.

Solving PVIs via flow matching. We instantiate the preceding formulation with base distribution $p_0(\cdot) = \mathcal{N}_n(\cdot; 0, I)$ and choose $u_t^\theta(x; w)$ from a parameterized hypothesis class $\mathcal{F}_\Theta$ (e.g., neural networks), where $\theta \in \Theta \subseteq \mathbb{R}^q$ is the model parameter, and Θ is the model parameter space. To learn a single model that works for all VI instances, we minimize the population objective

$$\mathcal{L}(\theta) \triangleq \mathbb{E}_{w\sim p(w)} \big[\mathcal{L}_{\mathrm{CFM}}(\theta; w)\big], \tag{4}$$

where $p(w)$ is the distribution of w. Given training data $\mathcal{D} = \{(t^{(i)}, w^{(i)}, X_0^{(i)}, X_1^{(i)})\}_{i=1}^N \sim p(t, w, X_0, X_1)$.[1] We optimize the empirical loss

$$\widehat{\mathcal{L}}_{\mathcal{D}}(\theta) \triangleq \frac{1}{N} \sum_{i=1}^{N} \left\| u_{t^{(i)}}^\theta(Z_t^{(i)}; w^{(i)}) - u_{t^{(i)}|1}(Z_t^{(i)}; w^{(i)} \mid X_1^{(i)}) \right\|_2^2,$$

[1] In practice the supervision effectively consists of the pairs $(w^{(i)}, X_1^{(i)})$; the time variable $t^{(i)}$ and base sample $X_0^{(i)}$ can be drawn independently during training from $\mathrm{Unif}[0, 1]$ and p_0, respectively.

where $Z_t^{(i)} = (1 - t^{(i)})X_0^{(i)} + t^{(i)}X_1^{(i)}$. The targets $X_1^{(i)}$ are solution samples for the corresponding PVI instances; how they are generated depends on the application and is described in the experimental sections (Section 5). Given the training data $\mathcal{D}$, we minimize the empirical loss $\widehat{\mathcal{L}}_{\mathcal{D}}(\theta)$ with off-the-shelf optimizers, and denote the output by $\theta_{\mathcal{D}}$.

At inference time, for a new w, we sample $\hat{X}_0 \sim p_0$ and integrate (1) to obtain one or more candidates from $\hat{p}^{\theta_{\mathcal{D}}}(\cdot \mid w)$ for optional downstream selection.

4 Theoretical Analysis

This section translates the learning error, defined in (4), into guarantees on the learned conditional distribution, feasibility, and VI stationarity. We first control the distributional discrepancy between the learned and target distributions, then use PVI regularity to convert it into bounds on feasibility and gap-based stationarity. *Proofs of all theorems are deferred to Appendix A.*

We retain the notation from Section 3. For each w, let $\Phi_t(\cdot\,; w)$ denote the target flow map and $\hat{\Phi}_t^{\theta_{\mathcal{D}}}(\cdot\,; w)$ the learned flow map. Then, the learned and target distributions satisfy $\hat{p}^{\theta_{\mathcal{D}}}(\cdot \mid w) = \hat{\Phi}_1^{\theta_{\mathcal{D}}}(\cdot\,; w)\#p_0$ and $p(\cdot \mid w) = \Phi_1(\cdot\,; w)\#p_0$, respectively.

We distinguish structural assumptions on the PVI and data-generating process from properties of the chosen model class and the optimization error incurred during training.

- **A1 Operator regularity.** The operator $F_w(\cdot)$ is single-valued, L_F-uniformly Lipschitz in x, and ρ_F-uniformly Lipschitz in w; in particular, $F_w(x)$ is jointly continuous in (w, x).
- **A2 Feasible-set regularity.** The feasible set $\mathcal{X}_w$ is nonempty, convex, compact, and uniformly bounded: there exists $\bar{R} > 0$ such that $\mathcal{X}_w \subseteq B_{\bar{R}}(0) \triangleq \{x \in \mathbb{R}^n : \|x\|_2 \leq \bar{R}\}$. Moreover, the map $w \mapsto \mathcal{X}_w$ is H-Hausdorff Lipschitz: $d_H(\mathcal{X}_{w_1}, \mathcal{X}_{w_2}) \leq H\|w_1 - w_2\|, \forall w_1, w_2 \in \mathcal{W}$.
- **A3 Solution-distribution regularity.** The map $w \mapsto p(\cdot \mid w)$ is H'-Lipschitz in the Wasserstein-1 metric: $\mathbb{W}_1(p(\cdot \mid w_1), p(\cdot \mid w_2)) \leq H'\|w_1 - w_2\|, \forall w_1, w_2 \in \mathcal{W}$. Equivalently, for every $w_1, w_2 \in \mathcal{W}$, there exists a coupling (X_1, X_2) of $p(\cdot \mid w_1)$ and $p(\cdot \mid w_2)$ such that $\mathbb{E}\|X_1 - X_2\| \leq H'\|w_1 - w_2\|$.
- **A4 Parameter-domain regularity.** The parameter set $\mathcal{W}$ is compact, the parameter distribution admits a density bounded away from zero on $\mathcal{W}$, i.e., $m_W \triangleq \inf_{w \in \mathcal{W}} p(w) > 0$, and $\mathcal{W}$ is (r_0, κ_0)-fat: $\exists\, r_0, \kappa_0 > 0$ such that $\mathrm{Vol}(B_r(w) \cap \mathcal{W}) \geq \kappa_0 \,\mathrm{Vol}(B_r(0))\,, \forall w \in \mathcal{W},\ r \in (0, r_0]$.

A1–**A**2 reflect regularity conditions common in sensitivity and stability analyses (Dafermos, 1988; Dontchev & Rockafellar, 2009; Chen et al., 2026). **A**3 is an additional regularity condition on the chosen solution distribution. It requires nearby parameters to admit labels coupled with expected displacement proportional to $\|w_1 - w_2\|$, ruling out labeling mechanisms that jump discontinuously between separated solution branches. Related smoothness assumptions for distribution-valued regression targets and Fréchet regression functions appear in Dombry et al. (2025) and Petersen & Müller (2019). **A**4 combines a density lower bound with a (r_0, κ_0)-fatness condition on $\mathcal{W}$, analogous to the strong-density/regular-support conditions used for local averaging in nonparametric statistics (Audibert & Tsybakov, 2007; Tsybakov, 2009); together they lower-bound the probability mass in any small ball around any $w \in \mathcal{W}$. Full-dimensional domains with density bounded away from zero satisfy **A**4, whereas thin, cuspidal, or lower-dimensional regions generally do not.

Model class. For the finite-sample analysis, we choose a bounded and Lipschitz model class $\mathcal{F}_\Theta$ (cf. (7) in Appendix), such that the velocity field is uniformly Lipschitz in time, state, and VI instance parameter with Lipschitz constants γ, κ_t, and ρ_t, respectively, and the functions $t \mapsto \kappa_t$ and $t \mapsto \rho_t$ satisfy $\int_0^1 \kappa_t \mathrm{d}t$ and $\int_0^1 \rho_t \mathrm{d}t < \infty$. It can be shown that the model class has finite covering complexity (cf. Lemma B.3).

4.1 Distributional Guarantee

We first control the discrepancy between the learned and target conditional distributions in Wasserstein distance. The argument is a standard flow stability estimate that converts regression error on vector fields into terminal distributional error (Benton et al., 2024; Albergo & Vanden-Eijnden, 2023).

Theorem 4.1. *Let w be independent of $\mathcal{D}$. Then, the following hold:*

(a) ***Expected Wasserstein bound.*** $\mathbb{E}_w[\mathbb{W}_2^2\big(p(\cdot \mid w), \hat{p}^{\theta_\mathcal{D}}(\cdot \mid w)\big) \mid \mathcal{D}] \le \kappa^2\, \mathcal{L}(\theta_\mathcal{D})$*, where* κ *is the stability constant from the flow ODE; in particular, if for each* t *the learned velocity field is* κ_t*-Lipschitz in* x *uniformly over* w*, then one may take* $\kappa = \exp(\int_0^1 \kappa_t \, \mathrm{d}t)$*.*

(b) ***Data-conditional high-probability bound.*** *For any* $\delta \in (0,1)$*, conditioned on* $\mathcal{D}$*, with probability at least* $1-\delta$ *over* w*,*

$$\mathbb{W}_2^2\big(p(\cdot \mid w), \hat{p}^{\theta_\mathcal{D}}(\cdot \mid w)\big) = O\big(\min\left\{\mathcal{L}(\theta_\mathcal{D})\,\delta^{-1},\, U_\mathcal{D}(\delta)\right\}\big)\,,$$

where

$$U_\mathcal{D}(\delta) \triangleq \mathcal{L}(\theta_\mathcal{D}) + \mathcal{L}(\theta_\mathcal{D})^{\frac{1}{p+1}} \log(2/\delta) + \mathcal{L}(\theta_\mathcal{D}) \left(C_W\, \mathcal{L}(\theta_\mathcal{D})^{-\frac{p}{p+1}} - 1\right)^{1/2} \log^{1/2}(2/\delta),$$

and $C_W > 0$ *is a* $\mathcal{W}$*-dependent constant.*

(c) ***Sample-based bound.*** *If* $\mathcal{E}_{N,\delta/2} \le 1$*, then with probability at least* $1-\delta$ *over* $(\mathcal{D}, w)$*,*

$$\mathbb{W}_2^2\big(p(\cdot \mid w), \hat{p}^{\theta_\mathcal{D}}(\cdot \mid w)\big) = O\left(\min\left\{\mathcal{E}_{N,\delta/2}\,\delta^{-1},\, \mathcal{E}_{N,\delta/2}^{\frac{1}{p+1}} \log(1/\delta)\right\}\right),$$

where

$$\mathcal{E}_{N,\delta} \triangleq \inf_{\theta\in\Theta} \mathcal{L}(\theta) + \eta_\mathcal{D} + \widetilde{O}(\mathrm{polylog}(1/\delta)\cdot N^{-1/2}), \tag{5}$$

is the total error. The first term $\inf_{\theta\in\Theta}\mathcal{L}(\theta)$ *is related to the model approximation error (Appendix A),* $\eta_\mathcal{D} \triangleq \widehat{\mathcal{L}}_\mathcal{D}(\theta_\mathcal{D}) - \inf_{\theta\in\Theta}\widehat{\mathcal{L}}_\mathcal{D}(\theta)$ *is the optimization error, and the last* $\widetilde{O}$ *term is the finite-sample generalization error for our chosen model class, where* $\widetilde{O}$ *suppresses logarithmic factors in* N *and fixed problem- and model-dependent constants (cf. Lemma B.5).*

Theorem 4.1-(a) shows that learning error controls the average Wasserstein distance between the learned and target distributions. Theorem 4.1-(b)–(c) extend this guarantee to a random test instance w and to a finite training sample, respectively.

4.2 Feasibility Guarantee

The samples from the learned distribution $\hat{p}^{\theta_\mathcal{D}}(\cdot \mid w)$ need not lie exactly in the feasible space $\mathcal{X}_w$, so the next step is to translate distributional error into feasibility. Without additional problem structure, a natural measure of (in)feasibility is the point-to-set distance $d(x, \mathcal{X}_w) \triangleq \inf_{y\in\mathcal{X}_w} \|x-y\|_2$.

Theorem 4.2 (Feasibility Guarantees)**.** *Let* w *be independent of* $\mathcal{D}$*. Then, the following hold:*

(a) ***Expected approximate feasibility.*** $\mathbb{E}_{w,X_0}[d^2(\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w), \mathcal{X}_w) \mid \mathcal{D}] \le \kappa^2\, \mathcal{L}(\theta_\mathcal{D})$*.*

(b) ***High-probability approximate feasibility.*** *If* $\mathcal{E}_{N,\delta/2} \le 1$ *(cf. (5)), then with probability at least* $1-\delta$ *over* $(\mathcal{D}, w)$*,*

$$\mathbb{E}_{X_0}\left[d\left(\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w), \mathcal{X}_w\right) \mid \mathcal{D}, w\right] = O\left(\min\left\{\mathcal{E}_{N,\delta/2}^{1/2}\delta^{-1/2}, \mathcal{E}_{N,\delta/2}^{\frac{1}{2(p+1)}} \log(1/\delta)\right\}\right).$$

Theorem 4.2-(a) shows that learning error controls approximate feasibility on average, with the same form and constant as the distributional guarantee (Theorem 4.1-(a)). Theorem 4.2-(b) extends this to a finite-sample high-probability guarantee over the training data and test instance. Exact feasibility of the learned model itself, for finite training data, still requires constrained generative mechanisms such as reflected or gauge flow matching (Lou & Ermon, 2023; Xie et al., 2024; Li et al., 2026).

4.3 VI-Stationarity Guarantee

We next quantify how close the samples from the learned distribution are to solving the VI itself. Without monotonicity, a natural stationarity measure is the gap function $G_w(x) \triangleq \sup_{y\in\mathcal{X}_w} F_w(x)^\top(x-y)$, which satisfies $G_w(x) \ge 0$ for $x \in \mathcal{X}_w$ and $G_w(x) = 0$ if and only if $x \in \mathcal{S}(w)$. When $x \notin \mathcal{X}_w$, $G_w(x)$ should be interpreted as a signed extension of the VI gap rather than as a nonnegative residual.

Theorem 4.3 (Stationarity Guarantees)**.** *Let* $\bar{M} \triangleq \sup_{(w,x)\in\mathcal{W}\times B_{\bar{R}}(0)} \|F_w(x)\| < \infty$*, and* w *be independent of* $\mathcal{D}$*. Then, the following hold:*

(a) ***Expected stationarity.*** $\mathbb{E}_{w,X_0}[\|G_w(\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0;w))\|\,|\mathcal{D}] \le \rho_G^{x,U}\,\kappa\,\sqrt{\mathcal{L}(\theta_\mathcal{D})}$, *where* $\rho_G^{x,U} \triangleq \left(\mathbb{E}_{X_0}\left[(L_F\|X_0\| + L_F U + 3\bar{R}L_F + \bar{M})^2\right]\right)^{1/2} < \infty$, *and* $X_0 \sim N(0,I)$,

(b) ***High-probability stationarity.*** *If* $\mathcal{E}_{N,\delta/2} \le 1$ *(cf. (5)), then with probability at least* $1-\delta$ *over* $(\mathcal{D}, w)$,

$$\mathbb{E}_{X_0}\left[\left\|G_w\left(\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0;w)\right)\right\| \mid \mathcal{D}, w\right] = O\left(\min\left\{\mathcal{E}_{N,\delta/2}^{1/2}\delta^{-1}, \mathcal{E}_{N,\delta/2}^{\frac{1}{2(p+1)}}\log(1/\delta)\right\}\right).$$

Theorem 4.3-(a) shows that learning error controls the expected VI stationarity gap. Theorem 4.3-(b) extends this to a finite-sample high-probability guarantee over the training data and test instance.

5 Experiments

We evaluate our method on three complementary benchmarks designed to probe different aspects of distributional PVI solving. A Braess traffic-equilibrium network tests whether the learned distribution recovers the diversity of a non-singleton equilibrium set. An alternating-current optimal power flow (ACOPF) benchmark examines theory-predicted feasibility and optimality scaling with learning error, as well as time–accuracy tradeoffs against iterative and learning-based baselines. Finally, non-monotone bimatrix games test equilibrium quality in a setting without a scalar potential and evaluate distributional prediction through best-of-M and paired welfare-based selection.

5.1 Traffic User Equilibrium: A Braess Network with Non-Unique Solutions

We evaluate set-valued solution recovery on the Braess network (Braess, 1968) modified with a duplicated shortcut link so that its Wardrop equilibrium set is provably non-singleton. A *Wardrop equilibrium* (Wardrop, 1952) is a flow at which no traveler can reduce their own travel cost by unilaterally switching paths. Fig. 1 shows the network: origin A, destination D, and intermediate nodes B, C, connected by six links, two of which (BC_1, BC_2) are identical, flow-independent-cost copies of the shortcut in the Braess paradox. The parameter w is the scalar origin–destination travel demand, and x is the flow vector over the four A–D paths (two direct, two via the shortcut). The equilibrium solves the corresponding VI with a congestion-based (Bureau of Public Roads, 1964) path-cost operator F_w. Solving in closed form (see Appendix C.1 for the full derivation) shows that whenever both shortcut copies carry flow, their cost tie makes the split between them a free direction, giving a one-dimensional segment of equilibria, arising from the routing symmetry between the duplicated shortcuts.

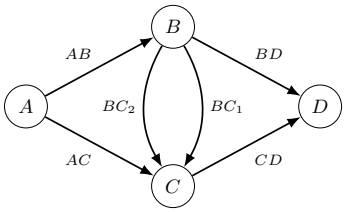

Figure 1: **Braess network with duplicated shortcut.** AB, CD: congestion-sensitive cost; AC, BD: constant cost 45; BC_1, BC_2: identical, flow-independent-cost copies of the classical shortcut.

Baselines. We compare against three learning-based optimization baselines adapted to our VI setting: *L2WS* (Sambharya et al., 2024) learns warm starts for fixed-point optimization algorithms, *DiOpt* (Ding et al., 2025) uses diffusion models for constrained optimization, and *MISO* (Sharony et al., 2024) predicts multiple initial solutions for optimization problems. For MISO, we report its winner-take-all variant, MISO (wta), and its mixture variant, MISO (mix).

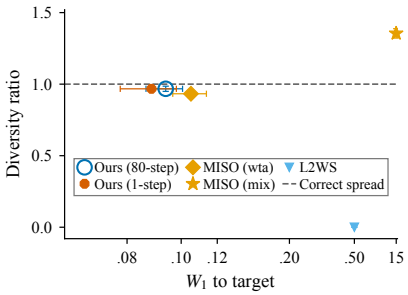

Figure 2: **Recover solution diversity.** Diversity ratio measures the shortcut-split spread (1.0 is correct); W_1 measures distance to $\mathrm{Unif}[-1,1]$. Points show five-seed means with 95% CIs.

Results. Figure 2 highlights the main set-valued behavior: *our method preserves variation along the non-unique equilibrium direction and learns the equilibrium distribution.* Our 80-step flow has the highest mean diversity ratio among the learned methods, and the one-step endpoint-distilled (Liu et al., 2023) flow has the lowest mean W_1, with both flows closer to the target law than MISO (wta). Deterministic L2WS collapses to zero diversity, whereas MISO (mix) over-disperses well beyond the target spread. Details are deferred to Table 2 in Appendix C.1.

5.2 Alternating Current Optimal Power Flow

ACOPF stress-tests the theory outside its convex setting. We use ACOPF on the IEEE 30-, 57-, and 118-bus systems from OPFLearn (Joswig-Jones et al., 2022). Metric definitions and evaluation details are deferred to Appendix C.2. ACOPF minimizes generation cost subject to nonlinear AC power-flow equations and physical constraints, so its feasible set is generally nonconvex (see (Zhao & Barati, 2024) for a recent overview), which is formally outside **A**2's convexity requirement (Section 4). Because OPFLearn provides only one optimizer-selected label per load profile, *this benchmark is not used to evaluate solution diversity*. Instead, it asks whether the predicted relationship between feasibility error and square-root learning error, $\sqrt{\mathcal{L}}$ in (4), remains visible in a nonconvex setting.

Figure 3 gives the ACOPF scaling check: on the largest 118-bus network, *normalized constraint violation scales nearly linearly with $\sqrt{\mathcal{L}}$ in log–log coordinates, close to the slope* 1 *predicted by Theorem 4.2*. The full diagnostic in Appendix D is more mixed: IEEE 30- and 57-bus violation saturates near the OPFLearn label-imperfection floor early in training, and relative suboptimality decreases faster than the worst-case $O(\sqrt{\mathcal{L}})$ rate. Therefore, ACOPF should be viewed as a stress test with one theory-facing signal, rather than as uniform evidence that every metric and case follows the predicted exponent.

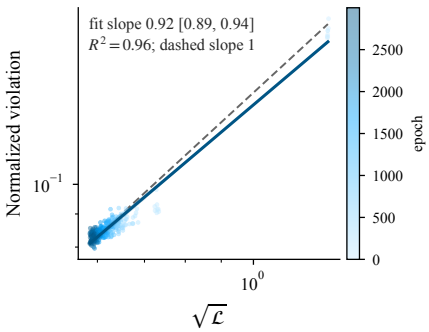

Figure 3: **IEEE 118-bus scaling.** Each point is one seed/checkpoint pair, colored by epoch.

Comparison against ACOPF baselines. We compare our learned sampler's single-draw accuracy and speed against two published learning-based ACOPF methods: *DC3* (Donti et al., 2021), which is self-supervised (trains directly against the true objective and constraints via a differentiable power-flow completion step plus unrolled gradient correction, using no solver labels at all), and *Fioretto et al.*'s Lagrangian-dual method (Fioretto et al., 2020), which penalizes constraint violation with dual-ascent-updated multipliers during supervised training. Figure 4 reports mean values over five seeds, timed identically to our own single-draw time. The figure also includes one-step distilled students trained to match the corresponding 70-step flow teachers. Full baseline numbers (with standard deviations) are given in Table 3, Appendix C.2.

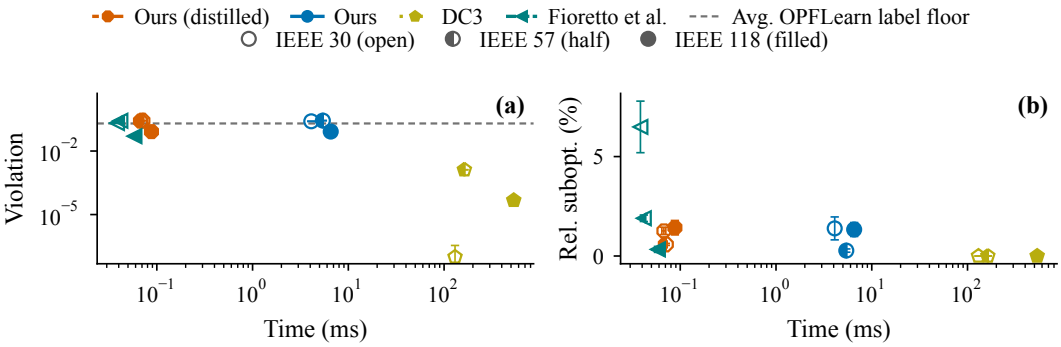

Figure 4: **ACOPF time–accuracy frontier.** Violation (left) and relative suboptimality (right) vs. single-instance wall-clock time across three IEEE cases (open/half/filled markers). Each case also includes a one-step distilled sampler. Error bars are 95% CIs over five seeds. The dashed gray line marks the average OPFLearn label-level violation floor across the three cases.

Results. Figure 4 shows a tradeoff: DC3 gives the lowest violation in every case because it trains against the true constraints and applies an unrolled correction step, but that correction makes it much slower than both our method and Fioretto et al. (2020). Across all three IEEE cases, the distilled samplers collapse the multi-step flow into one network evaluation while retaining violation close to the original teachers and the case-specific OPFLearn label floors. Relative suboptimality is mixed: our method is lower than Fioretto et al. (2020) on IEEE 30 and 57, while Fioretto et al. (2020) is lower

on IEEE 118. Against a classical ACOPF solver run from scratch (`pp.runopp`, Appendix C.2), our sampler is 34–43× faster per query (with distillation, we can gain roughly 60–70× speedup). Overall, our sampler provides fast, label-level feasible amortized predictions in a nonconvex benchmark, while specialized ACOPF baselines can be better when the target is only a single deterministic solution.

5.3 Parametric Non-monotone General-sum Bimatrix Game

We finally consider parametric two-player non-monotone general-sum bimatrix games with payoff matrices $A(w) = A_0 + \sum_{i=1}^{k} w_i A_i$ and $B(w) = B_0 + \sum_{i=1}^{k} w_i B_i$, where $\{A_i, B_i\}_{i=0}^{k}$ are fixed base matrices, and $w \triangleq [w_1, \dots, w_k]^\top$ is random. Each player chooses a mixed strategy in the probability simplex $\Delta_n \triangleq \{z \in \mathbb{R}^n_+ : \mathbf{1}^\top z = 1\}$, and a Nash equilibrium $x^\star = (p^\star, q^\star)$ solves the VI on $\mathcal{X} = \Delta_n \times \Delta_n$ with operator $F_w(p, q) = [-A(w)q; -B(w)^\top p]$. This problem is generally non-monotone and may have multiple equilibria. We evaluate three game families: n=5, k=7 (small), n=8, k=12 (medium), and n=50, k=16 (large). We report Nash gap, best-of-M selection, and equal-time comparisons; full metric definitions and protocol details are deferred to Appendix C.3. We compare against *Optimistic Gradient Descent–Ascent (OGDA)* (Daskalakis & Panageas, 2018), *Extragradient (EG)* (Korpelevich, 1976), and *Optimistic Mirror Descent (OMD)* (Mertikopoulos et al., 2019). We also reuse the same learning-based optimization baselines introduced in Section 5.1 (MISO, L2WS, DiOpt), here re-adapted to the game's settings (Appendix D.4 gives details for each).

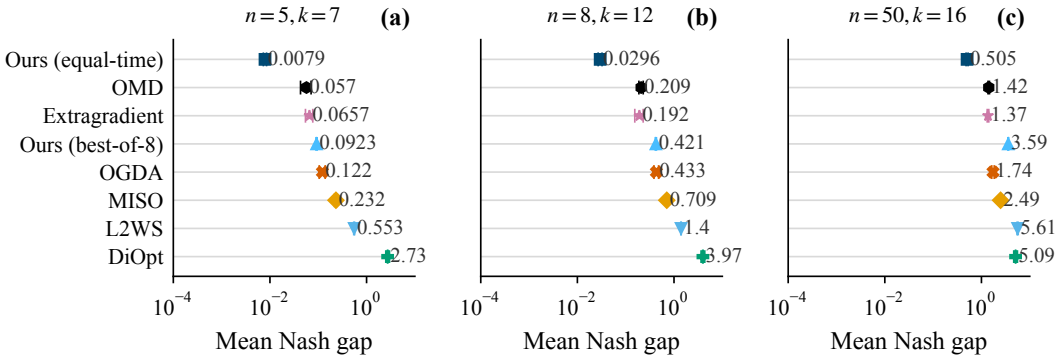

Figure 5: **Nash-gap comparison across game families.** Our method (best-of-8 and equal-time-to-OMD), three learning-based baselines (MISO, L2WS, DiOpt), and three tuned classical VI solvers (OGDA, Extragradient, OMD) are compared one panel per game family, with methods ordered by their (n=5, k=7) mean gap and held in that same order throughout. Error bars are 95% CIs over five seeds. Full Nash-gap statistics are in Table 9 (Appendix D.2).

Results. Figure 5 shows that sampling is essential for our method: *a single generated strategy is often inaccurate, while best-of-8 selection moves the flow model close to the strongest methods in each family.* On the small (n=5, k=7) and medium (n=8, k=12) games, our best-of-8 sampler has the lowest mean Nash gap among the learning-based methods. On the large (n=50, k=16) game, our method improves over L2WS and DiOpt, while MISO attains a lower mean gap. Classical solvers remain strong. However, with the same computational time budget as OMD, our method achieves the lowest mean Nash gap among all methods.

Social-welfare selection. We filter candidates by Nash gap ($\leq 10^{-2}$) and select the highest-welfare candidate. Figure 6 separates two complementary aspects of performance: the filled markers compare each method with OMD on the subset of instances where both produce an admissible candidate (markers are shown only when this subset contains at least 50 instances), while the dashed curves report each method's admissible rate. Our method achieves a positive mean paired welfare difference across all three game families, showing that sampling can expose higher-welfare equilibria on some instances. Because this comparison is conditional on joint admissibility, the positive mean does not imply a uniform per-instance or statistically significant advantage; paired intervals, subset sizes, and oracle-regret comparisons are reported in Appendix Table 11. L2WS/DiOpt(/MISO) also achieve positive paired welfare differences for the small (and median) families, but with substantially lower admissibility rates than our method. Further details are reported in Appendix D.2.

6 Conclusion

We presented a flow-matching framework for solving parametric variational inequalities by learning a conditional distribution over solutions rather than committing to a single predictor. This formulation is particularly useful when solution sets are non-singleton and related VI instances must be solved repeatedly: after training, new instances can be handled by sampling, with multiple samples available for downstream selection. Our analysis connects flow-matching error to Wasserstein distributional accuracy, approximate feasibility, and VI gap-based stationarity, together with finite-sample guarantees that account for model approximation, optimization, and generalization errors. Across three benchmarks, the learned sampler preserves solution diversity, supports amortized inference, and enables post-hoc selection based on criteria such as solution quality or social welfare. These results suggest that solution-distribution learning is a promising paradigm for parametric equilibrium and optimization problems where diversity, repeated inference, or downstream selection matters.

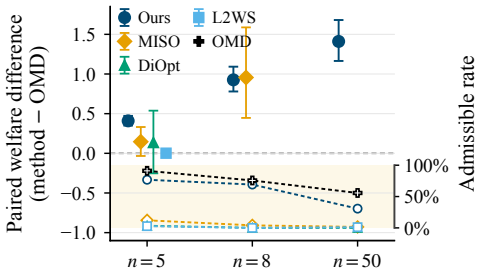

Figure 6: **Welfare selection.** Mean within-instance welfare difference (method minus OMD; filled markers and left y-axis) after filtering candidates. Each welfare comparison is evaluated on its paired subset with OMD. Markers are shown only when at least 50 paired instances are available. Error bars are hierarchical-bootstrap 95% CIs, resampling seeds and instances. Positive values favor the method and negative values favor OMD. Dashed curves with open markers use the right y-axis and show the admissible rates.

Future work. Several directions extend naturally from this framework. Stronger feasibility and stationarity guarantees could be pursued through projection, correction, or constraint-aware flow variants, including reflected or gauge flow matching (Lou & Ermon, 2023; Xie et al., 2024; Li et al., 2026). On the data side, extending the analysis to imperfect solution labels, such as approximately solved ones, would clarify how label quality propagates through the learned solution distribution to PVI guarantees. On the theoretical side, improving the parameter-dimension dependence of the high-probability bounds, potentially by exploiting low intrinsic dimensionality, or additional regularity of the PVI family, is an important direction. Finally, an end-to-end study of amortization, which includes offline label-generation and training costs and break-even behavior over repeated queries, could further characterize when distributional PVI learning provides the computational benefit and inform more sample- and training-efficient solvers.

Use of Generative AI

In preparing this work, we used generative AI assistants, including OpenAI Codex and Anthropic Claude Code, across several stages of the research and writing process. They helped refine the theoretical framing, formulate and edit mathematical statements, explore proof strategies, and audit draft proofs for gaps or inconsistent assumptions. The authors subsequently reviewed and verified every mathematical claim and proof. They also assisted with experimental design; writing, modifying, and debugging research code; running and monitoring experiments; analyzing the Braess, ACOPF, and bimatrix-game results; and generating or revising figures and tables. The authors inspected the implementations, protocols, raw outputs, and reported numerical results. Generative AI further supported literature searches, paper summarization, technical comparisons with related work, citation and bibliography audits, manuscript drafting and editing, appendix organization, LaTeX formatting and template migration, compilation checks, and preparation of the reproducibility codebase. Citations and factual claims were checked against the cited sources by the authors. Generative AI was not used to manufacture or replace experimental observations, and it was not used to generate the training or evaluation datasets; those instances and labels were produced by the solver-based pipelines described in the paper. The authors reviewed all AI-assisted contributions and take responsibility for the final manuscript, proofs, code, data-processing decisions, and experimental claims.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023.

Brandon Amos. Tutorial on amortized optimization. *Foundations and Trends® in Machine Learning*, 16(5):592–732, 2023. doi:10.1561/2200000082.

Jean-Yves Audibert and Alexandre B Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of Statistics*, 35(2):608–633, 2007.

Heinz H Bauschke and Patrick L Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer, 2 edition, 2017. doi:10.1007/978-3-319-48311-5.

Amir Beck. *First-Order Methods in Optimization*. SIAM, 2017.

Joe Benton, George Deligiannidis, and Arnaud Doucet. Error bounds for flow matching methods. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=uqQPyWFDhY.

Dimitri P Bertsekas. *Nonlinear Programming*. Athena Scientific, 2 edition, 1999.

Mathieu Besancon, Miguel F. Anjos, and Luce Brotcorne. Robust bilevel optimization for near-optimal lower-level solutions. *Journal of Global Optimization*, 90:813–842, 2024.

Vladimir I Bogachev. *Measure Theory*. Springer, 2007.

Stephen P Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Dietrich Braess. Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12(1): 258–268, 1968.

Bureau of Public Roads. Traffic assignment manual. Technical report, U.S. Department of Commerce, Urban Planning Division, Washington, D.C., 1964.

Xiaojun Chen, Jin Zhang, and Yixuan Zhang. Optimization with parametric variational inequality constraints on a moving set. *arXiv preprint arXiv:2603.05196*, 2026.

Stella Dafermos. Traffic equilibrium and variational inequalities. *Transportation Science*, 14(1): 42–54, 1980.

Stella Dafermos. Sensitivity analysis in variational inequalities. *Mathematics of Operations Research*, 13(3):421–434, 1988.

Constantinos Daskalakis and Ioannis Panageas. The limit points of (optimistic) gradient descent in min-max optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Shutong Ding, Yimiao Zhou, Ke Hu, Xi Yao, Junchi Yan, Xiaoying Tang, and Ye Shi. Exploring the boundary of diffusion-based methods for solving constrained optimization. *arXiv preprint arXiv:2502.10330*, 2025.

Clément Dombry, Thibault Modeste, and Romain Pic. Stone's theorem for distributional regression in Wasserstein distance. *Journal of Nonparametric Statistics*, 37(2):430–452, 2025. doi:10.1080/10485252.2024.2393172.

Asen L Dontchev and R Tyrrell Rockafellar. *Implicit Functions and Solution Mappings*. Springer, 2009.

Priya L Donti, David Rolnick, and J Zico Kolter. DC3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations*, 2021.

Rick Durrett. *Probability: Theory and Examples*, volume 49. Cambridge University Press, 2019.

Francisco Facchinei and Jong-Shi Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer, 2003.

Ferdinando Fioretto, Terrence W K Mak, and Pascal Van Hentenryck. Predicting AC optimal power flows: Combining deep learning and Lagrangian dual methods. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):630–637, 2020. doi:10.1609/aaai.v34i01.5403.

Thomas Hakon Gronwall. Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Annals of Mathematics*, 20(4):292–296, 1919.

John C. Harsanyi and Reinhard Selten. *A General Theory of Equilibrium Selection in Games*. MIT Press, 1988.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, pp. 9902–9915. PMLR, 2022.

Trager Joswig-Jones, Kyri Baker, and Ahmed S. Zamzam. OPF-Learn: An open-source framework for creating representative AC optimal power flow datasets. In *2022 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5. IEEE, 2022. doi:10.1109/ISGT50606.2022.9817509.

David Kinderlehrer and Guido Stampacchia. *An Introduction to Variational Inequalities and Their Applications*. SIAM, 2000.

Galina M Korpelevich. The extragradient method for finding saddle points and other problems. *Matekon*, 12:747–756, 1976.

Jerzy Kyparisis. Parametric variational inequalities with multivalued solution sets. *Mathematics of Operations Research*, 17(2):341–364, 1992.

Béatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000.

Xinpeng Li, Enming Liang, and Minghua Chen. Gauge flow matching: Efficient constrained generative modeling over general convex set and beyond. In *International Conference on Learning Representations*, 2026.

Zihao Li, Hui Yuan, Kaixuan Huang, Chengzhuo Ni, Yinyu Ye, Minshuo Chen, and Mengdi Wang. Diffusion model for data-driven black-box optimization. *arXiv preprint arXiv:2403.13219*, 2024.

Enming Liang and Minghua Chen. Generative learning for solving non-convex problem with multi-valued input-solution mapping. In *The Twelfth International Conference on Learning Representations*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Aaron Lou and Stefano Ermon. Reflected diffusion models. In *International Conference on Machine Learning*, pp. 22675–22701. PMLR, 2023.

Panayotis Mertikopoulos, Christos Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2703–2717. SIAM, 2018.

Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *International Conference on Learning Representations*, 2019.

Anna Nagurney. *Network Economics: A Variational Inequality Approach*, volume 10. Springer Science & Business Media, 2013.

Alexander Petersen and Hans-Georg Müller. Fréchet regression for random objects with Euclidean predictors. *The Annals of Statistics*, 47(2):691–719, 2019.

R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

Rajiv Sambharya, Georgina Hall, Brandon Amos, and Bartolomeo Stellato. Learning to warm-start fixed-point optimization algorithms. *Journal of Machine Learning Research*, 25(166):1–46, 2024.

Elad Sharony, Heng Yang, Tong Che, Marco Pavone, Shie Mannor, and Peter Karkus. Learning multiple initial solutions to optimization problems. *arXiv preprint arXiv:2411.02158*, 2024.

Leon Thurner, Alexander Scheidler, Florian Schäfer, Jan-Hendrik Menke, Julian Dollichon, Friederike Meier, Steffen Meinecke, and Martin Braun. pandapower—an open-source python tool for convenient modeling, analysis, and optimization of electric power systems. *IEEE Transactions on Power Systems*, 33(6):6510–6521, 2018. doi:10.1109/TPWRS.2018.2829021.

Alexandre B Tsybakov. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer, 2009.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019.

John Glen Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1(3):325–362, 1952.

Wolfram Wiesemann, Angelos Tsoukalas, Polyxeni Kleniati, and B. Rustem. Pessimistic bilevel optimization. *SIAM Journal on Optimization*, 23(1):353–380, 2013.

Tianyu Xie, Yu Zhu, Longlin Yu, Tong Yang, Ziheng Cheng, Shiyue Zhang, Xiangyu Zhang, and Cheng Zhang. Reflected flow matching. In *International Conference on Machine Learning*, pp. 54614–54634. PMLR, 2024.

Meng Zhao and Masoud Barati. Synergizing machine learning with ACOPF: A comprehensive overview. *arXiv preprint arXiv:2406.10428*, 2024.

Ray Daniel Zimmerman, Carlos E. Murillo-Sánchez, and Robert John Thomas. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, 2011. doi:10.1109/TPWRS.2010.2051168.

Appendix

A Proofs of Main Theorems **13**

A.1 Proof of Theorem 4.1 . . . 13

A.2 Proof of Theorem 4.2 . . . 15

A.3 Proof of Theorem 4.3 . . . 15

B Supplementary Technical Lemmas **16**

B.1 Relation Between FM Loss and CFM Loss . . . 17

B.2 Bayes CFM Risk . . . 17

B.3 Covering Number Bound . . . 17

B.4 Generalization Bound . . . 18

B.5 Continuity of the Gap Function and the Solution Distribution in the Parameter . . . 23

B.6 Lipschitz Continuity of the CFM Loss in the Parameter . . . 23

B.7 Parameter-Pointwise Expected Gap Bound . . . 24

B.8 Uniform-in-Parameter Control of the CFM Loss . . . 25

B.9 Mean-square Flow Coupling . . . 26

C Implementation Details **27**

C.1 Braess Network . . . 27

C.2 ACOPF . . . 29

C.3 Non-Potential Bimatrix Game Training Data Generation . . . 31

D Additional Numerical Experiment Results **35**

D.1 ACOPF Scaling Diagnostics . . . 35

D.2 Full Bimatrix-Game Results Table and Consistency-vs-Exactness Summary . . . 36

D.3 Short-Restart Multistart Comparison for OGDA, Extragradient, and OMD . . . 36

D.4 Learning-Based Baseline Methodology: MISO, L2WS, and DiOpt . . . 37

D.5 Large-Game Label-Generation Protocol . . . 40

D.6 ODE Sampler-Step Ablation Protocol . . . 40

E Additional Related Work and Discussion **40**

A Proofs of Main Theorems

In this section, we prove the main theorems from Section 4. All technical lemmas used in the proofs are provided in Appendix B.

A.1 Proof of Theorem 4.1

Proof. (a). Fix w and suppress it from the notation where convenient. Let $u_t(\cdot)$ denote the true velocity field and $\hat{u}_t(\cdot) = u_t^{\theta_{\mathcal{D}}}(\cdot)$ the learned velocity field. Let $(X_t)_{t\in[0,1]}$ and $(\hat{X}_t)_{t\in[0,1]}$ be the unique solutions of the ODEs

$$\dot{X}_t = u_t(X_t), \qquad \dot{\hat{X}}_t = \hat{u}_t(\hat{X}_t), \qquad t \in [0,1],$$

with shared initialization $X_0 = \hat{X}_0 \sim p_0(\cdot)$. Denote by $p_t(\cdot \mid w)$ and $\hat{p}_t(\cdot \mid w)$ the laws of X_t and $\hat{X}_t$. By construction, $p_1(\cdot \mid w) = p(\cdot \mid w)$.

Since $(X_1, \hat{X}_1)$ is a coupling of $(p_1, \hat{p}_1)$, we have

$$\mathbb{W}_2^2(\hat{p}_1, p_1) \le \mathbb{E}[\|\hat{X}_1 - X_1\|^2 \mid \mathcal{D}].$$

Note that the map $x \mapsto \hat{u}_t(x)$ is κ_t-Lipschitz for any $t \in [0,1]$ according to our choice of the model class (see (7)). Let $\Delta_t \triangleq \hat{X}_t - X_t$. Using this Lipschitz property and Grönwall's inequality (Gronwall, 1919) yields

$$\|\Delta_1\| \le \kappa \int_0^1 \|\hat{u}_s(X_s) - u_s(X_s)\| \mathrm{d}s, \qquad \kappa \triangleq \exp\left(\int_0^1 \kappa_r \, dr\right) < \infty.$$

Therefore,

$$\left\|\widehat{X}_1 - X_1\right\|^2 = \|\Delta_1\|^2 \le \kappa^2 \int_0^1 \|\hat{u}_s(X_s) - u_s(X_s)\|^2 \mathrm{d}s.$$

Taking conditional expectation given $\mathcal{D}$ and using Fubini-Tonelli,

$$\mathbb{E}[\|\hat{X}_1 - X_1\|^2 \mid \mathcal{D}] \le \kappa^2 \int_0^1 \mathbb{E}_{p_s(\cdot|w)}[\|u_s^{\theta_\mathcal{D}} - u_s\|^2 \mid \mathcal{D}] \mathrm{d}s \triangleq \kappa^2 \mathcal{L}_{\mathrm{FM}}(\theta_\mathcal{D}; w), \tag{6}$$

Combining with Lemma B.1,

$$\mathbb{W}_2^2(\hat{p}_1(\cdot \mid w), p(\cdot \mid w)) \le \kappa^2 \mathcal{L}_{\mathrm{FM}}(\theta_\mathcal{D}; w) \le \kappa^2 \mathcal{L}_{\mathrm{CFM}}(\theta_\mathcal{D}; w).$$

Finally, taking expectation with respect to w of the inequality above yields

$$\mathbb{E}_w\left[\mathbb{W}_2^2(\hat{p}_1(\cdot \mid w), p(\cdot \mid w)) \mid \mathcal{D}\right] \le \kappa^2 \mathbb{E}_w\left[\mathcal{L}_{\mathrm{FM}}(\theta_\mathcal{D}; w) \mid \mathcal{D}\right] \le \kappa^2 \mathcal{L}(\theta_\mathcal{D}).$$

This proves (a) in its stated (weaker) form. The tighter form in terms of $\mathbb{E}_w\left[\mathcal{L}_{\mathrm{FM}}(\theta_\mathcal{D}; w) \mid \mathcal{D}\right]$ is not used for (a).

(b). The first argument in the minimum follows directly from Markov's inequality applied to (a)'s bound: $\mathbb{P}(\mathbb{W}_2^2 > t \mid \mathcal{D}) \le \kappa^2 \mathbb{E}_w[\mathcal{L}_{\mathrm{FM}}(\theta_\mathcal{D}; w) \mid \mathcal{D}]/t$.

For the second branch, conditioned on $\mathcal{D}$, Lemma B.10 shows that $Y(w) \triangleq \mathcal{L}_{\mathrm{CFM}}(\theta_\mathcal{D}; w) \le C_W \mathcal{L}(\theta_\mathcal{D})^{\frac{1}{p+1}}$. We have $\mathbb{E}_w[Y \mid \mathcal{D}] = \mathcal{L}(\theta_\mathcal{D})$. Let $\mu \triangleq \mathcal{L}(\theta_\mathcal{D})$. Then,

$$\begin{aligned}\mathrm{Var}_w(Y \mid \mathcal{D}) &= \mathbb{E}_w[Y^2 \mid \mathcal{D}] - \mu^2 \le C_W \mathcal{L}(\theta_\mathcal{D})^{\frac{1}{p+1}} \mathbb{E}_w[Y \mid \mathcal{D}] - \mu^2 \\ &= \mu(C_W \mathcal{L}(\theta_\mathcal{D})^{\frac{1}{p+1}} - \mu) = \mu^2 \left(C_W \mathcal{L}(\theta_\mathcal{D})^{-\frac{p}{p+1}} - 1\right).\end{aligned}$$

This last quantity is always nonnegative, since $Y(w) \le C_W \mathcal{L}(\theta_\mathcal{D})^{1/(p+1)}$ holds for all w (Lemma B.10), averaging over w gives $\mathcal{L}(\theta_\mathcal{D}) = \mathbb{E}_w[Y \mid \mathcal{D}] \le C_W \mathcal{L}(\theta_\mathcal{D})^{1/(p+1)}$, i.e., $C_W \mathcal{L}(\theta_\mathcal{D})^{-p/(p+1)} \ge 1$, for every $\mathcal{L}(\theta_\mathcal{D}) > 0$.

By one-sided Bernstein inequality, with probability at least $1 - \delta$ over w (Wainwright, 2019, Chapter 2),

$$Y < \mu + \sqrt{2\mu(C_W \mathcal{L}(\theta_\mathcal{D})^{\frac{1}{p+1}} - \mu) \log \frac{1}{\delta}} + \frac{C_W \mathcal{L}(\theta_\mathcal{D})^{\frac{1}{p+1}}}{3} \log \frac{1}{\delta}.$$

Finally, by the per-instance bound established in (a),

$$\mathbb{W}_2^2\left(p(\cdot \mid w), \hat{p}^{\theta_\mathcal{D}}(\cdot \mid w)\right) \le \kappa^2 \mathcal{L}_{\mathrm{CFM}}(\theta_\mathcal{D}; w) = \kappa^2 Y(w).$$

Combining this branch with the Markov branch above (each gives an independently valid upper bound on $\mathbb{W}_2^2$) yields (b).

(c). Apply (b) at level $\delta/2$ (over w, conditioned on $\mathcal{D}$), and separately condition on the event from Lemma B.5-(a) at level $\delta/2$ (over $\mathcal{D}$), under which $\mathcal{L}(\theta_\mathcal{D}) \le \mathcal{E}_{N,\delta/2}$. By a union bound, the intersection of these two events has probability at least $1 - \delta$ over $(\mathcal{D}, w)$ jointly. On this intersection, (b) holds and $\mathcal{L}(\theta_\mathcal{D}) \le \mathcal{E}_{N,\delta/2}$, so substituting $\mathcal{E}_{N,\delta/2}$ into both branches of (b) and simplifying under $\mathcal{E}_{N,\delta/2} \le 1$, $\log(2/\delta) \ge 1$ (the second branch's Bernstein expression is bounded exactly as before, its first term being $\mathcal{E}_{N,\delta/2}^{1/(p+1)} \log(2/\delta)$ and the middle term no larger since $(p+2)/(2(p+1)) \ge 1/(p+1)$) gives (c). A sharper Markov branch is given in Remark A.1. □

A.2 Proof of Theorem 4.2

Proof. (a). Since $\Phi_1(X_0; w)$ with $X_0 \sim p_0$ has law $p(\cdot \mid w)$ for each fixed w (Section 3), and $p(\cdot \mid w)$ is supported on $\mathcal{S}(w) \subseteq \mathcal{X}_w$ (Section 3), $\Phi_1(X_0; w) \in \mathcal{X}_w$ almost surely. Hence, taking $y = \Phi_1(X_0; w) \in \mathcal{X}_w$ in the definition of the point-to-set distance $d(z, \mathcal{X}_w) = \inf_{y' \in \mathcal{X}_w} \|z - y'\| \leq \|z - y\|$, and setting $z = \hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w)$, gives, for every realization of X_0,

$$d\left(\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w), \mathcal{X}_w\right) \leq \left\|\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w) - \Phi_1(X_0; w)\right\|$$

almost surely. Squaring and taking expectation over (w, X_0) conditioned on $\mathcal{D}$, Lemma B.11 gives

$$\begin{aligned}\mathbb{E}_{w,X_0}\left[d^2\left(\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w), \mathcal{X}_w\right) \mid \mathcal{D}\right] &\leq \mathbb{E}_{w,X_0}\left[\left\|\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w) - \Phi_1(X_0; w)\right\|^2 \mid \mathcal{D}\right] \\ &\leq \kappa^2\, \mathbb{E}_w[\mathcal{L}_{\mathrm{FM}}(\theta_\mathcal{D}; w)] \leq \kappa^2 \mathcal{L}(\theta_\mathcal{D}),\end{aligned}$$

which proves (a).

(b). Let $Z(w) \triangleq \mathbb{E}_{X_0}[d(\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w), \mathcal{X}_w) \mid \mathcal{D}, w]$ and $m_\mathcal{D} \triangleq \mathbb{E}_w[Z(w) \mid \mathcal{D}]$. The first argument in the minimum follows from Markov's inequality applied to $Z(w)^2$: by Jensen's inequality $Z(w)^2 \leq \mathbb{E}_{X_0}[d^2(\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w), \mathcal{X}_w) \mid \mathcal{D}, w]$, so (a) gives $\mathbb{E}_w[Z(w)^2 \mid \mathcal{D}] \leq \kappa^2 \mathbb{E}_w[\mathcal{L}_{\mathrm{FM}}(\theta_\mathcal{D}; w)] \leq \kappa^2 \mathcal{L}(\theta_\mathcal{D})$ (Lemma B.1); Markov's inequality on $Z(w)^2$ then gives $\mathbb{P}(Z(w) > t \mid \mathcal{D}) \leq \kappa^2 \mathcal{L}(\theta_\mathcal{D})/t^2$, and solving for t at level δ yields, after the generalization substitution below, the $\mathcal{E}_{N,\delta}^{1/2} \delta^{-1/2}$ rate.

By the same argument as (a) (without squaring) and Jensen's inequality,

$$Z(w) \leq \mathbb{E}_{X_0}\left[\left\|\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w) - \Phi_1(X_0; w)\right\| \,\middle|\, \mathcal{D}, w\right] \leq \kappa \sqrt{C_W}\, \mathcal{L}(\theta_\mathcal{D})^{\frac{1}{2(p+1)}} \triangleq A_\mathcal{D},$$

where the last inequality follows from Lemma B.10. Since $Z(w) \leq A_\mathcal{D}$ deterministically,

$$\mathrm{Var}_w(Z(w) \mid \mathcal{D}) \leq \mathbb{E}_w[Z(w)^2 \mid \mathcal{D}] \leq A_\mathcal{D}^2.$$

A one-sided Bernstein inequality then gives, with probability at least $1 - \delta/2$ over w (Wainwright, 2019, Chapter 2),

$$Z(w) < m_\mathcal{D} + \frac{A_\mathcal{D}}{3} \log \frac{2}{\delta} + \sqrt{2 A_\mathcal{D}^2 \log \frac{2}{\delta}}.$$

Both branches above (the Markov branch, and the Lipschitz/fatness branch just derived, via Lemma B.10) are combined, exactly as in the proof of Theorem 4.1-(c), with the generalization event from Lemma B.5-(a) at level $\delta/2$ (over $\mathcal{D}$) and the $\delta/2$-level w-event above, via a union bound giving probability at least $1 - \delta$ over $(\mathcal{D}, w)$ jointly: $\mathcal{L}(\theta_\mathcal{D}) \leq \mathcal{E}_{N,\delta/2}$ is substituted into both branches. Simplifying under $\mathcal{E}_{N,\delta/2} \leq 1$ (and $\log(2/\delta) \geq 1$) yields (b), with the same rate and constant as Theorem 4.1-(c). A sharper Markov branch is given in Remark A.1. □

A.3 Proof of Theorem 4.3

Proof. (a). Lemma B.9 states that, conditioned on $(\mathcal{D}, w)$,

$$\mathbb{E}_{X_0}\left[\left|G_w\left(\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w)\right)\right| \mid \mathcal{D}, w\right] \leq \rho_G^{x,U}\, \kappa \sqrt{\mathcal{L}_{\mathrm{FM}}(\theta_\mathcal{D}; w)}.$$

Taking expectation over w and applying Jensen's inequality,

$$\mathbb{E}_{w,X_0}\left[\left|G_w\left(\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w)\right)\right| \mid \mathcal{D}\right] \leq \rho_G^{x,U}\, \kappa \sqrt{\mathbb{E}_w[\mathcal{L}_{\mathrm{FM}}(\theta_\mathcal{D}; w)]} \leq \rho_G^{x,U}\, \kappa \sqrt{\mathcal{L}(\theta_\mathcal{D})}.$$

(b). The first argument in the minimum follows from Markov's inequality applied to (a)'s bound.

For the second (Lipschitz/fatness) branch, let $Z_\mathrm{g}(\mathcal{D}, w) \triangleq \mathbb{E}_{X_0}\left[\left|G_w\left(\hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w)\right)\right| \mid \mathcal{D}, w\right]$ and $m_\mathcal{D}^\mathrm{g} \triangleq \mathbb{E}_w[Z_\mathrm{g}(\mathcal{D}, w) \mid \mathcal{D}]$. Conditioning on $\mathcal{D}$, Lemma B.9 and Lemma B.10 give

$$0 \leq Z_\mathrm{g} \leq \rho_G^{x,U} \kappa \sqrt{C_W}\, \mathcal{L}(\theta_\mathcal{D})^{\frac{1}{2(p+1)}} \triangleq B_\mathcal{D}.$$

Since Z_{g} is nonnegative and bounded by $B_{\mathcal{D}}$ for every w, a one-sided Bernstein inequality for bounded random variables implies that, with probability at least $1-\delta/2$ over w (Wainwright, 2019, Chapter 2),

$$Z_{\mathrm{g}} \le m^{\mathrm{g}}_{\mathcal{D}} + \frac{B_{\mathcal{D}}}{3}\log\frac{2}{\delta} + \sqrt{2B_{\mathcal{D}}m^{\mathrm{g}}_{\mathcal{D}}\log\frac{2}{\delta}}.$$

Since $m^{\mathrm{g}}_{\mathcal{D}} = \mathbb{E}_w[Z_{\mathrm{g}} \mid \mathcal{D}] \le B_{\mathcal{D}}$,

$$Z_{\mathrm{g}} \le B_{\mathcal{D}} + \frac{B_{\mathcal{D}}}{3}\log\frac{2}{\delta} + B_{\mathcal{D}}\sqrt{2\log\frac{2}{\delta}}.$$

Combining both branches, exactly as in the proofs of Theorem 4.1-(c) and Theorem 4.2-(b), with the generalization event from Lemma B.5-(a) at level $\delta/2$ (over $\mathcal{D}$) and the $\delta/2$-level w-event above via a union bound (probability at least $1-\delta$ over $(\mathcal{D}, w)$ jointly): $\mathcal{L}(\theta_{\mathcal{D}}) \le \mathcal{E}_{N,\delta/2}$ is substituted into $B_{\mathcal{D}}$ above and into the Markov branch. If $\mathcal{E}_{N,\delta/2} \le 1$ and $\log(2/\delta) \ge 1$, substituting and simplifying gives (b), with the same rate as Theorem 4.1-(c). A sharper Markov branch is given in Remark A.1. □

Remark A.1 (Excess-risk refinement)**.** *The main results above are stated using $\mathcal{E}_{N,\delta}$ (5) for simplicity, and because $\mathcal{L}_{\mathrm{CFM}}(\theta_{\mathcal{D}})$ is estimable from held-out samples and is directly aligned with the training objective, whereas $\mathbb{E}_w[\mathcal{L}_{\mathrm{FM}}(\theta_{\mathcal{D}}; w)]$ depends on the generally unavailable marginal velocity field u_t. The Markov branch of each theorem can nonetheless be sharpened, since the θ-independent Bayes term $\mathcal{L}^*_{\mathrm{CFM}}(w)$ (Lemma B.2) cancels when comparing CFM risks. Define*

$$\mathcal{A}_\Theta \triangleq \inf_{\theta\in\Theta} \mathbb{E}_w[\mathcal{L}_{\mathrm{FM}}(\theta; w)], \qquad \mathcal{E}^{\mathrm{exc}}_{N,\delta} \triangleq \mathcal{A}_\Theta + \eta_{\mathcal{D}} + \mathrm{Gen}(N,\delta).$$

Since $\mathcal{L}_{\mathrm{CFM}}$ and $\mathcal{L}_{\mathrm{FM}}$ differ only by the θ-independent constant $\mathcal{L}^\star \triangleq \mathbb{E}_w[\mathcal{L}^*_{\mathrm{CFM}}(w)]$,

$$\mathcal{L}(\theta_{\mathcal{D}}) - \inf_{\theta\in\Theta}\mathcal{L}(\theta) = \mathbb{E}_w[\mathcal{L}_{\mathrm{FM}}(\theta_{\mathcal{D}}; w)] - \mathcal{A}_\Theta,$$

so Lemma B.5 gives $\mathbb{E}_w[\mathcal{L}_{\mathrm{FM}}(\theta_{\mathcal{D}}; w)] \le \mathcal{E}^{\mathrm{exc}}_{N,\delta}$ with the same probability guarantee as $\mathcal{L}(\theta_{\mathcal{D}}) \le \mathcal{E}_{N,\delta}$. Substituting this tighter bound in place of $\mathcal{E}_{N,\delta}$ in each theorem's Markov branch gives refined rates

$$\begin{aligned}
&\textit{Theorem 4.1: } O\big(\mathcal{E}^{\mathrm{exc}}_{N,\delta/2}\,\delta^{-1}\big),\\
&\textit{Theorem 4.2: } O\big((\mathcal{E}^{\mathrm{exc}}_{N,\delta/2})^{1/2}\delta^{-1/2}\big),\\
&\textit{Theorem 4.3: } O\big((\mathcal{E}^{\mathrm{exc}}_{N,\delta/2})^{1/2}\delta^{-1}\big).
\end{aligned}$$

The Lipschitz/fatness branches remain based on $\mathcal{E}_{N,\delta}$, since they use Lemma B.10's uniform-in-w control of $\mathcal{L}_{\mathrm{CFM}}$ directly.

B Supplementary Technical Lemmas

This section proves the technical lemmas used in Appendix A.

For the covering-number argument below, we take $\mathcal{F}$ to be the MLP class defined below. The same proof applies to any bounded, Lipschitz neural-network family with finite covering complexity.

$$\mathcal{F} \triangleq \left\{ \begin{array}{l} u(t,x,w) = (W_D\sigma(\cdot)+b_D)\circ\cdots\circ(W_1\sigma(\cdot)+b_1)([t,x^\top,w^\top]^\top): \\ \quad \sigma(\cdot)\text{ is Lipschitz}, \quad \text{networks width bounded by } M, \quad \sup_{t,x,w}\|u(t,x,w)\| \le U, \\ \quad \max\{\|b_i\|_\infty, \|W_i\|_\infty\} \le K, \quad \forall i = 1,...,D, \\ \quad \|u(t_1,x,w) - u(t_2,x,w)\| \le \gamma\|t_1 - t_2\|, \forall x, w, \\ \quad \|u(t,x_1,w) - u(t,x_2,w)\| \le \kappa_t\|x_1 - x_2\|, \forall w, \quad \int_0^1 \kappa_t dt < \infty \\ \quad \|u(t,x,w_1) - u(t,x,w_2)\| \le \rho_t\|w_1 - w_2\|, \forall x, \quad \int_0^1 \rho_t dt < \infty. \end{array} \right\}, \tag{7}$$

where $\theta = \{(W_i, b_i)\}_{i=1}^D \in \Theta$ summarizes all learnable parameters.

B.1 Relation Between FM Loss and CFM Loss

Lemma B.1 (Relation Between FM Loss and CFM Loss)**.** *For any θ and w,*

$$\mathcal{L}_{\mathrm{CFM}}(\theta; w) = \mathcal{L}_{\mathrm{FM}}(\theta; w) + \mathbb{E}_{t,p_t,p_{1|t}}\left[\|u_t(X_t; w) - u_{t|1}(X_t; w \mid X_1)\|^2\right].$$

In particular, $\mathcal{L}_{\mathrm{FM}}(\theta; w) \le \mathcal{L}_{\mathrm{CFM}}(\theta; w)$.

Proof. Applying the L^2-projection/Pythagorean identity for conditional expectation (Durrett, 2019, Chapter 4), we obtain

$$\begin{aligned}
&\mathbb{E}_{p_1,p_{t|1}}\left[\|u_t^\theta(X_t; w) - u_{t|1}(X_t; w \mid X_1)\|^2\right] \\
&\quad= \mathbb{E}_{p_t}\left[\left\|\mathbb{E}_{p_{1|t}}\left[u_t^\theta(X_t; w) - u_{t|1}(X_t; w \mid X_1) \mid X_t\right]\right\|^2\right] \\
&\quad+ \mathbb{E}_{p_t}\left[\left\|u_t^\theta(X_t; w) - u_{t|1}(X_t; w \mid X_1) - \mathbb{E}_{p_{1|t}}\left[u_t^\theta(X_t; w) - u_{t|1}(X_t; w \mid X_1) \mid X_t\right]\right\|^2\right] \\
&\quad= \mathbb{E}_{p_t}\left[\|u_t^\theta(X_t; w) - u_t(X_t; w)\|^2\right] + \mathbb{E}_{p_t}\left[\|u_t(X_t; w) - u_{t|1}(X_t; w \mid X_1)\|^2\right] \\
&\quad= \mathbb{E}_{p_t}\left[\|u_t^\theta(X_t; w) - u_t(X_t; w)\|^2\right] + (1-t)^{-2}\,\mathbb{E}_{p_t,p_{1|t}}\left[\|\mathbb{E}[X_1 \mid X_t] - X_1\|^2\right].
\end{aligned}$$

Taking expectation on both sides with respect to t yields

$$\begin{aligned}
\mathcal{L}_{\mathrm{CFM}}(\theta; w) &= \mathbb{E}_{t,p_1,p_{t|1}}\left[\|u_t^\theta(X_t; w) - u_{t|1}(X_t; w|X_1)\|^2\right] \\
&= \mathbb{E}_{t,p_t}\left[\|u_t^\theta(X_t; w) - u_t(X_t; w)\|^2\right] + \mathbb{E}_{t,p_t}\left[\|u_t(X_t; w) - u_{t|1}(X_t; w|X_1)\|^2\right] \\
&= \mathcal{L}_{\mathrm{FM}}(\theta; w) + \mathbb{E}_{t,p_t,p_{1|t}}\left[\|u_t(X_t; w) - u_{t|1}(X_t; w|X_1)\|^2\right] \ge \mathcal{L}_{\mathrm{FM}}(\theta; w).
\end{aligned}$$

□

B.2 Bayes CFM Risk

Lemma B.1 shows that $\mathcal{L}_{\mathrm{CFM}}(\theta; w)$ and $\mathcal{L}_{\mathrm{FM}}(\theta; w)$ differ by a term that does not depend on θ. This term is the *irreducible* part of the CFM objective: even the population-optimal, unrestricted choice $u_t^\theta = u_t$ cannot remove it, since it never appears in $\mathcal{L}_{\mathrm{FM}}$ at all. We record this explicitly, since it is what separates the learnable part of the CFM loss (governed by $\mathcal{L}_{\mathrm{FM}}$) from its absolute value.

Lemma B.2 (Bayes CFM Risk)**.** *For every w, $\mathcal{L}^*_{\mathrm{CFM}}(w) \triangleq \inf_{u \text{ measurable}} \mathcal{L}_{\mathrm{CFM}}(u; w)$ is attained at the marginal velocity field $u = u_t(\cdot; w)$, and for every θ, $\mathcal{L}_{\mathrm{CFM}}(\theta; w) - \mathcal{L}^*_{\mathrm{CFM}}(w) = \mathcal{L}_{\mathrm{FM}}(\theta; w)$. Moreover, $0 \le \mathcal{L}^*_{\mathrm{CFM}}(w) \le \bar{R}^2 + n$.*

Proof. Since $\mathcal{L}_{\mathrm{FM}}(\theta; w) \ge 0$ for every θ with equality only when $u_t^\theta = u_t$ (p_t-a.e.), Lemma B.1 shows $\mathcal{L}_{\mathrm{CFM}}(\theta; w) \ge \mathcal{L}_{\mathrm{CFM}}(u_t; w) = \mathbb{E}_{t,p_t,p_{1|t}}[\|u_t(X_t; w) - u_{t|1}(X_t; w \mid X_1)\|^2]$ for every θ, with equality at $\theta = u_t$; hence $\mathcal{L}^*_{\mathrm{CFM}}(w)$ equals this quantity, and the identity is Lemma B.1 rearranged. Nonnegativity is immediate, since $\mathcal{L}^*_{\mathrm{CFM}}(w)$ is an expectation of a squared norm. For the upper bound, since $u_t(X_t; w) = \mathbb{E}[X_1 - X_0 \mid X_t]$ minimizes the conditional second moment among all $\sigma(X_t)$-measurable choices,

$$\mathcal{L}^*_{\mathrm{CFM}}(w) = \mathbb{E}_t\left[\mathrm{Var}(X_1 - X_0 \mid X_t)\right] \le \mathbb{E}\|X_1 - X_0\|^2 = \mathbb{E}\|X_1\|^2 + \mathbb{E}\|X_0\|^2 - 2\,\mathbb{E}[X_1^\top X_0].$$

Since X_0 is drawn independently of (w, X_1) (Section 3) and $\mathbb{E}[X_0] = 0$, the cross term vanishes: $\mathbb{E}[X_1^\top X_0] = \mathbb{E}[X_1]^\top \mathbb{E}[X_0] = 0$. Using $\|X_1\| \le \bar{R}$ a.s. (since $X_1 \in \mathcal{S}(w) \subseteq B_{\bar{R}}(0)$) and $\mathbb{E}\|X_0\|^2 = \mathrm{tr}(I_n) = n$ gives $\mathcal{L}^*_{\mathrm{CFM}}(w) \le \bar{R}^2 + n$. □

We emphasize that Lemma B.2 does *not* characterize when $\mathcal{L}^*_{\mathrm{CFM}}(w) = 0$. All that is used is $\mathcal{L}^*_{\mathrm{CFM}}(w) \ge 0$ and the exact identity $\mathcal{L}_{\mathrm{CFM}}(\theta; w) - \mathcal{L}^*_{\mathrm{CFM}}(w) = \mathcal{L}_{\mathrm{FM}}(\theta; w)$.

B.3 Covering Number Bound

The next lemma bounds the covering number of the hypothesis class $\mathcal{F}$ defined in (7). Recall that $\Theta \subseteq [-K, K]^q$ denotes the set of all possible model parameters θ for elements of $\mathcal{F}$, where q is the dimension of θ. We state the bound for the ordinary dense bounded parameter class.

Lemma B.3 (Covering Number Bound). *For any $\rho > 0$, $\log \mathcal{N}_{\text{cover}}(\rho, \Theta, \|\cdot\|_\infty) \le q \log(1 + 2K/\rho)$. In particular, at $\rho = 1/N$, $\log \mathcal{N}_{\text{cover}}(1/N, \Theta, \|\cdot\|_\infty) \le q \log(1 + 2KN)$.*

Proof. Identify each $\theta \in \Theta$ with a vector in $\mathbb{R}^q$ obtained by stacking all entries of all W_i and b_i, so $\Theta \subseteq [-K, K]^q$ and $\|\cdot\|_\infty$ on Θ is the ambient ℓ_∞ norm on $\mathbb{R}^q$. A uniform grid on each of the q coordinates with mesh size $\rho \le K$ gives a ρ-cover of $[-K, K]^q$ of cardinality at most $(1 + 2K/\rho)^q$; taking logarithms gives the claim. Setting $\rho = 1/N$ gives the second display. □

Remark B.1 (Optional sparsity refinement). *If the trained parameters are additionally known to be J-sparse ($\|\theta\|_0 \le J$ for some $1 \le J \le q$), the same grid argument applied only to the $\binom{q}{j}$ possible supports of each cardinality $j \le J$, combined with the classical bound $\sum_{j=0}^{J} \binom{q}{j} \le (eq/J)^J$ for $1 \le J \le q$ (e.g., Wainwright, 2019) (with no additional condition such as $J \le q/2$), gives the sharper bound*

$$\log \mathcal{N}_{\text{cover}}(1/N, \Theta, \|\cdot\|_\infty) \le J \log \frac{eq}{J} + J \log(1 + 2KN).$$

For $J = 0$, the parameter set contains only the zero vector and the covering number is one.

We write $\mathcal{C}(\rho, \Theta, \|\cdot\|_\infty) \equiv \mathcal{N}_{\text{cover}}(\rho, \Theta, \|\cdot\|_\infty)$ interchangeably below, matching the notation already in use in Lemma B.4.

B.4 Generalization Bound

Let $\widehat{R} > 0$. We define the *truncated loss*, *truncated empirical loss*, and *truncated population loss* as:

$$\ell_\theta^{(\widehat{R})}(t, w, x_0, x_1) \triangleq \left\| u_t^\theta(x_t; w) - u_{t|1}(x_t; w \mid x_1) \right\|^2 \mathbb{1}\left\{ \|x_0\| \le \widehat{R} \right\},$$

$$\widehat{\mathcal{L}}_{\mathcal{D}}^{(\widehat{R})}(\theta) \triangleq \frac{1}{N} \sum_{i=1}^{N} \ell_\theta^{(\widehat{R})}(t^{(i)}, w^{(i)}, x_0^{(i)}, x_1^{(i)}), \quad \mathcal{D} = \{(t^{(i)}, w^{(i)}, x_0^{(i)}, x_1^{(i)})\}_{i=1}^N,$$

$$\mathcal{L}^{(\widehat{R})}(\theta) \triangleq \mathbb{E}_{t,w,x_0,x_1}\left[\ell_\theta^{(\widehat{R})}(t, w, x_0, x_1) \right].$$

Lemma B.4. *The following statements are true:*

(a) $\exists G_{\widehat{R}}, C_{\widehat{R}} > 0$ such that $\ell_\theta^{(\widehat{R})}$ is $G_{\widehat{R}}$-uniformly Lipschitz in θ and uniformly bounded by $C_{\widehat{R}}$. The same is true for both $\widehat{\mathcal{L}}_{\mathcal{D}}^{(\widehat{R})}$ and $\mathcal{L}^{(\widehat{R})}$;

(b) $\exists \tau_{\widehat{R}}$ such that $\sup_{\theta \in \Theta} |\mathcal{L}(\theta) - \mathcal{L}^{(\widehat{R})}(\theta)| \le \tau_{\widehat{R}}$, and $\lim_{\widehat{R} \to \infty} \tau_{\widehat{R}} = 0$;

(c) $\exists \bar{K}$ such that, for any fixed $\beta \in (0, \bar{K})$ and $\delta \in (0, 1)$, if $N \ge C_{\text{rel}} \frac{1}{\beta^2} \log \frac{1}{\delta}$ for a constant C_{rel} depending on $\bar{K}$, then with probability at least $1 - \delta$, $\sup_{\theta \in \Theta} |\widehat{\mathcal{L}}_{\mathcal{D}}(\theta) - \widehat{\mathcal{L}}_{\mathcal{D}}^{(\widehat{R})}(\theta)| \le \tau_{\widehat{R}} + \beta$;

(d) Let $\bar{K}$ be as defined in (c), $\beta \in (0, \bar{K})$, $\delta \in (0, 1)$, and $N \ge C_{\text{rel}} \frac{1}{\beta^2} \log(2/\delta)$. Then, with probability at least $1 - \delta$ over $\mathcal{D}$,

$$\mathcal{L}(\theta_{\mathcal{D}}) < \inf_{\theta \in \Theta} \mathcal{L}(\theta) + \eta_{\mathcal{D}} + 2\epsilon_{N,\widehat{R}}(\delta/2) + 3\tau_{\widehat{R}} + 2\beta,$$

where $\epsilon_{N,\widehat{R}}(\delta) \triangleq C_{\widehat{R}} \sqrt{\frac{\log(2\mathcal{C}(1/N, \Theta, \|\cdot\|_\infty)/\delta)}{2N}} + \frac{2G_{\widehat{R}}}{N}$. Moreover, for any fixed $\widehat{R} > 0$, $\lim_{N \to \infty} \epsilon_{N,\widehat{R}}(\delta) = 0$;

Proof. (a) The uniform Lipschitz continuity of $\ell_\theta^{(\widehat{R})}$ in θ follows directly from the choice of $\mathcal{F}$ and the boundedness of the input (t, w, x_0, x_1). Since $\ell_\theta^{(\widehat{R})}$ is continuous over a compact domain, it is uniformly bounded. The same properties hold for $\widehat{\mathcal{L}}_{\mathcal{D}}^{(\widehat{R})}$ and $\mathcal{L}^{(\widehat{R})}$ because both are convex combinations of $\ell_\theta^{(\widehat{R})}$.

(b) For every $\theta \in \Theta$, the difference between the population and truncated loss is given by:

$$\mathcal{L}(\theta) - \mathcal{L}^{(\widehat{R})}(\theta) = \mathbb{E}\left[\ell_\theta \mathbb{1}\{\|X_0\| > \widehat{R}\} \right], \quad \ell_\theta \triangleq \|u_t^\theta(X_t; w) - (X_1 - X_0)\|^2.$$

Using the triangle inequality and the bounds on the velocity field and X_1 gives

$$\ell_\theta = \|u_t^\theta(X_t; w) - (X_1 - X_0)\|^2 \le 2\|u_t^\theta(X_t; w)\|^2 + 2\|X_1 - X_0\|^2$$
$$\le 2U^2 + 4\|X_1\|^2 + 4\|X_0\|^2 \le 2U^2 + 4\bar{R}^2 + 4\|X_0\|^2.$$

Substituting this upper bound into the expectation gives

$$\mathcal{L}(\theta) - \mathcal{L}^{\widehat{R}}(\theta) \le \mathbb{E}\left[(2U^2 + 4\bar{R}^2 + 4\|X_0\|^2)\mathbb{1}\{\|X_0\| > \widehat{R}\}\right]$$
$$= (2U^2 + 4\bar{R}^2)\mathbb{P}(\|X_0\| > \widehat{R}) + 4\mathbb{E}\left[\|X_0\|^2\mathbb{1}\{\|X_0\| > \widehat{R}\}\right].$$

Since $X_0 \sim \mathcal{N}_n(0, I)$, standard Gaussian and χ^2 tail bounds imply that for $\widehat{R} \ge \sqrt{n}$ there exists a constant $C_n > 0$ such that (Laurent & Massart, 2000, Section 4.1, Lemma 1); see also (Vershynin, 2018, Chapter 3):

$$\mathbb{P}(\|X_0\| > \widehat{R}) \le \exp\left(-\frac{(\widehat{R} - \sqrt{n})^2}{2}\right),$$

$$\mathbb{E}\left[\|X_0\|^2\mathbb{1}\{\|X_0\| > \widehat{R}\}\right] \le C_n(1 + \widehat{R}^2)\exp\left(-\frac{(\widehat{R} - \sqrt{n})^2}{2}\right).$$

Combining these two tail estimates gives the final exponential bound for $\tau_{\widehat{R}}$.

(c) For any $\theta \in \Theta$, the difference between the empirical loss and the truncated empirical loss is

$$\widehat{\mathcal{L}}_{\mathcal{D}}(\theta) - \widehat{\mathcal{L}}_{\mathcal{D}}^{(\widehat{R})}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell_\theta^{(i)}\mathbb{1}\{\|X_0^{(i)}\| > \widehat{R}\}.$$

Applying the pointwise bound from part (b), $\ell_\theta^{(i)} \le 2U^2 + 4\bar{R}^2 + 4\|X_0^{(i)}\|^2$, gives the uniform bound

$$\sup_{\theta\in\Theta}\left|\widehat{\mathcal{L}}_{\mathcal{D}}(\theta) - \widehat{\mathcal{L}}_{\mathcal{D}}^{(\widehat{R})}(\theta)\right| \le \frac{1}{N}\sum_{i=1}^{N} Z_i \triangleq \hat{\tau}_{\widehat{R},\mathcal{D}},$$

where $Z_i \triangleq (2U^2 + 4\bar{R}^2 + 4\|X_0^{(i)}\|^2)\mathbb{1}\{\|X_0^{(i)}\| > \widehat{R}\}$. By definition, $\mathbb{E}[Z_i] = \tau_{\widehat{R}}$.

Since $X_0 \sim \mathcal{N}_n(0, I)$, the squared norm $\|X_0^{(i)}\|^2$ follows a χ_n^2 distribution. Its moment generating function satisfies $\mathbb{E}[e^{\lambda\|X_0^{(i)}\|^2}] = (1 - 2\lambda)^{-n/2}$ for $\lambda < 1/2$; see, e.g., (Vershynin, 2018, Chapter 3). Therefore, for every $\lambda < 1/8$:

$$\mathbb{E}[e^{\lambda Z_i}] \le e^{\lambda(2U^2+4\bar{R}^2)}\mathbb{E}[e^{4\lambda\|X_0^{(i)}\|^2}] = e^{\lambda(2U^2+4\bar{R}^2)}(1 - 8\lambda)^{-n/2} < \infty.$$

Hence Z_i is sub-exponential. It follows that the centered variable $Y_i \triangleq Z_i - \mathbb{E}[Z_i] = Z_i - \tau_{\widehat{R}}$ is also sub-exponential; let $K_{\widehat{R}} \triangleq \|Y_i\|_{\psi_1} < \infty$ denote the sub-exponential norm (Vershynin, 2018, Chapter 2).

The Bernstein inequality for independent mean-zero sub-exponential random variables implies there exists a universal constant $c > 0$ such that for every $\beta > 0$ (Vershynin, 2018, Chapter 2):

$$\mathbb{P}\left(\frac{1}{N}\sum_{i=1}^{N} Y_i \ge \beta\right) \le \exp\left(-cN\min\left\{\frac{\beta^2}{K_{\widehat{R}}^2}, \frac{\beta}{K_{\widehat{R}}}\right\}\right).$$

Equivalently,

$$\mathbb{P}\left(\hat{\tau}_{\widehat{R},\mathcal{D}} - \tau_{\widehat{R}} \ge \beta\right) \le \exp\left(-cN\min\left\{\frac{\beta^2}{K_{\widehat{R}}^2}, \frac{\beta}{K_{\widehat{R}}}\right\}\right).$$

To ensure that $\mathbb{P}(\hat{\tau}_{\widehat{R},\mathcal{D}} \ge \tau_{\widehat{R}} + \beta) \le \delta$, it is sufficient to choose a sample size N satisfying

$$N \ge \frac{1}{c}\max\left\{\frac{K_{\widehat{R}}^2}{\beta^2}, \frac{K_{\widehat{R}}}{\beta}\right\}\log\frac{1}{\delta}.$$

To obtain a sample complexity independent of $\widehat{R}$, we bound $K_{\widehat{R}}$. Using the centering property of sub-exponential norms, we have $K_{\widehat{R}} \le 2\|Z_i\|_{\psi_1}$ (Vershynin, 2018, Chapter 2).

Let $A \triangleq 2U^2 + 4\bar{R}^2$. By definition, $Z_i = (A + 4\|X_0^{(i)}\|^2)\mathbb{1}\{\|X_0^{(i)}\| > \widehat{R}\}$. Since $A, \|X_0^{(i)}\|^2 \ge 0$, it holds pointwise that $0 \le Z_i \le A + 4\|X_0^{(i)}\|^2$. By the monotonicity of the Orlicz norm, we have $\|Z_i\|_{\psi_1} \le \|A + 4\|X_0^{(i)}\|^2\|_{\psi_1}$. Applying the triangle inequality yields:

$$\|Z_i\|_{\psi_1} \le \|A\|_{\psi_1} + 4\|\|X_0^{(i)}\|^2\|_{\psi_1}.$$

Using the fact that $\|X_0^{(i)}\|^2 = \sum_{j=1}^n \xi_j^2$, where $\xi_j \sim \mathcal{N}(0,1)$ are i.i.d., we have

$$\|Z_i\|_{\psi_1} \le \frac{A}{\ln 2} + 4\sum_{j=1}^n \|\xi_j^2\|_{\psi_1} = \frac{2U^2 + 4\bar{R}^2}{\ln 2} + 4n\|\xi_1^2\|_{\psi_1}.$$

Thus, we can define a constant $\bar{K}$ independent of $\widehat{R}$:

$$K_{\widehat{R}} \le 2\|Z_i\|_{\psi_1} \le \bar{K} \triangleq 2\left(\frac{2U^2 + 4\bar{R}^2}{\ln 2} + 4n\|\xi_1^2\|_{\psi_1}\right).$$

Substituting $\bar{K}$ into the Bernstein bound shows that for any $\beta \in (0, \bar{K})$ and $\delta \in (0,1)$, if

$$N \ge \frac{\bar{K}}{c}\frac{1}{\beta^2}\log\frac{1}{\delta} \triangleq C_{\mathrm{rel}}\frac{1}{\beta^2}\log\frac{1}{\delta},$$

then with probability at least $1-\delta$ over $\mathcal{D}$,

$$\sup_{\theta\in\Theta}\left|\widehat{\mathcal{L}}_{\mathcal{D}}(\theta) - \widehat{\mathcal{L}}_{\mathcal{D}}^{(\widehat{R})}(\theta)\right| \le \tau_{\widehat{R}} + \beta.$$

Since K depends only on $(U, \bar{R}, n)$ and not on $\widehat{R}$, the sample complexity is independent of $\widehat{R}$.

(d) Let $\widehat{R} > 0$. Using a covering-number discretization with an ϵ-cover of Θ, the $G_{\widehat{R}}$-Lipschitz continuity of the truncated loss implies:

$$\sup_{\theta\in\Theta}|\mathcal{L}^{(\widehat{R})}(\theta) - \widehat{\mathcal{L}}_{\mathcal{D}}^{(\widehat{R})}(\theta)| \le 2G_{\widehat{R}}\epsilon + \max_{k=1,\dots,\mathcal{C}(\epsilon,\Theta,\|\cdot\|_\infty)}|\mathcal{L}^{(\widehat{R})}(\theta_k) - \widehat{\mathcal{L}}_{\mathcal{D}}^{(\widehat{R})}(\theta_k)|.$$

Applying Hoeffding's inequality and a union bound over $\mathcal{C}(\epsilon, \Theta, \|\cdot\|_\infty)$ points, and setting $\epsilon = 1/N$, we obtain (Wainwright, 2019, Chapter 2):

$$\mathbb{P}_{\mathcal{D}}\left(\sup_{\theta\in\Theta}|\mathcal{L}^{(\widehat{R})}(\theta) - \widehat{\mathcal{L}}_{\mathcal{D}}^{(\widehat{R})}(\theta)| \le \epsilon_{N,\widehat{R}}(\delta/2)\right) \ge 1 - \frac{\delta}{2},$$

where

$$\epsilon_{N,\widehat{R}}(\delta/2) \triangleq C_{\widehat{R}}\sqrt{\frac{\log(4\mathcal{C}(1/N,\Theta,\|\cdot\|)/\delta)}{2N}} + \frac{2G_{\widehat{R}}}{N}.$$

Let $\beta > 0$ and $N \ge C_{\mathrm{rel}}\frac{1}{\beta^2}\log\frac{2}{\delta}$. Define the event:

$$E_{\mathcal{D}} \triangleq \left\{\sup_{\theta\in\Theta}|\mathcal{L}^{(\widehat{R})}(\theta) - \widehat{\mathcal{L}}_{\mathcal{D}}^{(\widehat{R})}(\theta)| \le \epsilon_{N,\widehat{R}}(\delta/2)\right\} \bigcap \left\{\sup_{\theta\in\Theta}|\widehat{\mathcal{L}}_{\mathcal{D}}(\theta) - \widehat{\mathcal{L}}_{\mathcal{D}}^{(\widehat{R})}(\theta)| \le \tau_{\widehat{R}} + \beta\right\}.$$

By Lemma B.4-(c) and the bound above for the truncated process, each event in the intersection has probability at least $1 - \delta/2$, so $\mathbb{P}_{\mathcal{D}}(E_{\mathcal{D}}) \ge 1 - \delta$. On this event, we have the following chain of

inequalities:

$$\begin{aligned}
\mathcal{L}(\theta_{\mathcal{D}}) &\le \mathcal{L}^{(\widehat{R})}(\theta_{\mathcal{D}}) + \tau_{\widehat{R}} && (10)\\
&\le \widehat{\mathcal{L}}^{(\widehat{R})}_{\mathcal{D}}(\theta_{\mathcal{D}}) + \epsilon_{N,\widehat{R}}(\delta/2) + \tau_{\widehat{R}} && (11)\\
&\le \widehat{\mathcal{L}}_{\mathcal{D}}(\theta_{\mathcal{D}}) + \epsilon_{N,\widehat{R}}(\delta/2) + 2\tau_{\widehat{R}} + \beta && (12)\\
&< \inf_{\theta\in\Theta} \widehat{\mathcal{L}}_{\mathcal{D}}(\theta) + \eta_{\mathcal{D}} + \epsilon_{N,\widehat{R}}(\delta/2) + 2\tau_{\widehat{R}} + \beta && (13)\\
&\le \inf_{\theta\in\Theta} \widehat{\mathcal{L}}^{(\widehat{R})}_{\mathcal{D}}(\theta) + \eta_{\mathcal{D}} + \epsilon_{N,\widehat{R}}(\delta/2) + 3\tau_{\widehat{R}} + 2\beta && (14)\\
&\le \inf_{\theta\in\Theta} \mathcal{L}^{(\widehat{R})}(\theta) + \eta_{\mathcal{D}} + 2\epsilon_{N,\widehat{R}}(\delta/2) + 3\tau_{\widehat{R}} + 2\beta\\
&\le \inf_{\theta\in\Theta} \mathcal{L}(\theta) + \eta_{\mathcal{D}} + 2\epsilon_{N,\widehat{R}}(\delta/2) + 3\tau_{\widehat{R}} + 2\beta.
\end{aligned}$$

Here, (10) follows from Lemma B.4-(b), (11) and the penultimate inequality follow from the first event in $E_{\mathcal{D}}$, (12) and (14) follow from the second event in $E_{\mathcal{D}}$, (13) follows from the definition of $\eta_{\mathcal{D}}$, and the last inequality uses the pointwise bound $\mathcal{L}^{(\widehat{R})}(\theta) \le \mathcal{L}(\theta)$. □

Remark B.2 (Explicit truncated-loss constants)**.** *The constants $C_{\widehat{R}}, G_{\widehat{R}}$ in Lemma B.4-(a) admit the following explicit bounds. On $\{\|x_0\| \le \widehat{R}\}$, $\|u^\theta_t(x_t; w)\| \le U$ and $\|u_{t|1}(x_t; w \mid x_1)\| = \|x_1 - x_0\| \le \bar{R} + \widehat{R}$ (using $x_1 \in \mathcal{S}(w) \subseteq B_{\bar{R}}(0)$), so*

$$C_{\widehat{R}} \le 2U^2 + 2(\bar{R} + \widehat{R})^2 \le 2U^2 + 4\bar{R}^2 + 4\widehat{R}^2.$$

*For $G_{\widehat{R}}$: on $\{\|x_0\| \le \widehat{R}\}$, $x_t = (1-t)x_0 + tx_1$ with $x_1 \in B_{\bar{R}}(0)$ gives $\|x_t\| \le \bar{R} + \widehat{R}$, and $\mathcal{W}$ is bounded (**A4**), so the network input $z = [t, x_t^\top, w^\top]^\top$ satisfies $\|z\| \le Z_{\widehat{R}} := 1 + \widehat{R} + \bar{R} + \sup_{w\in\mathcal{W}} \|w\| = O(1 + \widehat{R})$. Writing u^θ as the depth-D composition in (7), a layer-by-layer telescoping bound (perturbing one layer's weights at a time and propagating through the remaining layers, each $L_\sigma M K$-Lipschitz in its input by the width-M, weight-bound-K, activation-Lipschitz-L_σ structure of (7)) gives*

$$\|u^\theta(z) - u^{\theta'}(z)\| \le C_\Theta\, Z_{\widehat{R}}\, \|\theta - \theta'\|_\infty \le C'_\Theta\, (1 + \widehat{R})\, \|\theta - \theta'\|_\infty,$$

with C_Θ, C'_Θ depending only on (D, M, K, L_σ). Since $\|a-b\|^2 - \|a'-b\|^2 \le (\|a-b\| + \|a'-b\|)\|a-a'\|$, and, on $\{\|x_0\| \le \widehat{R}\}$, both $\|u^\theta(z)\|, \|u^{\theta'}(z)\| \le U$ and $\|u_{t|1}(x_t; w \mid x_1)\| = \|x_1 - x_0\| \le \bar{R} + \widehat{R}$,

$$\left|\ell^{(\widehat{R})}_\theta - \ell^{(\widehat{R})}_{\theta'}\right| \le 2(U + \bar{R} + \widehat{R}) \cdot C'_\Theta (1 + \widehat{R})\, \|\theta - \theta'\|_\infty \le C_{\Theta,\mathrm{par}} (1 + \widehat{R})^2\, \|\theta - \theta'\|_\infty,$$

for a constant $C_{\Theta,\mathrm{par}}$ depending only on $(D, M, K, L_\sigma, U, \bar{R})$. Hence $G_{\widehat{R}} \le C_{\Theta,\mathrm{par}}(1 + \widehat{R})^2$.

Lemma B.5 (Finite-Sample Generalization Bound)**.** *Let $\widehat{R}_{N,\delta} \triangleq \sqrt{n} + \sqrt{2\log(N/\delta)}$,*

$$\begin{aligned}
C^\star(N,\delta) &\triangleq 2U^2 + 4\bar{R}^2 + 8n + 16\log\tfrac{N}{\delta},\\
G^\star(N,\delta) &\triangleq 3C_{\Theta,\mathrm{par}}\big(1 + n + 2\log\tfrac{N}{\delta}\big),\\
\tau^\star(N,\delta) &\triangleq \Big[2U^2 + 4\bar{R}^2 + 4C_n\big(1 + 2n + 4\log\tfrac{N}{\delta}\big)\Big]\frac{\delta}{N},\\
N_0(\delta) &\triangleq \frac{C_{\mathrm{rel}}\log(2/\delta)}{\bar{K}^2},
\end{aligned}$$

using the explicit $C_{\widehat{R}}, G_{\widehat{R}}$ bounds of Remark B.2 evaluated at $\widehat{R} = \widehat{R}_{N,\delta}$, and the dense covering number of Lemma B.3. Define

$$\mathrm{Gen}(N,\delta) \triangleq \sqrt{2}\, C^\star(N,\delta)\sqrt{\frac{q\log(1 + 2KN) + \log(4/\delta)}{N}} + \frac{4G^\star(N,\delta)}{N} + 3\tau^\star(N,\delta) + 2\sqrt{\frac{C_{\mathrm{rel}}\log(2/\delta)}{N}}.$$

Then:

(a) ***Finite-N.*** *For every $\delta \in (0,1)$ and $N > N_0(\delta)$, with probability at least $1-\delta$ over $\mathcal{D}$,*

$$\mathcal{L}(\theta_{\mathcal{D}}) \le \inf_{\theta\in\Theta} \mathcal{L}(\theta) + \eta_{\mathcal{D}} + \mathrm{Gen}(N,\delta).$$

(b) ***Sample complexity.*** *For every $\epsilon \in (0,1]$, $\delta \in (0,1)$, and*

$$N \ge N_{\mathrm{gen}}(\epsilon,\delta) \triangleq \max\Big\{N_0(\delta),\ \widetilde{O}\Big(\tfrac{(n+\log\frac{1}{\epsilon\delta})^2(q+\log\frac{1}{\delta})}{\epsilon^2}\Big)\Big\},$$

where $\widetilde{O}(\cdot)$ hides constants $U, \bar{R}, C_{\Theta,\mathrm{par}}, C_n, C_{\mathrm{rel}}, \bar{K}$, one additional factor of $\log(1/(\epsilon\delta))$ arising from the self-referential inversion of $\mathrm{Gen}(N,\delta) \le \epsilon$, and purely logarithmic (including iterated-logarithmic) dependence on n, q, K; with probability at least $1-\delta$ over $\mathcal{D}$,

$$\mathcal{L}(\theta_{\mathcal{D}}) \le \inf_{\theta\in\Theta} \mathcal{L}(\theta) + \eta_{\mathcal{D}} + \epsilon.$$

Moreover, $\mathrm{Gen}(N,\delta) = O\big((n+\log\frac{N}{\delta})\sqrt{(q\log(1+2KN)+\log(1/\delta))/N}\big)$, and for fixed $n, q, K, U, \bar{R}, \delta$ (and other architecture constants), as $N\to\infty$,

$$\mathrm{Gen}(N,\delta) = O\left(\sqrt{\frac{(\log N)^3}{N}}\right),$$

and for fixed architecture/dimension and δ, as $\epsilon \to 0$,

$$N_{\mathrm{gen}}(\epsilon) = O\big(\epsilon^{-2}\log^3(1/\epsilon)\big).$$

If instead only the architecture (and dimension) is fixed while δ is allowed to vary, the same bound tracks δ explicitly: since $(n+\log\frac{N}{\delta})\sqrt{q\log(1+2KN)+\log(1/\delta)} = O\big((\log N+\log(1/\delta))^{3/2}\big) = O\big((\log N)^{3/2}+(\log(1/\delta))^{3/2}\big)$ for fixed architecture, absorbing the $(\log N)^{3/2}$ term into $\widetilde{O}(\cdot)$ gives

$$\mathrm{Gen}(N,\delta) = \widetilde{O}\big(\,\mathrm{polylog}(1/\delta)\cdot N^{-1/2}\big),$$

where $\widetilde{O}(\cdot)$ suppresses logarithmic factors in N (and the fixed architecture constants) while keeping the δ-dependence explicit. Correspondingly, for fixed architecture/dimension and $\epsilon\to 0$,

$$N_{\mathrm{gen}}(\epsilon,\delta) = \widetilde{O}\big(\,\mathrm{polylog}(1/\delta)\cdot \epsilon^{-2}\big),$$

where $\widetilde{O}(\cdot)$ now suppresses logarithmic factors in $1/\epsilon$ while keeping the δ-dependence explicit.

Proof. Choice of $\widehat{R}_{N,\delta}$ and $N_0(\delta)$. Taking $\widehat{R} = \widehat{R}_{N,\delta}$ in the standard Gaussian concentration bound for the 1-Lipschitz map $x \mapsto \|x\|_2$ (see, e.g., (Vershynin, 2018, Chapter 5)) gives $e^{-(\widehat{R}_{N,\delta}-\sqrt{n})^2/2} = \delta/N$ exactly, and $\widehat{R}^2_{N,\delta} \le 2n + 4\log(N/\delta)$ (via $(a+b)^2 \le 2a^2+2b^2$), yielding $\tau_{\widehat{R}_{N,\delta}} \le \tau^\star(N,\delta)$; since $\widehat{R}_{N,\delta}$ is a deterministic function of (N,δ) chosen before observing $\mathcal{D}$, this costs no probability budget. Substituting $\widehat{R}_{N,\delta}$ into Remark B.2's bounds gives $C_{\widehat{R}_{N,\delta}} \le C^\star(N,\delta)$ and $G_{\widehat{R}_{N,\delta}} \le G^\star(N,\delta)$ (using $\widehat{R}^2_{N,\delta} \le 2n+4\log(N/\delta)$ and $(1+\widehat{R}_{N,\delta})^2 \le 3(1+n+2\log(N/\delta))$).

Choice of β. Lemma B.4-(c) requires $\beta \in (0,\bar{K})$ and $N \ge C_{\mathrm{rel}}\beta^{-2}\log(2/\delta)$; the minimal such β for given N is $\beta_{N,\delta} = \sqrt{C_{\mathrm{rel}}\log(2/\delta)/N}$, contributing exactly $2\beta_{N,\delta} = 2\sqrt{C_{\mathrm{rel}}\log(2/\delta)/N}$ to Lemma B.4-(d)'s inequality. Enforcing $\beta_{N,\delta} < \bar{K}$ strictly gives $N > N_0(\delta) = C_{\mathrm{rel}}\log(2/\delta)/\bar{K}^2$.

(a). Substituting $\widehat{R}_{N,\delta}$ and $\beta_{N,\delta}$ into Lemma B.4-(d)'s inequality $\mathcal{L}(\theta_{\mathcal{D}}) < \inf_\theta \mathcal{L}(\theta) + \eta_{\mathcal{D}} + 2\epsilon_{N,\widehat{R}}(\delta/2) + 3\tau_{\widehat{R}} + 2\beta$, and substituting Lemma B.3's covering-number bound $\log\mathcal{N}_{\mathrm{cover}}(1/N,\Theta,\|\cdot\|_\infty) \le q\log(1+2KN)$ into $\epsilon_{N,\widehat{R}}(\delta/2) = C_{\widehat{R}}\sqrt{\log(4\mathcal{N}_{\mathrm{cover}}(1/N,\Theta)/\delta)/(2N)} + 2G_{\widehat{R}}/N$ (note: $\delta/2$ enters here, giving $\log(4\cdot/\delta)$, distinct from the $\log(2/\delta)$ used for β above, which stems from a separate $\delta/2$-level event), gives $2\epsilon_{N,\widehat{R}_{N,\delta}}(\delta/2) \le \sqrt{2}\,C^\star(N,\delta)\sqrt{(q\log(1+2KN)+\log(4/\delta))/N} + 4G^\star(N,\delta)/N$. Summing with $3\tau^\star(N,\delta) + 2\beta_{N,\delta}$ gives exactly $\mathrm{Gen}(N,\delta)$.

Leading-order rate. For $N > N_0(\delta)$, the term $2\sqrt{C_{\mathrm{rel}}\log(2/\delta)/N} = O(\sqrt{\log(1/\delta)/N})$; the term $4G^\star(N,\delta)/N = O((n+\log(N/\delta))/N)$; and the term $3\tau^\star(N,\delta) = O((n+\log(N/\delta))\delta/N)$ (in

particular $\tau^\star(N,\delta) = O(\log(N/\delta)/N)$). Direct comparison (for fixed n, q, K, δ, as $N \to \infty$) shows all three are dominated by the first term, giving the stated leading-order rate; for fixed architecture/dimension and δ, $\log(N/\delta) = O(\log N)$ and $\log(1 + 2KN) = O(\log N)$, giving $\mathrm{Gen}(N,\delta) = O(\sqrt{(\log N)^3/N})$.

(b). Solving $\mathrm{Gen}(N,\delta) \le \epsilon$ for N (substituting the ansatz $N = N_{\mathrm{gen}}(\epsilon,\delta)$ back into $\log(N/\delta)$ and $\log(1+2KN)$ and verifying the resulting inequality holds, exactly as in the leading-order computation above, up to enlarging the constant hidden in $\widetilde{O}$) yields the stated sample complexity; taking n, q, K, δ fixed and $\epsilon \to 0$ gives $N_{\mathrm{gen}}(\epsilon) = O(\epsilon^{-2}\log^3(1/\epsilon))$, since $(n + \log(1/(\epsilon\delta)))^2 = O(\log^2(1/\epsilon))$ and the extra $\log(1/(\epsilon\delta))$ factor contributes one more power of $\log(1/\epsilon)$. □

B.5 Continuity of the Gap Function and the Solution Distribution in the Parameter

The Lipschitz continuity of the CFM loss in w (Lemma B.8 below) requires a coupling between $p(\cdot \mid w_1)$ and $p(\cdot \mid w_2)$ with small expected transport cost. The Hausdorff-Lipschitz regularity of the feasible sets $\mathcal{X}_w$ (**A**2) alone does not furnish such a coupling between the *solution distributions* $p(\cdot \mid w_1), p(\cdot \mid w_2)$: two distributions supported on Hausdorff-close solution sets can still place their mass on separated branches. We therefore assume this coupling directly in **A**3. This assumption is a property of the offline solution-label distribution itself, not a consequence of feasibility-set regularity or exact support alone.

Lemma B.6 (Lipschitz Continuity of the Gap Function in x)**.** *Under **A**1, for every $w \in \mathcal{W}$ and $x_1, x_2 \in B_{\bar{R}}(0)$,*

$$|G_w(x_1) - G_w(x_2)| \le \rho_G^x \|x_1 - x_2\|, \qquad \rho_G^x \triangleq 2\bar{R}L_F + \bar{M},$$

where $\bar{M} \triangleq \sup_{(w,x)\in\mathcal{W}\times B_{\bar{R}}(0)} \|F_w(x)\| < \infty$, finite because F_w is jointly continuous on the compact set $\mathcal{W} \times B_{\bar{R}}(0)$.

Proof. Fix w, $\epsilon > 0$, and choose $y_1 \in \mathcal{X}_w$ with $F_w(x_1)^\top(x_1 - y_1) > G_w(x_1) - \epsilon$ (possible since $G_w(x_1) = \sup_{y\in\mathcal{X}_w} F_w(x_1)^\top(x_1 - y)$). Then

$$F_w(x_1)^\top(x_1 - y_1) = F_w(x_2)^\top(x_2 - y_1) + [F_w(x_1) - F_w(x_2)]^\top(x_1 - y_1) + F_w(x_2)^\top(x_1 - x_2),$$

a direct algebraic identity verified by expanding the right-hand side. Since $y_1 \in \mathcal{X}_w$, $F_w(x_2)^\top(x_2 - y_1) \le G_w(x_2)$ by definition of $G_w(x_2)$ as a supremum over $y \in \mathcal{X}_w$ – this step only needs $y_1 \in \mathcal{X}_w$, regardless of whether x_1 or x_2 themselves lie in $\mathcal{X}_w$, which is exactly what lets the lemma apply to x_1, x_2 anywhere in $B_{\bar{R}}(0)$ rather than only on $\mathcal{X}_w$. Bounding the remaining two terms: $\|[F_w(x_1) - F_w(x_2)]^\top(x_1 - y_1)\| \le L_F\|x_1 - x_2\| \cdot \|x_1 - y_1\| \le 2\bar{R}L_F\|x_1 - x_2\|$ (using $\|x_1 - y_1\| \le \|x_1\| + \|y_1\| \le 2\bar{R}$ since $x_1, y_1 \in B_{\bar{R}}(0)$), and $F_w(x_2)^\top(x_1 - x_2) \le \bar{M}\|x_1 - x_2\|$. Combining, $G_w(x_1) - \epsilon < G_w(x_2) + (2\bar{R}L_F + \bar{M})\|x_1 - x_2\|$. Letting $\epsilon \to 0$ and swapping x_1, x_2 gives the claim. □

Lemma B.7 (Continuity of the Solution Distribution in w)**.** *Under **A**3, there exists a coupling (X_1, X_2) of $p(\cdot \mid w_1)$ and $p(\cdot \mid w_2)$ such that*

$$\mathbb{E}\|X_1 - X_2\| \le H'\|w_1 - w_2\|$$

*where the constant H' is as defined in **A**3.*

Proof. This is exactly **A**3's Wasserstein-continuity requirement on the labeling process. □

B.6 Lipschitz Continuity of the CFM Loss in the Parameter

Lemma B.8 (Lipschitz Continuity of CFM Loss in w)**.** *For any fixed $\mathcal{D}$, $\mathcal{L}_{\mathrm{CFM}}(\theta_{\mathcal{D}}; w)$ is $L^W_{\mathrm{CFM}} \triangleq 2(U + \bar{R} + \mathbb{E}[\|X_0\|])(\int_0^1 \rho_t dt + (1 + \int_0^1 t\kappa_t dt)H')$-Lipschitz in w, with H' as in Lemma B.7.*

Proof. Let (X_1, X_2) be the coupling of $p(\cdot \mid w_1)$ and $p(\cdot \mid w_2)$ from Lemma B.7 (**A**3), satisfying $\mathbb{E}\|X_1 - X_2\| \le H'\|w_1 - w_2\|$. Let $X_0 \sim p_0$, and denote by $X_t^{(1)} \triangleq (1-t)X_0 + tX_1$ and $X_t^{(2)} \triangleq (1-t)X_0 + tX_2$. Then

$$\begin{aligned}
&\left|\|u_t^{\theta_{\mathcal{D}}}(X_t^{(1)}; w_1) - u_{t|1}(X_t^{(1)}; w_1 \mid X_1)\|^2 - \|u_t^{\theta_{\mathcal{D}}}(X_t^{(2)}; w_2) - u_{t|1}(X_t^{(2)}; w_2 \mid X_2)\|^2\right| \\
\le &\left|\|u_t^{\theta_{\mathcal{D}}}(X_t^{(1)}; w_1)\| + \|u_{t|1}(X_t^{(1)}; w_1 \mid X_1)\| + \|u_t^{\theta_{\mathcal{D}}}(X_t^{(2)}; w_2)\| + \|u_{t|1}(X_t^{(2)}; w_2 \mid X_2)\|\right| \\
&\cdot \left\|u_t^{\theta_{\mathcal{D}}}(X_t^{(1)}; w_1) - u_t^{\theta_{\mathcal{D}}}(X_t^{(1)}; w_2) + u_t^{\theta_{\mathcal{D}}}(X_t^{(1)}; w_2) - u_t^{\theta_{\mathcal{D}}}(X_t^{(2)}; w_2) - (X_1 - X_2)\right\| \\
\le &2\left|U + \bar{R} + \|X_0\|\right| \cdot \left(\rho_t\|w_1 - w_2\| + \kappa_t\|X_t^{(1)} - X_t^{(2)}\| + \|X_1 - X_2\|\right).
\end{aligned}$$

Here, the first inequality uses the facts that for scalars a, b, $|a^2 - b^2| = |a-b||a+b|$, and for vectors a, b, $|\|a\| - \|b\|| \le \|a - b\|$.

Moreover,

$$\begin{aligned}
&|\mathcal{L}_{\mathrm{CFM}}(\theta_{\mathcal{D}}; w_1) - \mathcal{L}_{\mathrm{CFM}}(\theta_{\mathcal{D}}; w_2)| \\
\le &\mathbb{E}\left[2\left|U + \bar{R} + \|X_0\|\right|\left(\int_0^1 \rho_t dt\|w_1 - w_2\| + \int_0^1 \kappa_t\|X_t^{(1)} - X_t^{(2)}\|dt + \|X_1 - X_2\|\right)\right] \\
\le &2\left(U + \bar{R} + \mathbb{E}[\|X_0\|]\right)\int_0^1 \rho_t dt\|w_1 - w_2\| \\
&+2\left(U + \bar{R} + \mathbb{E}[\|X_0\|]\right)\left(1 + \int_0^1 t\kappa_t dt\right)\mathbb{E}\left[\|X_1 - X_2\|\right] \\
\le &2\left(U + \bar{R} + \mathbb{E}[\|X_0\|]\right)\left(\int_0^1 \rho_t dt + \left(1 + \int_0^1 t\kappa_t dt\right)H'\right)\|w_1 - w_2\|,
\end{aligned}$$

where the second inequality uses $X_t^{(1)} - X_t^{(2)} = t(X_1 - X_2)$ together with the independence of X_0 from the coupling (X_1, X_2), which holds because X_0 is drawn independently of (w, X_1) during training (Section 3) and X_2 is constructed from the same randomness as X_1 via Lemma B.7 so that $\mathbb{E}\left[|U + \bar{R} + \|X_0\|| \cdot \|X_1 - X_2\|\right] = \mathbb{E}\left[|U + \bar{R} + \|X_0\||\right] \cdot \mathbb{E}\|X_1 - X_2\|$, and the last inequality substitutes the bound $\mathbb{E}\|X_1 - X_2\| \le H'\|w_1 - w_2\|$ from Lemma B.7. Note that only this expectation bound is needed; no almost-sure pathwise bound on $\|X_1 - X_2\|$ is required anywhere in the argument. □

B.7 Parameter-Pointwise Expected Gap Bound

Lemma B.9 (w-pointwise Expected Gap Bound)**.** *Conditioned on* $(\mathcal{D}, w)$,

$$\mathbb{E}_{X_0}\left[\left|G_w\left(\hat{\Phi}_1^{\theta_{\mathcal{D}}}(X_0; w)\right)\right| \mid \mathcal{D}, w\right] \le \rho_G^{x,U}\,\kappa\,\sqrt{\mathcal{L}_{\mathrm{FM}}(\theta_{\mathcal{D}}; w)} \le \rho_G^{x,U}\,\kappa\,\sqrt{\mathcal{L}_{\mathrm{CFM}}(\theta_{\mathcal{D}}; w)},$$

where $\rho_G^{x,U} \triangleq \left(\mathbb{E}_{X_0}\left[(L_F\|X_0\| + L_F U + 3\bar{R}L_F + \bar{M})^2\right]\right)^{1/2} < \infty$ *and* $\bar{M} \triangleq \sup_{(w,x)\in\mathcal{W}\times B_{\bar{R}}(0)}\|F_w(x)\|$.

Proof. Conditioned on $(\mathcal{D}, w)$, let $X_0 \sim N(0, I)$ be independent of $(\mathcal{D}, w)$, $X_1 \triangleq \hat{\Phi}_1^{\theta_{\mathcal{D}}}(X_0; w)$, and $X_2 \triangleq \Phi_1(X_0; w)$. Since $X_2 = \Phi_1(X_0; w)$ with $X_0 \sim p_0$ has law $p(\cdot \mid w)$, and $p(\cdot \mid w)$ is supported on $\mathcal{S}(w)$ by Section 3, $X_2 \in \mathcal{S}(w) \subseteq B_{\bar{R}}(0)$ almost surely, so $G_w(X_2) = 0$ almost surely (Section 4). The learned velocity is uniformly bounded by U in (7); integrating the learned ODE over $[0, 1]$ gives $\|X_1\| \le \|X_0\| + U$.

We first record a gap comparison that does not require X_1 to lie in $B_{\bar{R}}(0)$. Fix $\epsilon > 0$ and choose $y_1 \in \mathcal{X}_w$ with $F_w(X_1)^\top(X_1 - y_1) > G_w(X_1) - \epsilon$. The same algebra as Lemma B.6, using $X_2, y_1 \in B_{\bar{R}}(0)$, gives

$$\begin{aligned}
G_w(X_1) - G_w(X_2) &< [F_w(X_1) - F_w(X_2)]^\top(X_1 - y_1) + F_w(X_2)^\top(X_1 - X_2) + \epsilon \\
&\le \left(L_F(\|X_1\| + \bar{R}) + \bar{M}\right)\|X_1 - X_2\| + \epsilon.
\end{aligned}$$

For the reverse direction, choose $y_2 \in \mathcal{X}_w$ with $F_w(X_2)^\top(X_2 - y_2) > G_w(X_2) - \epsilon$. Expanding around X_1 gives

$$\begin{aligned} G_w(X_2) - G_w(X_1) &< [F_w(X_2) - F_w(X_1)]^\top (X_2 - y_2) + F_w(X_1)^\top (X_2 - X_1) + \epsilon \\ &\le \left(3\bar{R}L_F + \bar{M} + L_F\|X_1\|\right)\|X_1 - X_2\| + \epsilon, \end{aligned}$$

where the last inequality uses $\|F_w(X_1)\| \le \|F_w(X_2)\| + L_F\|X_1 - X_2\| \le \bar{M} + L_F(\|X_1\| + \bar{R})$. Letting $\epsilon \to 0$ in both displays and using $G_w(X_2) = 0$ yields

$$|G_w(X_1)| \le \left(L_F\|X_1\| + 3\bar{R}L_F + \bar{M}\right)\|X_1 - X_2\|.$$

Substituting $\|X_1\| \le \|X_0\| + U$ and applying Cauchy–Schwarz gives

$$\mathbb{E}_{X_0}[|G_w(X_1)| \mid \mathcal{D}, w] \le \left(\mathbb{E}_{X_0}\left[(L_F\|X_0\| + L_F U + 3\bar{R}L_F + \bar{M})^2\right]\right)^{1/2} \sqrt{\mathbb{E}_{X_0}[\|X_1 - X_2\|^2 \mid \mathcal{D}, w]}$$

$$\le \rho_G^{x,U}\kappa\sqrt{\mathcal{L}_{\mathrm{FM}}(\theta_{\mathcal{D}}; w)} \le \rho_G^{x,U}\kappa\sqrt{\mathcal{L}_{\mathrm{CFM}}(\theta_{\mathcal{D}}; w)},$$

where the penultimate step uses (6) (i.e. $\mathbb{E}_{X_0}[\|X_1 - X_2\|^2 \mid \mathcal{D}, w] \le \kappa^2\mathcal{L}_{\mathrm{FM}}(\theta_{\mathcal{D}}; w)$, as in Lemma B.11) and the final step uses $\mathcal{L}_{\mathrm{FM}} \le \mathcal{L}_{\mathrm{CFM}}$ (Lemma B.1). The constant is finite because X_0 is Gaussian. This proves the claim. □

B.8 Uniform-in-Parameter Control of the CFM Loss

Lemma B.10 (Uniform-in-w Control of the CFM Loss)**.** *Under **A4**, there exists a deterministic constant $C_W > 0$, depending on m_W, r_0, κ_0, the parameter dimension p, the Lipschitz constant L^W_{CFM} from Lemma B.8, and the global loss bound $\bar{L} = 2U^2 + 4\bar{R}^2 + 4n$, such that, for any fixed $\mathcal{D}$,*

$$\sup_{w \in \mathcal{W}} \mathcal{L}_{\mathrm{CFM}}(\theta_{\mathcal{D}}; w) \le C_W\, \mathcal{L}(\theta_{\mathcal{D}})^{\frac{1}{p+1}}.$$

Proof. Let $Y(w) \triangleq \mathcal{L}_{\mathrm{CFM}}(\theta_{\mathcal{D}}; w) \ge 0$, and $M_{\mathcal{D}} \triangleq \sup_{w \in \mathcal{W}} Y(w)$, which is finite since $Y(\cdot)$ is continuous in w and $\mathcal{W}$ is compact. We only consider the case $M_{\mathcal{D}} > 0$; otherwise, the bound trivially holds. If $L^W_{\mathrm{CFM}} = 0$, then Y is constant on $\mathcal{W}$, so $M_{\mathcal{D}} = \mathcal{L}(\theta_{\mathcal{D}})$. Using the uniform loss bound $\mathcal{L}(\theta_{\mathcal{D}}) \le \bar{L}$ established below gives $M_{\mathcal{D}} \le \bar{L}^{p/(p+1)}\mathcal{L}(\theta_{\mathcal{D}})^{1/(p+1)}$. With the zero-loss case already covered, for the remainder assume $L^W_{\mathrm{CFM}} > 0$. Let $\kappa_p \triangleq \pi^{p/2}/\Gamma(p/2 + 1)$. Recall $m_W > 0$ and the (r_0, κ_0)-fatness of $\mathcal{W}$ from **A**4.

Fix $c > 2$ and $\epsilon \in (0, (c-2)M_{\mathcal{D}}/c)$, and choose $w^\star \in \mathcal{W}$ (not necessarily in the interior) such that $M_{\mathcal{D}} < Y(w^\star) + \epsilon$; such $w^\star$ exists since Y is continuous and $\mathcal{W}$ is compact. Then, for any $w \in \mathcal{W}$,

$$\begin{aligned} M_{\mathcal{D}} - Y(w) < Y(w^\star) - Y(w) + \epsilon &\le L^W_{\mathrm{CFM}}\|w^\star - w\| + \epsilon \\ &\Leftrightarrow Y(w) > M_{\mathcal{D}} - L^W_{\mathrm{CFM}}\|w^\star - w\| - \epsilon, \end{aligned}$$

where the second inequality uses the L^W_{CFM}-Lipschitz continuity of Y in w (cf. Lemma B.8). Note that $w^\star$ *need not lie deep in the interior of* $\mathcal{W}$: rather than shrinking a ball $B_r(w^\star)$ until it fits entirely inside $\mathcal{W}$ (which would require a $w^\star$-dependent, and hence data-dependent radius r), we use the fatness of $\mathcal{W}$ to lower-bound the volume of $B_r(w^\star) \cap \mathcal{W}$ directly, for any $w^\star \in \mathcal{W}$.

Let $r_\star \triangleq \min\{r_0, M_{\mathcal{D}}/(cL^W_{\mathrm{CFM}})\}$. For $w \in B_{r_\star}(w^\star) \cap \mathcal{W}$, $Y(w) > M_{\mathcal{D}} - L^W_{\mathrm{CFM}}r_\star - \epsilon$. We consider two cases.

Case 1: $r_\star = M_{\mathcal{D}}/(cL^W_{\mathrm{CFM}})$. Then $Y(w) > \frac{c-1}{c}M_{\mathcal{D}} - \epsilon > \frac{1}{c}M_{\mathcal{D}}$ on $B_{r_\star}(w^\star) \cap \mathcal{W}$. Since $r_\star \le r_0$, (r_0, κ_0)-fatness gives $\mathrm{Vol}(B_{r_\star}(w^\star) \cap \mathcal{W}) \ge \kappa_0\kappa_p r_\star^p$, so, using $p(w) \ge m_W$,

$$\mathcal{L}(\theta_{\mathcal{D}}) \ge \int_{B_{r_\star}(w^\star)\cap\mathcal{W}} Y(w)p(dw) \ge \frac{1}{c}M_{\mathcal{D}}\, m_W\kappa_0\kappa_p r_\star^p = \frac{m_W\kappa_0\kappa_p}{c}\left(\frac{1}{cL^W_{\mathrm{CFM}}}\right)^p M_{\mathcal{D}}^{p+1},$$

hence

$$M_{\mathcal{D}} \le \underbrace{cL^W_{\mathrm{CFM}}\left(\frac{c}{m_W\kappa_0\kappa_p}\right)^{\frac{1}{p+1}}}_{\triangleq\, C_1} \mathcal{L}(\theta_{\mathcal{D}})^{\frac{1}{p+1}}.$$

Case 2: $r_\star = r_0$, i.e., $M_\mathcal{D} > cL^W_{\mathrm{CFM}} r_0$. Then $L^W_{\mathrm{CFM}} r_0 < M_\mathcal{D}/c$, so on $B_{r_0}(w^\star) \cap \mathcal{W}$, $Y(w) > M_\mathcal{D} - M_\mathcal{D}/c - \epsilon > \frac{1}{c} M_\mathcal{D}$ (using $\epsilon < (c-2)M_\mathcal{D}/c$). By (r_0, κ_0)-fatness of $\mathcal{W}$,

$$\mathcal{L}(\theta_\mathcal{D}) \geq \frac{1}{c} M_\mathcal{D}\, m_W \kappa_0 \kappa_p r_0^p, \qquad \text{i.e.,} \qquad M_\mathcal{D} \leq \underbrace{\frac{c}{m_W \kappa_0 \kappa_p r_0^p}}_{\triangleq\, C_2} \mathcal{L}(\theta_\mathcal{D}).$$

Since $u_{t|1}(Z_t; w \mid X_1) = X_1 - X_0$ by our choice and $\|u_t^\theta\| \leq U$ for every $\theta \in \Theta$, the same computation as in the proof of Lemma B.4(b) gives $\mathcal{L}(\theta) \leq 2U^2 + 4\bar{R}^2 + 4n \triangleq \bar{L}$ for every $\theta \in \Theta$ (using $\mathbb{E}\|X_0\|^2 = n$), so $\mathcal{L}(\theta_\mathcal{D})^{p/(p+1)} \leq \bar{L}^{p/(p+1)}$ and

$$M_\mathcal{D} \leq C_2\, \mathcal{L}(\theta_\mathcal{D}) = C_2\, \mathcal{L}(\theta_\mathcal{D})^{\frac{p}{p+1}}\, \mathcal{L}(\theta_\mathcal{D})^{\frac{1}{p+1}} \leq C_2 \bar{L}^{\frac{p}{p+1}}\, \mathcal{L}(\theta_\mathcal{D})^{\frac{1}{p+1}}.$$

Combining both cases with $C_W \triangleq \max\{C_1,\, C_2 \bar{L}^{p/(p+1)}\}$ gives $M_\mathcal{D} \leq C_W\, \mathcal{L}(\theta_\mathcal{D})^{\frac{1}{p+1}}$. This constant depends only on $m_W, r_0, \kappa_0, p, L^W_{\mathrm{CFM}}$ and $\bar{L}$ through C_1, C_2 and the displayed maximum. □

Remark B.3. *Assumption **A4** prevents the maximizer of the CFM loss from lying in an arbitrarily thin region of $\mathcal{W}$ with negligible probability mass. Together with the Lipschitz continuity from Lemma B.8, it turns small population-average loss into uniform-in-w control with a deterministic constant.*

Remark B.4 (Sharper bound in the large-loss region)**.** *The exponent $1/(p+1)$ in Lemma B.10 is needed only to cover Case 1 of the proof. In Case 2[2], namely when $r^\star = r_0$ (equivalently, $M_\mathcal{D} > cL^\mathcal{W}_{\mathrm{CFM}} r_0$), the proof gives the sharper bound $M_\mathcal{D} \leq C_2 \mathcal{L}(\theta_\mathcal{D})$ with $C_2 = c/(m_\mathcal{W} \kappa_0 \kappa_p r_0^p)$. In this regime, the Lipschitz/fatness branches of Theorems 4.2 and 4.3 sharpen accordingly. Indeed, the uniform-in-w estimates used in their proofs give*

$$Z(w) \leq \kappa \sqrt{C_2 \mathcal{L}(\theta_\mathcal{D})}, \qquad Z_g(\mathcal{D}, w) \leq \rho_G^{x,U} \kappa \sqrt{C_2 \mathcal{L}(\theta_\mathcal{D})}.$$

Consequently, on the generalization event $\mathcal{L}(\theta_\mathcal{D}) \leq \mathcal{E}_{N,\delta}$, both quantities are $O\Big(\mathcal{E}_{N,\delta}^{1/2}\Big)$ simultaneously for all w, with probability at least $1-\delta$ over $\mathcal{D}$. Hence, in Case 2, the $\mathcal{E}_{N,\delta/2}^{1/[2(p+1)]} \log(1/\delta)$ terms in the corresponding high-probability bounds can be replaced by $\mathcal{E}_{N,\delta}^{1/2} \log(1/\delta)$.

B.9 Mean-square Flow Coupling

Lemma B.11 (Mean-square Flow Coupling)**.** *Let w be independent of $\mathcal{D}$. Then,*

$$\mathbb{E}_{w,X_0}\left[\left\|\Phi_1(X_0; w) - \hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w)\right\|^2 \mid \mathcal{D}\right] \leq \kappa^2\, \mathbb{E}_w[\mathcal{L}_{\mathrm{FM}}(\theta_\mathcal{D}; w) \mid \mathcal{D}] \leq \kappa^2 \mathcal{L}(\theta_\mathcal{D}).$$

Proof. Taking expectation with respect to w on both sides of the intermediate estimate in the proof of Theorem 4.1, item (a), yields

$$\mathbb{E}_{w,X_0}\left[\left\|\Phi_1(X_0; w) - \hat{\Phi}_1^{\theta_\mathcal{D}}(X_0; w)\right\|^2 \mid \mathcal{D}\right] \leq \kappa^2 \int_0^1 \mathbb{E}_{w,X_s}\left[\left\|u_s(X_s; w) - u_s^{\theta_\mathcal{D}}(X_s; w)\right\|^2 \mid \mathcal{D}\right] ds$$

$$= \kappa^2 \mathbb{E}_w\left[\mathcal{L}_{\mathrm{FM}}(\theta_\mathcal{D}; w) \mid \mathcal{D}\right] \leq \kappa^2 \mathcal{L}(\theta_\mathcal{D}).$$

□

[2]Case 2 corresponds to the regime in which the worst-case CFM loss $M_\mathcal{D}$ is sufficiently large relative to its variation over the fixed fatness radius r_0, i.e., $M_\mathcal{D} > cL^\mathcal{W}_{\mathrm{CFM}} r_0$. In this regime, the loss remains a constant fraction of its maximum throughout an r_0-neighborhood of a maximizer, so the fixed-radius fatness condition can be applied directly without shrinking the neighborhood as $\mathcal{L}(\theta_\mathcal{D})$ decreases.

C Implementation Details

Computational resources. All experiments were conducted on a workstation with an AMD Ryzen Threadripper PRO 7985WX CPU (64 cores/128 threads), 502 GB of RAM, and an NVIDIA RTX 6000 Ada Generation GPU. Model training used GPU acceleration. Timing conventions differ by experiment and are stated where each table is introduced, since the two experiments make different fairness trade-offs: for ACOPF (Table 4), all reported wall-clock timing comparisons are single-instance, unbatched, CPU measurements, matching a deployment setting that answers queries one at a time and avoiding the batch-amortized dispatch-overhead understatement. For the game experiment (Table 9), by contrast, the learned methods (ours, MISO, L2WS, DiOpt) are timed with GPU-batched inference over the full held-out set, reflecting the amortized-batch-inference regime that Table 7's break-even argument is about, while the classical iterative solvers (OGDA, Extragradient, OMD) are inherently CPU-sequential, one instance per call, since this codebase does not vectorize their per-instance iteration across instances; Table 9's caption and Appendix D.2 give the unified accounting used across both regimes, and neither convention claims classical solvers and GPU-batched learned methods are timed on identical hardware.

C.1 Braess Network

Network and cost model. The network has four nodes (origin A, destination D, intermediate B, C) and six links: AB and CD are congestion-sensitive, AC and BD have constant cost, and the classical single shortcut BC is duplicated into two parallel copies BC_1, BC_2 with identical, near-zero, flow-independent cost. Link cost follows the BPR (Bureau of Public Roads, 1964) form $t(f) = t_0\big(1 + \alpha(f/\text{cap})^\beta\big)$ with $\alpha = \beta = 1$; we use $t_0 = 1$, cap $= 100$ for AB, CD, $t_0 = 45$ for AC, BD (effectively infinite capacity), and $t_0 = 0.5$ for BC_1, BC_2 (also effectively infinite capacity). The parameter w is the scalar A-D travel demand, and $x = (x_1, x_2, x_{3a}, x_{3b}) \in \mathbb{R}^4_+$ with $x_1 + x_2 + x_{3a} + x_{3b} = w$ is the path-flow vector over $P_1{=}A{\to}B{\to}D$, $P_2{=}A{\to}C{\to}D$, $P_{3a}{=}A{\to}B{\to}C{\to}D$ (via BC_1), $P_{3b}{=}A{\to}B{\to}C{\to}D$ (via BC_2). The VI operator $F_w(x)$ is the vector of path costs induced by the link costs above.

Closed-form equilibrium. Because $AB{=}CD$ and $BC_1{=}BC_2$ by construction, the four path costs collapse to two free parameters, and solving the Wardrop equilibrium conditions (Wardrop, 1952) gives an exact closed form in three demand regimes, with thresholds d_{low} solving $t(d_{\text{low}}) = 45 - 0.5$ under the shared BPR law (here $d_{\text{low}} = 4350$) and $d_{\text{high}} = 2d_{\text{low}} = 8700$:

$$(x_1, x_2, y) = \begin{cases} (0,\, 0,\, w), & w < d_{\text{low}} \quad \text{(shortcut only)}, \\ (w - d_{\text{low}},\, w - d_{\text{low}},\, d_{\text{high}} - w), & d_{\text{low}} \le w < d_{\text{high}} \quad \text{(all four paths)}, \\ (w/2,\, w/2,\, 0), & w \ge d_{\text{high}} \quad \text{(shortcut abandoned)}, \end{cases}$$

where $y \triangleq x_{3a} + x_{3b}$. Whenever $y > 0$, the individual split (x_{3a}, x_{3b}) is a degenerate direction: BC_1, BC_2 have identical, flow-independent cost, so any split with $x_{3a} + x_{3b} = y$ is equally optimal. This was verified numerically (Wardrop-gap $< 10^{-6}$ for random splits at and around both thresholds) before being used as the labeling function. The dataset draws $w \sim \text{Unif}(200, 14000)$ (covering all three regimes, ≈31/31/38% of rows respectively) and, for each row, one point from the resulting solution set (drawing $x_{3a} \sim \text{Unif}(0, y)$ when $y > 0$).

Baselines. We compare against three published learning-based optimization baselines, re-derived for this VI's path-flow feasible set and Wardrop-gap objective. *L2WS* (Sambharya et al., 2024) learns warm starts for fixed-point optimization algorithms; we refine its deterministic warm start with 10 unrolled projected-extragradient steps. *DiOpt* (Ding et al., 2025) is a diffusion method for constrained optimization; we train it by supervised denoising followed by best-of-8 self-distillation against the true Wardrop gap. *MISO* (Sharony et al., 2024) learns multiple initial solutions for optimization problems; we use a $K{=}8$-headed network with winner-take-all regression and optional pairwise diversity (MISO (wta) and MISO (mix), corresponding to `miso-wta` and `miso-mix`). All methods use the same 12,000-row train / 512-row test split and seeds $\{7, 19, 42, 101, 233\}$. The same three adapted baselines are also used for bimatrix games (Section 5.3, Appendix C.3).

Split-distribution metric. For a held-out w in the degenerate regimes ($y_{\text{true}}(w) > 0$), each method draws $K = 8$ candidates and we compute the normalized split $d = (x_{3a} - x_{3b})/y_{\text{true}}(w)$ across

Table 1: **Braess-network results** (for the $K{=}8$ candidate-set evaluation; the full numbers underlying Fig. 2 use a separate $K{=}1024$ sampling-law evaluation; see Table 2 and Section 5.1). Mean Wardrop gap measures equilibrium accuracy (best-of-8 where applicable; L2WS is a single deterministic point after solver refinement). Diversity ratio (1.0= correct spread) measures the spread along the degenerate split direction, and W_1 measures eight-candidate distribution discrepancy to the target split law (lower is better). We report $\mathrm{mean}_{\pm\mathrm{std}}$ over five seeds, except for the ideal i.i.d. split reference, which reports variation across 200,000 independent sets of eight draws; gap and runtime are not applicable to this split-only reference (—). The teacher and one-step distilled sampler use the same training split and evaluation protocol. Both use best-of-8 for the gap; diversity and W_1 use all eight candidates. Their matched CPU timings measure generation of eight candidates in one batch, excluding gap scoring.

Method	Mean gap	Diversity ratio	$W_1 \downarrow$	Time (ms)
L2WS (Sambharya et al., 2024) (refined)	$0.0027_{\pm 0.0010}$	$0.00_{\pm 0.00}$	$0.50_{\pm 0.00}$	$0.006_{\pm 0.001}$
DiOpt (Ding et al., 2025)	$0.0315_{\pm 0.0130}$	$0.616_{\pm 0.053}$	$0.51_{\pm 0.06}$	$0.012_{\pm 0.004}$
MISO (Sharony et al., 2024) (wta)	$\underline{0.0003_{\pm 0.0000}}$	$\underline{0.94_{\pm 0.03}}$	$\underline{0.10_{\pm 0.01}}$	$\underline{0.0015_{\pm 0.0003}}$
MISO (Sharony et al., 2024) (mix)	$0.0010_{\pm 0.0004}$	$1.35_{\pm 0.04}$	$14.49_{\pm 3.16}$	$0.0016_{\pm 0.0005}$
Ideal i.i.d. $\mathrm{Unif}[-1,1]$	—	$0.918_{\pm 0.179}$	$0.224_{\pm 0.093}$	—
Ours (80-step, Best-of-8)	$0.00188_{\pm 0.00028}$	$0.934_{\pm 0.045}$	$0.242_{\pm 0.037}$	$2.862_{\pm 0.075}$
Ours (1-step, Best-of-8)	$0.00183_{\pm 0.00024}$	$0.931_{\pm 0.043}$	$0.241_{\pm 0.037}$	$0.02616_{\pm 0.00039}$

the K draws. The target label distribution has $d \sim \mathrm{Unif}[-1,1]$ with $\mathrm{Var}(d) = 1/3$. We therefore report the *diversity ratio*, defined as $\mathrm{std}(\mathrm{clip}(d,-1,1))/\sqrt{1/3}$, using the population standard deviation over the K draws (1.0 indicates the correct spread), and the one-dimensional Wasserstein distance $W_1(\frac{1}{K}\sum_{k=1}^{K}\delta_{d_k}, \mathrm{Unif}[-1,1])$, averaged over 8 representative demand values spanning both degenerate regimes. Even exact i.i.d. draws from the target have nonzero empirical W_1: a Monte Carlo reference with 200,000 repetitions gives a mean (std.) of $0.224\,(0.093)$ for $K = 8$. This is an i.i.d. sampling reference, not a lower bound; eight deterministic midpoint quantiles attain $W_1 = 1/(2K) = 0.0625$. Thus, *lower W_1 indicates a closer empirical candidate distribution, but does not by itself establish a better stochastic sampler.* MISO (wta)'s low W_1 is consistent with quantile-like placement of its fixed heads, whereas our teacher and distilled sampler obtain W_1 near the i.i.d. reference while allowing fresh candidates from new noise draws. These finite-K results support distributional prediction and preservation of diversity under distillation; comparison of the induced distributions requires a larger-sample evaluation. MISO (mix)'s large W_1 is caused by over-dispersed split values outside the target interval.

Larger-sample distribution evaluation. To distinguish finite candidate-set quality from the induced sampling law, we draw $K = 1024$ candidates per demand, without best-of-K selection. The flow models use fresh Gaussian noise, and MISO draws head indices independently and uniformly from its eight fixed heads. We average the unclipped split W_1 over three independent sampling replicates and the same eight demands, then report mean (std.) across five training seeds in Table 2. Both flow samplers have lower average discrepancy than the uniformly weighted MISO (wta) distribution, although rankings vary across seeds. Their discrepancy remains above the ideal i.i.d. reference, indicating residual distributional error. This comparison concerns the specified sampling laws, whereas Table 1 evaluates eight-candidate representations.

Table 2: **Braess split-law comparison with** 1024 **draws.** No candidate selection is applied. Entries use $\mathrm{mean}_{\pm\mathrm{std}}$ across five training seeds after averaging demands and sampling replicates.

Sampling law	Diversity ratio	$W_1 \downarrow$
Ours (80-step teacher)	$\underline{0.9676_{\pm 0.0208}}$	$0.0933_{\pm 0.0083}$
Ours (1-step distilled)	$0.9673_{\pm 0.0203}$	$\underline{0.0879_{\pm 0.0113}}$
MISO (wta), uniform head sampling	$0.9328_{\pm 0.0324}$	$0.1047_{\pm 0.0097}$
MISO (mix), uniform head sampling	$1.3548_{\pm 0.0374}$	$14.4387_{\pm 3.0985}$

C.2 ACOPF

We use the OPFLEARN dataset (Joswig-Jones et al., 2022), which provides supervised ACOPF pairs $(w, x_w^\star)$ generated by sampling load profiles and solving the corresponding nonlinear OPF instances. The parameter $w \in \mathbb{R}^{2n_{\text{load}}}$ stacks active/reactive demands at load buses, $w = [P_D, Q_D]$, where n_{load} is the number of load buses. The decision vector follows the dataset packing $x = [P_G,\ Q_G,\ V_m,\ V_a] \in \mathbb{R}^{4|\mathcal{G}|}$, where $|\mathcal{G}|$ is the number of MATPOWER generator rows. We evaluate three MATPOWER cases (Zimmerman et al., 2011): IEEE 30-bus, 57-bus, and 118-bus.

We learn a conditional vector field $u_t^\theta(x_t; w)$ with a 3-layer MLP architecture. We train with ADAMW (Loshchilov & Hutter, 2019) using batch size 256. Learning rates are 10^{-4} (30-bus) and 10^{-3} (57/118-bus). The hidden width is 128 for 30-bus and 256 for 57/118-bus; the time embedding dimension is 32 for all cases. Every 100 epochs, we evaluate on a fixed batch of 64 held-out scenarios. At evaluation time, we sample $x_w \sim \hat{p}^{\hat{\theta}}(\cdot \mid w)$ by integrating $\dot{x} = u_t^{\hat{\theta}}(x_t; w)$ using the Euler sampler with 70 steps (time horizon $[0, 0.99]$), starting from $x_0 \sim \mathcal{N}(0, I)$.

Physics-based evaluation via PANDAPOWER (Thurner et al., 2018). To assess feasibility and cost under the actual network physics, we evaluate each sample with an AC power flow (PF) solve in PANDAPOWER. For each scenario $w = (P_D, Q_D)$, we: (i) set loads (P_D, Q_D), (ii) apply generator controls (P_G, V_m) extracted from x_w, and (iii) run PF. If PF fails to converge, we mark the sample as unsuccessful and assign unit violation; otherwise we compute feasibility and objective from the PF solution.

We attach quadratic generation costs by parsing MATPOWER `gencost` and mapping each MATPOWER generator row to the corresponding PANDAPOWER element (including the slack generator mapped to `ext_grid`). The realized objective is computed from PF outputs:

$$f_w(x) \triangleq \sum_{k \in \mathcal{G}} \left(c_{2,k} P_{G,k}^2 + c_{1,k} P_{G,k} + c_{0,k} \right),$$

where $P_{G,k}$ is the *post-PF* generation (including slack injection). This choice evaluates the operational cost of the actual PF-consistent state induced by the sampled controls.

Constraint violation metric (control vs. network). We report a normalized hinge-violation score computed *after PF* (or set to 1 if PF fails). Let $(\cdot)_+ = \max(\cdot, 0)$. We group constraints into: **Control constraints** (generator limits): active and reactive bounds for non-slack generators and (optionally) slack:

$$v^{P_G} = \left[\left(\tfrac{P^{\min} - P_G}{s_P}\right)_+,\ \left(\tfrac{P_G - P^{\max}}{s_P}\right)_+ \right], \quad s_P \triangleq \max(|P^{\min}|, |P^{\max}|, 1),$$
$$v^{Q_G} = \left[\left(\tfrac{Q^{\min} - Q_G}{s_Q}\right)_+,\ \left(\tfrac{Q_G - Q^{\max}}{s_Q}\right)_+ \right], \quad s_Q \triangleq \max(|Q^{\min}|, |Q^{\max}|, 1),$$

and **network constraints** (system limits): voltage magnitude bounds and thermal limits (lines/transformers if available):

$$v^V = \left[\left(\tfrac{V^{\min} - V}{s_V}\right)_+,\ \left(\tfrac{V - V^{\max}}{s_V}\right)_+ \right], \quad s_V \triangleq \max(V^{\max} - V^{\min}, 10^{-3}),$$
$$v^S = \left[\left(\tfrac{S - S^{\max}}{S^{\max}}\right)_+ \right], \quad S \triangleq \sqrt{P^2 + Q^2}.$$

We summarize each constraint family by the mean of its components and then average across families:

$$\text{vio}_{\text{ctrl}} \triangleq \text{mean}\{\text{mean}(v^{P_G}), \text{mean}(v^{Q_G}), \ldots\}, \quad \text{vio}_{\text{net}} \triangleq \text{mean}\{\text{mean}(v^V), \text{mean}(v^S), \ldots\},$$

and finally $\text{vio}_{\text{total}} = \alpha\, \text{vio}_{\text{ctrl}} + (1 - \alpha)\, \text{vio}_{\text{net}}$ with $\alpha = 0.5$.

Suboptimality. Let x_w denote a generated sample and $x_w^\star$ the OPFLEARN label for the same w. We report the objective suboptimality and relative suboptimality:

$$\text{subopt}(x_w; w) \triangleq \left(f_w(x_w) - f_w(x_w^\star)\right)_+, \quad \text{rel_subopt}(x_w; w) \triangleq \frac{\text{subopt}(x_w; w)}{|f_w(x_w^\star)| + \varepsilon},\ \varepsilon = 10^{-6}.$$

Why is the feasibility violation non-negligible? The following properties of OPFLEARN explain the observed behavior:

1. **Solver-feasible vs. PF-feasible solutions.** OPFLEARN labels are obtained by solving a nonlinear ACOPF using `IPOPT`, which returns solutions that satisfy KKT conditions up to numerical tolerances. Feasibility is therefore defined *within the optimizer*, not via a separate post-solution AC power flow. When these solutions are re-evaluated using an independent PF solver with hinge penalties, small but systematic constraint violations naturally appear.
2. **Soft handling of reactive power limits.** As discussed in (Joswig-Jones et al., 2022), generator reactive power limits are not strictly enforced in all cases, and PV/PQ switching is solver-dependent. As a result, OPFLEARN labels may lie slightly outside Q_G bounds while remaining acceptable to the OPF solver. Our evaluation explicitly measures post-PF Q_G violations, making these discrepancies visible.
3. **Incomplete or soft enforcement of network constraints.** In OPFLEARN, branch thermal limits and voltage constraints are not uniformly enforced across all cases and are sometimes treated as soft or inactive constraints. Consequently, OPFLEARN solutions are not guaranteed to satisfy all engineering limits when re-evaluated under a strict PF-based constraint checker.
4. **Slack-bus absorption of residual imbalance.** The OPFLEARN formulation allows the slack generator to absorb residual mismatches, and explicit bounds on slack injections are typically loose or absent. Our evaluator explicitly accounts for slack behavior, which can further contribute to apparent violations.

Taken together, these factors explain why feasibility plateaus at nonzero levels even for ground-truth labels. The key result is that as the square-root learning error decreases, our model approaches the feasibility profile of the OPFLEARN labels under the same physics-based evaluator while reducing relative suboptimality. Improving feasibility would require either re-generating labels under the PF-based constraint semantics or augmenting flow matching with explicit feasibility mechanisms (e.g., projection, Q-limit enforcement, or constrained flow variants).

DC3 and Fioretto et al. porting methodology. Both baselines are ported from their official released code into our exact evaluation harness (same OPFLEARN data, train/test split, network parsing, and feasibility/suboptimality metrics used for our own method). *DC3* (Donti et al., 2021): we port `utils.py`'s `ACOPFProblem` (equality-constraint completion via Newton-Raphson power flow, implemented as a custom differentiable `torch.autograd.Function` with an implicit-function-theorem backward pass) and `method.py`'s gradient-correction loop, adapted to source network data via our own MATPOWER parser instead of DC3's original PYPOWER case files. Two numerical stability fixes were required for DC3 to train without diverging on the largest network (IEEE 118). *Fioretto et al.* (Fioretto et al., 2020): since the paper's own released code targets a Julia/PowerModels/JuMP stack, we port only the training mechanism (a shared-trunk network predicting (P_G, V_m, V_a), trained with a bound-penalty loss whose per-constraint-type Lagrange multipliers are updated by dual ascent, $\lambda \mathrel{+}= \rho \cdot \mathrm{mean}(\text{violation})$, $\rho = 10^{-4}$, after each epoch), reimplemented directly against our own differentiable violation functions rather than their PowerModels-based ones. Both baselines are trained for 300 epochs at each of the five standard seeds $\{7, 19, 42, 101, 233\}$, matching this paper's retrain-per-seed convention for every learning-based method. Table 3's Time column for both baselines (and for ours) is measured single-instance (batch size 1), unbatched, on CPU, with model weights irrelevant to wall-clock cost, matching the protocol already used for t_{draw} in Table 4 below. Our own method's Violation/Rel. subopt. column in Table 3 is read directly from the final-epoch entry in the same per-seed training-history records that produce Fig. 7, restricted to the standard five seeds $\{7, 19, 42, 101, 233\}$ for consistency with the DC3/Fioretto columns (Fig. 7 itself averages over all eight available seeds).

Classical-solver wall-clock baseline. We time pandapower's built-in ACOPF solver (`pp.runopp`) directly in our own pipeline, solving each of 30 held-out test load profiles from scratch, reusing the exact same network/cost/bound setup already used for PF-based feasibility checking above. Table 4 reports the resulting wall-clock comparison against our trained sampler's per-instance draw time (70 Euler steps, same architecture as training, CPU, single instance, i.e., measured under the same unbatched CPU conditions as the solver baseline).

Table 3: **ACOPF baseline comparison** (full numbers underlying Fig. 4, Section 5.2). Violation and relative suboptimality (single-draw for both flow samplers; DC3 and Fioretto et al. are deterministic one-shot predictors); PF success is the fraction of samples for which a downstream power-flow solve converges. Time is single-instance, unbatched, CPU wall-clock. We report $\mathrm{mean}_{\pm\mathrm{std}}$ over five seeds where available; existing timing entries without recorded standard deviations are shown as single values. Distilled rows use the validated checkpoint-based evaluation underlying Fig. 4.

Case	Method	Violation	Rel. subopt. (%)	PF success	Time (ms)
IEEE 30	Ours	$0.254_{\pm 0.005}$	$1.39_{\pm 0.46}$	$1.00_{\pm 0.00}$	4.03
	Ours (1-step distilled)	$0.260_{\pm 0.002}$	$1.26_{\pm 0.15}$	$1.00_{\pm 0.00}$	$0.0675_{\pm 0.0026}$
	Fioretto (Fioretto et al., 2020)	$0.210_{\pm 0.007}$	$6.49_{\pm 1.04}$	$1.00_{\pm 0.00}$	0.040
	DC3 (Donti et al., 2021)	$0.000_{\pm 0.000}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	131.30
IEEE 57	Ours	$0.274_{\pm 0.004}$	$0.27_{\pm 0.07}$	$1.00_{\pm 0.00}$	4.82
	Ours (1-step distilled)	$0.273_{\pm 0.002}$	$0.58_{\pm 0.04}$	$1.00_{\pm 0.00}$	$0.0707_{\pm 0.0019}$
	Fioretto (Fioretto et al., 2020)	$0.269_{\pm 0.000}$	$1.90_{\pm 0.13}$	$1.00_{\pm 0.00}$	0.037
	DC3 (Donti et al., 2021)	$0.001_{\pm 0.000}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	174.31
IEEE 118	Ours	$0.083_{\pm 0.001}$	$1.33_{\pm 0.28}$	$1.00_{\pm 0.00}$	5.53
	Ours (1-step distilled)	$0.083_{\pm 0.001}$	$1.43_{\pm 0.29}$	$1.00_{\pm 0.00}$	$0.0879_{\pm 0.0005}$
	Fioretto (Fioretto et al., 2020)	$0.050_{\pm 0.001}$	$0.33_{\pm 0.12}$	$1.00_{\pm 0.00}$	0.043
	DC3 (Donti et al., 2021)	$0.00005_{\pm 0.00001}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	533.87

Table 4: **ACOPF classical-solver wall-clock baseline.** t_{solve}: `pp.runopp`, mean over 30 held-out instances. t_{draw}: our trained sampler, 70 Euler steps, single instance, CPU. $t_{\mathrm{draw}}^{\mathrm{distill}}$: one-step distilled sampler, $\mathrm{mean}_{\pm\mathrm{std}}$ over five seeds under the same timing protocol. Speedup and break-even multipliers refer to the original 70-step sampler. $M/N \triangleq t_{\mathrm{solve}}/(t_{\mathrm{solve}} - t_{\mathrm{draw}})$ is the amortization break-even multiplier.

Case	t_{solve} (ms)	t_{draw} (ms)	$t_{\mathrm{draw}}^{\mathrm{distill}}$ (ms)	Per-query speedup	Break-even mult. M/N
IEEE 30	136.6 (12.5)	4.03	$0.0675_{\pm 0.0026}$	34×	1.030
IEEE 57	162.9 (27.1)	4.82	$0.0707_{\pm 0.0019}$	34×	1.031
IEEE 118	239.0 (32.9)	5.53	$0.0879_{\pm 0.0005}$	43×	1.024

Because $t_{\mathrm{draw}} \ll t_{\mathrm{solve}}$ in every case, the online-vs.-label-generation break-even point is reached after only 2–3% more future queries than the number of training instances originally labeled with OPFLEARN, regardless of that number's specific value. Note that this comparison does not include the one-time cost of training the flow model.

C.3 NON-POTENTIAL BIMATRIX GAME TRAINING DATA GENERATION

We consider parametric two-player general-sum bimatrix games with affine payoffs $A(w) = A_0 + \sum_{i=1}^{k} w_i A_i$ and $B(w) = B_0 + \sum_{i=1}^{k} w_i B_i$. Player 1 selects $p \in \Delta_n$ and Player 2 selects $q \in \Delta_n$, where $\Delta_n \triangleq \{x \in \mathbb{R}_+^n : 1^\top x = 1\}$. A Nash equilibrium solves the associated VI on $\Delta_n \times \Delta_n$. In the main text we report Nash gap, best-of-M selection, and equal-time comparisons. The Nash gap of (p, q) is

$$\mathrm{Gap}(p, q; w) = \Big(\max_{i \in [n]} e_i^\top A(w) q - p^\top A(w) q\Big) + \Big(\max_{j \in [n]} p^\top B(w) e_j - p^\top B(w) q\Big),$$

and we define $G_w^{(M)} \triangleq \min_{m \le M} \mathrm{Gap}(\Pi(X_w^{(m)}); w)$ for i.i.d. draws from the learned conditional distribution $\hat{p}^{\hat{\theta}}(\cdot \mid w)$, where Π denotes the projection onto $\Delta_n \times \Delta_n$.

The expected utilities are $u_1(p, q; w) = p^\top A(w) q$ and $u_2(p, q; w) = p^\top B(w) q$. This construction induces a family of games with heterogeneous strategic structure, including non-zero-sum and non-potential interactions.

VI formulation. A Nash equilibrium $(p^\star, q^\star)$ satisfies the first-order optimality conditions

$$p^\star \in \arg\max_{p \in \Delta_n} \; p^\top A(w) q^\star, \qquad q^\star \in \arg\max_{q \in \Delta_n} \; (p^\star)^\top B(w) q.$$

These conditions are equivalent to the variational inequality

$$F_w(x^\star)^\top (x - x^\star) \geq 0, \qquad \forall x \in \mathcal{X},$$

where $x = (p, q)$, $\mathcal{X} = \Delta_n \times \Delta_n$, and

$$F_w(x) \triangleq \begin{bmatrix} -A(w)q \\ -B(w)^\top p \end{bmatrix}.$$

The operator F_w is not necessarily monotone, and the solution set may be non-singleton. Consequently, convergence guarantees for monotone VIs do not apply, and equilibrium selection is inherently ambiguous. Rather than predicting a single equilibrium, we learn a solution distribution. This setting provides a challenging yet representative testbed for validating our theoretical guarantees on expected gap bounds and solution-set approximation in non-monotone PVIs.

Training data generation. For the small and medium cases (n=5, k=7 and n=8, k=12), we generate training equilibrium labels using `nashpy`'s `vertex` enumeration routine, which returns exact Nash equilibria (up to minor numerical tolerance) by enumerating best-response polytopes. For the large game (n=50, k=16), full enumeration is intractable. Instead, we generate labels with a two-step entropic Mirror-Prox solver using step size 0.5, 10,000 iterations, and 16 random restarts; we retain an instance only when the best-of-restarts candidate has exploitability at most 10^{-3}, yielding a keep rate of approximately 90%.

Evaluation metrics. For each test instance specified by w, the learned model induces a conditional distribution $\hat{p}^{\hat{\theta}}(\cdot \mid w)$ over joint strategies x. Since the feasible set is $\mathcal{X} = \Delta_n \times \Delta_n$, we post-process each sampled strategy by projecting both marginals onto the simplex $\Pi(x) \triangleq \big(\Pi_{\Delta_n}(p),\, \Pi_{\Delta_n}(q)\big)$. All reported metrics are computed on $\hat{x}_w \triangleq \Pi(X_w)$ where $X_w \sim \hat{p}^{\hat{\theta}}(\cdot \mid w)$.

(i) Nash gap (exploitability). The Nash gap of (p, q) is the sum of unilateral best-response improvements:

$$\mathrm{Gap}(p, q; w) \triangleq \Big(\max_{p' \in \Delta_n} u_1(p', q; w) - u_1(p, q; w)\Big)_+ + \Big(\max_{q' \in \Delta_n} u_2(p, q'; w) - u_2(p, q; w)\Big)_+,$$

where $(z)_+ = \max\{z, 0\}$. Because utilities are linear in mixed strategies, best responses occur at pure actions, yielding

$$\mathrm{Gap}(p, q; w) = \Big(\max_{i \in [n]} e_i^\top A(w) q - p^\top A(w) q\Big)_+ + \Big(\max_{j \in [n]} p^\top B(w) e_j - p^\top B(w) q\Big)_+.$$

The gap is nonnegative and equals zero if and only if (p, q) is a Nash equilibrium. Moreover, it coincides with the gap concept $G(x; w) = \sup_{x' \in \mathcal{X}} F_w(x)^\top (x - x')$ defined earlier for variational inequalities, so it directly measures stationarity of the associated VI.

Reported gap statistics. For each method, we report the empirical mean, median, and 90th percentile (P90) of Nash gap over the test set. The P90 statistic characterizes tail behavior and robustness to difficult instances, which is particularly informative in non-monotone games.

(ii) Success rate. We report the fraction of test instances satisfying $\mathrm{Gap}(\hat{x}_w; w) \leq 10^{-3}$, denoted $\mathrm{Succ}@10^{-3}$. This provides an interpretable notion of approximate equilibrium attainment and complements mean-based metrics by quantifying how often a method reaches high-accuracy solutions.

(iii) Best-of-M selection. To account for the set-valued nature of the solution map, we evaluate a best-of-M selection rule $\mathrm{Gap}^{(M)}(w) \triangleq \min_{m \in [M]} \mathrm{Gap}\big(\Pi(X_w^{(m)}); w\big)$, with $X_w^{(m)} \overset{\text{i.i.d.}}{\sim} \hat{p}^{\hat{\theta}}(\cdot \mid w)$. This quantity is non-increasing in M and operationalizes equilibrium selection at inference time without modifying the training objective.

(iv) Equal-time comparison. To enable fair runtime comparisons with first-order VI baselines, we additionally report an equal-time variant in which the sampling budget M is chosen so that the average runtime per instance matches that of the baseline method. Runtime is measured in milliseconds per problem instance (ms/w).

Evaluation protocol and equal-time calibration. All methods are evaluated on the same per-seed held-out set and with the same vectorized Nash-gap/regret scorer applied inside the timed region for every method. For the flow-based method, we draw M i.i.d. samples from the learned conditional distribution using a fixed number of sampler steps (`sampler_steps` $= 70$ for single-draw and Best-of-8; the Euclidean-CFM equal-time and one-step-distilled rows instead use a 10-step Heun integrator, see Appendix D.6), project each draw onto $\Delta_n \times \Delta_n$, and compute the Nash gap and player regrets. This yields per-instance gap statistics for both a *single draw* and a *best-of-M* selection $\min_{m \le M} \mathrm{Gap}(\Pi(X_{w_i}^{(m)}); w_i)$.

Runtime is measured directly via wall-clock timing and reported as milliseconds per instance; for the flow-based method we estimate a per-draw cost $\widehat{t}_{\mathrm{draw}}$ and map sampling budgets to time by $t(M) = M\,\widehat{t}_{\mathrm{draw}}$. For first-order baselines (OGDA, Extragradient, and entropic OMD), we run a fixed iteration budget (`baseline_steps` $= 3000$) with a common step size and record both the resulting gaps and the corresponding time per instance. Finally, to enable equal-time comparisons, we choose M_{eq} such that $M_{\mathrm{eq}}\widehat{t}_{\mathrm{draw}}$ matches each baseline's measured time budget (capped by `equal_time_cap_M`), and report the best-of-M_{eq} gap distribution.

What the reported per-draw timing includes. The Time (ms/w) column in Table 9 and the per-draw cost $\widehat{t}_{\mathrm{draw}}$ used for the equal-time row cover, for every method in the table without exception, a 3-call discarded warmup, explicit GPU synchronization around the timed region for any GPU-touching call, sampling/solving, destandardization, simplex projection, *and* Nash-gap/regret scoring, i.e., the full per-candidate cost of best-of-M selection. This uniform accounting is what is meant by "the unified evaluation protocol" referenced throughout this appendix.

The equal-time rows in Table 9 use the same wall-clock calibration idea: for each baseline, we measure its per-instance runtime and choose the largest sampling budget M_{eq} whose measured flow runtime fits that budget (capped by `equal_time_cap_M`).

Separately-tuned baseline step sizes. The comparison above shares one common step size (0.05) across OGDA, EG, and OMD; we check the sensitivity of this choice, and also consider two multistart comparators, since these are the standard fairness checks for a comparison against iterative first-order baselines. A *compute-matched* multistart baseline (running the full 3000-iteration solve M times, $M\times$ the cost of one solve) is not wall-clock comparable to our method, since a single 3000-iteration solve already costs on the order of 2000–2700$\times$ more than a single flow draw (Table 7); the wall-clock-fair comparators are instead (i) the equal-time protocol already used throughout this appendix, which we apply with tuned step sizes below, and (ii) a *budget-split* multistart baseline that divides the same 3000-iteration total budget across R independent restarts, reported in Appendix D.3.

For step-size tuning, we grid-search step $\in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$ independently for each solver and each game family, selecting the value minimizing mean Nash gap over 32 held-out validation instances (distinct from the 512 test instances used for reporting), at the same 3000-iteration budget used throughout (so wall-clock cost per instance is unaffected by this choice – step size does not change the per-iteration computational cost). The selected step sizes are:

Table 5: **Tuned step sizes** vs. a shared step-size baseline.

Case	OGDA	Extragradient	OMD
n=5, k=7	0.2	0.5	0.2
n=8, k=12	0.5	0.5	0.1
n=50, k=16	0.5	0.5	0.5
Original (all cases)		0.05 (shared)	

Table 6 reports the single-init Nash-gap statistics before and after tuning (mean over 5 seeds, 512 test instances, identical protocol otherwise), underlying the updated OGDA/EG/OMD rows in Table 9. Tuning improves every solver in every game family, sometimes substantially (e.g., n=5, k=7 EG: mean gap $0.374 \to 0.066$, Succ@10^{-3} $0.62 \to 0.91$). We verified this is not an artifact of the grid search or the gap computation: at the tuned step, solvers converge to genuinely different, essentially exact equilibria across instances (spot-checked to 10^{-10} precision, with different pure/mixed strategy profiles per instance), not a degenerate fixed point.

Table 6: **Before/after step-size tuning**, single-init, 3000 iterations, mean over 5 seeds (512 test instances). "Before" uses the shared step 0.05; "after" uses the tuned per-solver, per-family step size from Table 5. Both columns use the same data (Appendix D.5 for n=50, k=16's label-generation protocol), so this table isolates the effect of tuning alone.

Case	Solver	Mean gap (before)	Mean gap (after)	Succ@10^{-3} (before)	Succ@10^{-3} (after)
n=5, k=7	OGDA	0.372	0.122	0.62	0.82
	EG	0.374	0.066	0.62	0.91
	OMD	0.153	0.057	0.67	0.87
n=8, k=12	OGDA	0.717	0.433	0.53	0.77
	EG	0.717	0.192	0.53	0.87
	OMD	0.353	0.209	0.58	0.72
n=50, k=16	OGDA	3.084	1.738	0.19	0.51
	EG	3.082	1.367	0.19	0.61
	OMD	2.325	1.418	0.27	0.56

Interpretation caveat across problem scales. This budgeted protocol makes the curves directly comparable across solvers with different per-iteration costs, and it explains why the flow curves are naturally expressed as functions of M while the baselines are functions of T. However, the resulting success-rate dynamics still depend on how labels are constructed: in the small/medium regimes (exact enumeration labels) all methods eventually hit a numerical/approximation floor and saturate, whereas in the large regime (OMD-generated and filtered labels) increasing-budget baselines can exhibit an "accelerating" rise in success because additional iterations unlock a growing fraction of instances that cross the 10^{-3} threshold under similar first-order dynamics.

Online-vs.-label-generation break-even (concrete numbers). The offline cost of solving N instances from scratch is recovered by online sampling after roughly $N \cdot t_{\text{solve}}/(t_{\text{solve}} - t_{\text{draw}})$ future queries, where t_{solve} is the cost of solving one instance from scratch and t_{draw} is our trained sampler's per-draw cost. Table 7 instantiates this using the same run underlying Table 9's OGDA row (for t_{solve}) and single-draw row (for t_{draw}), for all three game families. Because $t_{\text{draw}} \ll t_{\text{solve}}$ in every case ($t_{\text{draw}}/t_{\text{solve}} \approx 0.03$–$0.05\%$), the break-even multiplier $t_{\text{solve}}/(t_{\text{solve}} - t_{\text{draw}})$ is only ≈ 1.0004 throughout: the break-even point is reached after only about 0.04% more future queries than the number N of training instances originally labeled, regardless of the specific value of N actually used.

Table 7: **Amortization break-even**, bimatrix games. Mean over 5 seeds; t_{solve} from OGDA (3000 iterations, same run underlying Table 9), t_{draw} from our single-draw flow sampler.

Case	t_{solve} (ms)	t_{draw} (ms)	Per-query speedup	Break-even mult. M/N
n=5, k=7	54.88	0.0202	2719×	1.00037
n=8, k=12	54.71	0.0206	2652×	1.00038
n=50, k=16	62.25	0.0299	2080×	1.00048

An equivalent measurement for ACOPF is given in Appendix C.2 (Table 4): timing a classical nonlinear AC-OPF solver directly in our own pipeline gives a smaller but still favorable break-even multiplier ($\approx$ 1.02–1.03, vs. $\approx$ 1.0004 for bimatrix games), since our sampler itself takes a few milliseconds per instance there rather than a small fraction of a millisecond. The bimatrix-game and ACOPF numbers together should be read as evidence that the break-even argument is realistic across the two experimental settings where we have measured it.

What this measurement does not include. Both break-even calculations above measure only the flow method's online sampling cost against a per-instance classical-solve cost; neither includes the one-time cost of *training* the flow model itself (as opposed to collecting the labels it trains on), which is a separate additive term not captured by either side of the $t_{\text{solve}}/t_{\text{draw}}$ ratio. A full accounting including training time is left to future work.

D ADDITIONAL NUMERICAL EXPERIMENT RESULTS

This appendix collects focused diagnostics and supporting results: the ACOPF scaling regression underlying Section 5.2 and compact bimatrix-game results and baseline methodology underlying Section 5.3.

D.1 ACOPF SCALING DIAGNOSTICS

The ACOPF scaling diagnostic tests the predicted $O(\sqrt{\mathcal{L}})$ scaling by fitting $\log(\text{metric}) = \text{slope} \cdot \log(\sqrt{\mathcal{L}}) + c$ with ordinary least squares; the theory predicts slope $= 1$. We fit each of the eight seeds separately across its checkpoints, then report the mean slope and a t-based 95% confidence interval across seeds, so correlated checkpoints within one training run do not count as independent replicates. Table 8 reports this slope, the mean per-seed coefficient of determination R^2 (the fraction of variance in $\log(\text{metric})$ explained by each seed's fitted log–log line), and the corresponding uncertainty.

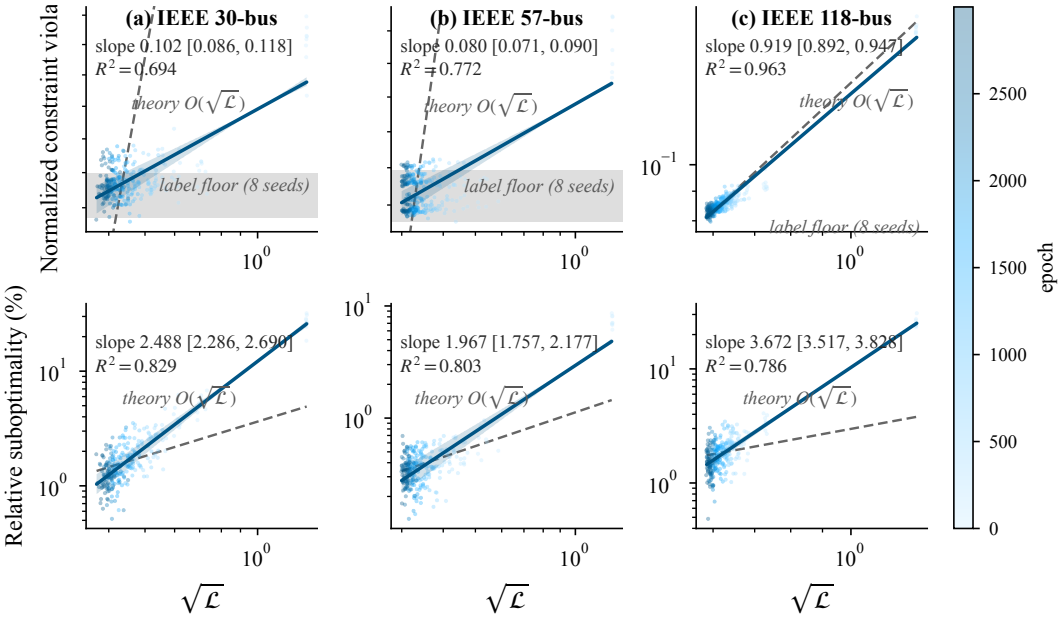

Figure 7: **Full ACOPF scaling diagnostics.** Each point is one seed/checkpoint pair pooled over eight seeds (n=248/panel), plotted against $\sqrt{\mathcal{L}}$; columns are IEEE 30/57/118-bus, rows are normalized constraint violation and relative suboptimality (%). The solid line is the pooled descriptive OLS fit; the shaded band uses the seed-level slope confidence limits; the dashed line marks the theory-predicted slope 1, and the gray band in the top row marks the OPFLEARN label-level violation floor.

For the IEEE 118-bus system, the fitted violation slope (0.919, seed-level 95% CI $[0.892, 0.947]$, mean per-seed R^2=0.963) is close to the theoretical prediction of 1, and this match is robust to the training window used for the fit (slope $\in [0.92, 1.11]$ whether restricted to only the first 100 epochs or computed over the full 3000). For the IEEE 30- and 57-bus systems, the fitted violation slope is far from 1 ($\approx$ 0.08–0.15) at every training window we tried. Inspecting the raw per-checkpoint trajectories explains why: *for these two smaller networks, constraint violation saturates near its* OPFLEARN *label-imperfection floor* (Appendix C.2) within the first 100 of the 3000 training epochs, while the training loss (and hence $\sqrt{\mathcal{L}}$) keeps decreasing for the remainder of training. The regression is consequently dominated by a near-flat post-floor plateau rather than the responsive pre-floor regime, and the fitted slope there reflects the flatness of that floor rather than the predicted rate. They do not contradict the theoretical upper bound in Theorem 4.2, but unlike the 118-bus case, they should not be interpreted as evidence for slope one. The relative-suboptimality slope is consistently *larger* than 1 across all three networks (2–3.7): suboptimality decreases faster than the theoretical worst case, consistent with an upper bound.

Table 8: **ACOPF scaling regression.** Per-seed fitted slope of $\log(\text{metric})$ on $\log(\sqrt{\mathcal{L}})$, averaged over eight seeds with a t-based 95% CI; R^2 is the mean across per-seed fits.

Metric	Statistic	IEEE 30	IEEE 57	IEEE 118
Total violation	Slope (95% CI)	0.102 [0.086, 0.118]	0.080 [0.071, 0.090]	0.919 [0.892, 0.947]
Total violation	R^2	0.694	0.772	0.963
Rel. subopt.	Slope (95% CI)	2.488 [2.286, 2.690]	1.967 [1.757, 2.177]	3.672 [3.517, 3.828]
Rel. subopt.	R^2	0.829	0.803	0.786

D.2 Full Bimatrix-Game Results Table and Consistency-vs-Exactness Summary

Table 9 gives the full Nash-gap statistics (mean, median, P90, success rate at 10^{-3}, and per-instance runtime) underlying Figure 5 and Section 5's Results discussion: our method (single-draw, Best-of-8, and equal-time-to-OMD), the three learning-based baselines (MISO, L2WS, DiOpt), and the three tuned classical VI solvers (OGDA, Extragradient, OMD), across all three game families.

D.3 Short-Restart Multistart Comparison for OGDA, Extragradient, and OMD

Table 9's classical-solver rows use one long, single-initialization run of 3000 iterations. A natural fairness question is whether the same total iteration budget is better spent as R independent, shorter restarts (each of $\lfloor 3000/R \rfloor$ iterations, from an independently-seeded random initialization), taking the best (minimum) Nash gap across the R restarts per instance, mirroring our own method's best-of-M selection at a matched *total* computational budget. This is a distinct question from the equal-time comparison above (which asks how many Flow draws fit in OMD's time budget); here we ask whether OMD/EG/OGDA's *own* budget is better spent as one long run or several short ones.

Table 10 reports mean Nash gap and success@10^{-3} across $R \in \{1, 2, 4, 8, 16, 32\}$ restarts, mean over 5 seeds, on the same 512-instance held-out set and tuned step sizes as Table 9. Per-instance wall-clock time is essentially flat across R (varying by at most a few percent, since the total iteration budget is fixed), so this table isolates the effect of restart count alone.

Mean gap and success@10^{-3} disagree on the best R, and the direction of the disagreement is family-dependent, which is consistent with the general caution that a method should not be described as universally superior when these two metrics conflict. For the easy family (n=5, k=7), more restarts monotonically *improves* mean gap for every solver but *degrades* success@10^{-3} monotonically: splitting the budget helps the typical instance but leaves fewer iterations for hard instances to cross the strict 10^{-3} threshold. For the hard family (n=50, k=16), the two metrics instead *agree*: both mean gap and success@10^{-3} improve with more restarts up to $R \approx$ 8–32. The medium family (n=8, k=12) is intermediate: success@10^{-3} peaks at a moderate $R \in \{2, 4\}$ then declines, while mean gap keeps improving through R=32. No single R is therefore adopted as a replacement for Table 9's R=1 convention.

Does early stopping change the conclusion? For the single-restart (R=1) run of each classical solver, we additionally checkpoint the Nash gap every 100 iterations and locate the first checkpoint $t^\star$ at which the gap changes by less than 1% relative to the previous checkpoint (or $t^\star$=3000 if no such point exists). For (n=5, k=7), no solver triggers this criterion before 3000 iterations: the easy family keeps improving throughout the full budget, so early stopping offers no time savings there. For (n=50, k=16), all three solvers trigger early ($t^\star \approx$ 900–1200, a 60–67% reduction in iteration count) but at a real accuracy cost: for OGDA, mean gap at $t^\star$ is higher than at t=3000 (e.g., 1.93 vs. 1.82) and success@10^{-3} is correspondingly lower (0.430 vs. 0.508). Early stopping therefore does not change the qualitative conclusion above: it offers a real wall-clock/accuracy trade-off for the hard family and essentially no lever for the easy one.

Comparisons in Figure 6. Table 11 gives the paired welfare statistics for all methods. The dashed curves in Figure 6 report the admissible rate (fraction of held-out instances with at least one candidate of Nash gap at most 10^{-2}) for all five methods, computed before restricting to paired welfare comparisons. For the welfare overlay, we retain a learning-based baseline only when its intersection with OMD contains at least 50 instances. This yields paired subsets of size 304 and 97 for MISO

Table 9: **Non-potential bimatrix games.** Nash-gap statistics, success rate (fraction of instances with Nash gap $\leq 10^{-3}$), and per-instance runtime across three game families, comparing our method against tuned first-order VI solvers (OGDA/EG/OMD) and three learning-based baselines (MISO, L2WS, DiOpt). All Time entries in this table (single-draw/Best-of-8/†/MISO/L2WS/DiOpt: GPU, batched over the full 512-instance held-out set; OGDA/EG/OMD: CPU, sequential, one instance per call, per Appendix C's computational-resources paragraph) are measured under one unified protocol: a 3-call discarded warmup, explicit GPU synchronization bracketing every GPU-timed call, and a single vectorized Nash-gap/regret scorer applied *inside* the timed region for every method, so that no row's reported time omits scoring or benefits from an unsynchronized GPU clock. Underlined entries are the best value in each column within a game family. †: one-step endpoint distillation of the corresponding teacher; mean$_{\pm\text{std}}$ over five evaluation/sample seeds with one trained student per family; these rows and the equal-time row additionally use a 10-step Heun integrator (single-draw/Best-of-8 use 70-step Euler; see Appendix D.6). The equal-time row's candidate budget M_{eq} was calibrated against an OMD wall-clock measurement taken in an earlier measurement session than this table's own OGDA/EG/OMD rows. Full evaluation protocol, baseline methodology, and the (n=50, k=16) label-generation protocol are given in Appendices C.3, D.4, and D.5.

Case	Method	Mean gap	Med gap	P90 gap	Succ@1e-3	Time (ms/w)
n=5, k=7	Ours (single-draw)	$0.453_{\pm 0.04}$	$0.328_{\pm 0.02}$	$1.041_{\pm 0.09}$	$0.050_{\pm 0.00}$	$0.020_{\pm 0.00}$
	Ours (Best-of-8)	$0.092_{\pm 0.00}$	$0.041_{\pm 0.00}$	$0.263_{\pm 0.02}$	$0.240_{\pm 0.02}$	$0.162_{\pm 0.00}$
	Ours (1-step, Best-of-8)†	$0.241_{\pm 0.01}$	$0.170_{\pm 0.02}$	$0.592_{\pm 0.03}$	$0.230_{\pm 0.01}$	$0.008_{\pm 0.00}$
	Ours (equal-time to OMD)	$0.008_{\pm 0.00}$	$0.000_{\pm 0.00}$	$0.025_{\pm 0.00}$	$0.690_{\pm 0.01}$	$48.760_{\pm 0.00}$
	MISO-MLP (Sharony et al., 2024) (Best-of-8)	$0.232_{\pm 0.02}$	$0.188_{\pm 0.02}$	$0.503_{\pm 0.03}$	$0.110_{\pm 0.03}$	$0.006_{\pm 0.00}$
	L2WS (Sambharya et al., 2024) (refined)	$0.553_{\pm 0.01}$	$0.473_{\pm 0.01}$	$1.119_{\pm 0.05}$	$0.020_{\pm 0.00}$	$0.005_{\pm 0.00}$
	DiOpt (Ding et al., 2025)	$2.729_{\pm 0.04}$	$2.546_{\pm 0.03}$	$4.435_{\pm 0.13}$	$0.000_{\pm 0.00}$	$0.005_{\pm 0.00}$
	OGDA (Daskalakis & Panageas, 2018) (tuned step)	$0.122_{\pm 0.02}$	$0.000_{\pm 0.00}$	$0.138_{\pm 0.05}$	$0.820_{\pm 0.01}$	$54.880_{\pm 0.15}$
	EG (Korpelevich, 1976) (tuned step)	$0.066_{\pm 0.01}$	$0.000_{\pm 0.00}$	$0.000_{\pm 0.00}$	$0.910_{\pm 0.01}$	$96.570_{\pm 0.16}$
	OMD (Mertikopoulos et al., 2019) (tuned step)	$0.057_{\pm 0.02}$	$0.000_{\pm 0.00}$	$0.007_{\pm 0.01}$	$0.870_{\pm 0.01}$	$55.160_{\pm 0.39}$
n=8, k=12	Ours (single-draw)	$1.552_{\pm 0.03}$	$1.405_{\pm 0.06}$	$2.988_{\pm 0.06}$	$0.030_{\pm 0.01}$	$0.021_{\pm 0.00}$
	Ours (Best-of-8)	$0.421_{\pm 0.01}$	$0.322_{\pm 0.02}$	$0.983_{\pm 0.02}$	$0.180_{\pm 0.01}$	$0.165_{\pm 0.00}$
	Ours (1-step, Best-of-8)†	$0.916_{\pm 0.03}$	$0.914_{\pm 0.04}$	$1.663_{\pm 0.04}$	$0.090_{\pm 0.02}$	$0.010_{\pm 0.00}$
	Ours (equal-time to OMD)	$0.030_{\pm 0.00}$	$0.000_{\pm 0.00}$	$0.104_{\pm 0.01}$	$0.660_{\pm 0.02}$	$47.770_{\pm 0.00}$
	MISO-MLP (Sharony et al., 2024) (Best-of-8)	$0.709_{\pm 0.03}$	$0.701_{\pm 0.04}$	$1.224_{\pm 0.04}$	$0.040_{\pm 0.01}$	$0.008_{\pm 0.00}$
	L2WS (Sambharya et al., 2024) (refined)	$1.398_{\pm 0.02}$	$1.357_{\pm 0.03}$	$1.983_{\pm 0.05}$	$0.000_{\pm 0.00}$	$0.005_{\pm 0.00}$
	DiOpt (Ding et al., 2025)	$3.965_{\pm 0.06}$	$3.764_{\pm 0.07}$	$5.951_{\pm 0.08}$	$0.000_{\pm 0.00}$	$0.005_{\pm 0.00}$
	OGDA (Daskalakis & Panageas, 2018) (tuned step)	$0.433_{\pm 0.05}$	$0.000_{\pm 0.00}$	$1.789_{\pm 0.26}$	$0.770_{\pm 0.01}$	$54.710_{\pm 0.11}$
	EG (Korpelevich, 1976) (tuned step)	$0.192_{\pm 0.04}$	$0.000_{\pm 0.00}$	$0.174_{\pm 0.18}$	$0.870_{\pm 0.01}$	$96.580_{\pm 0.27}$
	OMD (Mertikopoulos et al., 2019) (tuned step)	$0.209_{\pm 0.03}$	$0.000_{\pm 0.00}$	$0.479_{\pm 0.10}$	$0.720_{\pm 0.04}$	$55.430_{\pm 0.25}$
n=50, k=16	Ours (single-draw)	$7.583_{\pm 0.08}$	$7.128_{\pm 0.05}$	$12.184_{\pm 0.28}$	$0.000_{\pm 0.00}$	$0.030_{\pm 0.00}$
	Ours (Best-of-8)	$3.588_{\pm 0.08}$	$3.690_{\pm 0.08}$	$5.522_{\pm 0.10}$	$0.010_{\pm 0.01}$	$0.239_{\pm 0.00}$
	Ours (1-step, Best-of-8)†	$3.124_{\pm 0.03}$	$3.147_{\pm 0.05}$	$4.379_{\pm 0.06}$	$0.020_{\pm 0.01}$	$0.083_{\pm 0.00}$
	Ours (equal-time to OMD)	$0.505_{\pm 0.03}$	$0.358_{\pm 0.05}$	$1.293_{\pm 0.06}$	$0.300_{\pm 0.01}$	$54.200_{\pm 0.00}$
	MISO-MLP (Sharony et al., 2024) (Best-of-8)	$2.490_{\pm 0.03}$	$2.561_{\pm 0.01}$	$3.388_{\pm 0.03}$	$0.020_{\pm 0.00}$	$0.082_{\pm 0.00}$
	L2WS (Sambharya et al., 2024) (refined)	$5.609_{\pm 0.18}$	$5.641_{\pm 0.18}$	$8.768_{\pm 0.24}$	$0.010_{\pm 0.01}$	$0.014_{\pm 0.00}$
	DiOpt (Ding et al., 2025)	$5.091_{\pm 0.04}$	$4.966_{\pm 0.05}$	$6.636_{\pm 0.08}$	$0.000_{\pm 0.00}$	$0.018_{\pm 0.00}$
	OGDA (Daskalakis & Panageas, 2018) (tuned step)	$1.738_{\pm 0.12}$	$0.003_{\pm 0.01}$	$5.763_{\pm 0.38}$	$0.510_{\pm 0.01}$	$62.250_{\pm 0.15}$
	EG (Korpelevich, 1976) (tuned step)	$1.367_{\pm 0.06}$	$0.000_{\pm 0.00}$	$5.242_{\pm 0.39}$	$0.610_{\pm 0.02}$	$111.290_{\pm 0.22}$
	OMD (Mertikopoulos et al., 2019) (tuned step)	$1.418_{\pm 0.06}$	$0.000_{\pm 0.00}$	$4.969_{\pm 0.10}$	$0.560_{\pm 0.02}$	$63.630_{\pm 0.29}$

on (n=5, k=7) and (n=8, k=12), respectively, 83 for DiOpt on (n=5, k=7), and 70 for L2WS on (n=5, k=7). The remaining baseline–family intersections are smaller and are omitted from the welfare overlay.

D.4 Learning-Based Baseline Methodology: MISO, L2WS, and DiOpt

We compare against three recent learning-based baselines for the bimatrix-game experiment, chosen to span three distinct strategies for learning to solve a family of VI/equilibrium instances: MISO (Sharony et al., 2024) (multi-hypothesis one-shot regression), L2WS (Sambharya et al., 2024) (a learned warm start for a classical fixed-point solver), and DiOpt (Ding et al., 2025) (diffusion-based generation refined by self-distillation against the true objective). All three have official released code, which we adapt to our setting as described below. None targets a static one-shot VI/equilibrium-prediction task directly, so each requires porting the released loss or training procedure onto our problem structure while keeping the underlying mechanism unchanged.

Table 10: **Short-restart multistart sweep.** Mean Nash gap and success@10^{-3} (mean over 5 seeds) for R independent restarts of $\lfloor 3000/R \rfloor$ iterations each, best-of-R per instance, at the fixed total iteration budget 3000. The R=1 entries are an independently rerun single-start reference and therefore are not expected to reproduce the primary Table 9 values exactly (for example, Table 9 reports EG/OMD mean gaps $0.066/0.057$ for n=5, k=7 and $0.192/0.209$ for n=8, k=12). Per-instance time is flat across R (within a few percent) for every solver/family and is omitted.

Case	Solver	R=1	R=2	R=4	R=8	R=16	R=32
n=5, k=7	OGDA	0.124/0.80	0.096/0.79	0.054/0.79	0.035/0.76	0.032/0.72	0.034/0.70
	EG	0.079/0.91	0.035/0.91	0.023/0.91	0.015/0.89	0.011/0.87	0.011/0.82
	OMD	0.049/0.87	0.031/0.85	0.025/0.82	0.021/0.76	0.022/0.70	0.026/0.65
n=8, k=12	OGDA	0.404/0.78	0.250/0.81	0.158/0.81	0.109/0.81	0.077/0.79	0.057/0.77
	EG	0.199/0.87	0.108/0.88	0.068/0.88	0.044/0.87	0.031/0.85	0.022/0.82
	OMD	0.229/0.73	0.134/0.70	0.100/0.66	0.080/0.64	0.071/0.62	0.065/0.52
n=50, k=16	OGDA	1.816/0.49	0.918/0.59	0.473/0.64	0.285/0.67	0.169/0.67	0.127/0.68
	EG	1.387/0.60	0.684/0.68	0.369/0.71	0.176/0.72	0.116/0.70	0.112/0.69
	OMD	1.540/0.54	0.797/0.62	0.414/0.67	0.248/0.70	0.152/0.69	0.114/0.65

Table 11: **Paired welfare selection against OMD.** For each method, Δ is selected welfare minus OMD on the same instances where both methods produce a candidate with Nash gap at most 10^{-2}. CIs are 95% hierarchical-bootstrap intervals (seeds, then instances). Baseline rows with fewer than 50 paired instances are descriptive only and are not plotted in Figure 6. Oracle regret is relative to the maximum-welfare equilibrium found by complete support enumeration.

Method	Family	Joint n (rate)	Mean Δ [95% CI]	Median Δ	Method wins
Ours	n=5, k=7	1872 (73.1%)	0.409 [0.350, 0.472]	$< 10^{-6}$	45.8%
Ours	n=8, k=12	1530 (59.8%)	0.926 [0.780, 1.092]	$< 10^{-6}$	43.3%
Ours	n=50, k=16	549 (21.4%)	1.411 [1.164, 1.681]	$< 10^{-6}$	42.3%
MISO	n=5, k=7	304 (11.9%)	0.146 [−0.033, 0.332]	$< 10^{-6}$	15.5%
MISO	n=8, k=12	97 (3.8%)	0.956 [0.446, 1.588]	$< 10^{-6}$	28.9%
MISO	n=50, k=16	33 (1.3%)	2.847 [1.299, 4.576]	$< 10^{-6}$	57.6%
L2WS	n=5, k=7	70 (2.7%)	0.002 [−0.013, 0.034]	−0.001	1.4%
L2WS	n=50, k=16	18 (0.7%)	0.243 [−0.569, 1.607]	−0.000001	16.7%
DiOpt	n=5, k=7	83 (3.2%)	0.135 [−0.252, 0.538]	$< 10^{-6}$	27.7%

Equilibrium-oracle regret on joint successes

Family	Oracle regret (Ours/OMD)
n=5, k=7	0.027/0.436
n=8, k=12	0.062/0.988
n=50, k=16	—

MISO. We add a comparison against MISO, a multi-hypothesis predictor that outputs K diverse candidate solutions from a single forward pass, using its official released code (`github.com/EladSharony/miso`). MISO's released implementation targets control-trajectory prediction with an environment-specific dynamics rollout (cartpole/reacher/nuplan tasks), which has no analogue in our static VI setting (predict an equilibrium $(p^\star, q^\star)$ given payoff-matrix parameters w, with no trajectory or dynamics). We therefore port MISO's released loss function (`training/loss.py` and `envs/illustrative_example.py`'s `Loss` class: winner-take-all regression plus a soft-clipped pairwise-distance diversity term, selectable as `miso-wta`/`miso-pd`/`miso-mix`, with pairwise-diversity weights taken verbatim from the released code's own defaults) with the trajectory/time axis and the dynamics-rollout state-loss term removed, since our task is one-shot regression rather than control; the K-headed shared-trunk prediction architecture is otherwise unchanged.

Same-scale protocol. We train MISO on the same data pipeline used for our other learning-based baselines: the same three game-family datasets (the n=50, k=16 label-generation protocol is given in Appendix D.5), the same 5 seeds ($\{7, 19, 42, 101, 233\}$), the same training-set size (12,000 rows) and held-out evaluation size (512 instances), and the same hidden-width rule ($(128, 128)$ or $(256, 128)$ depending on the action-space dimension). We train for 120 epochs, matching the recipe used for the other learning-based baselines. We use $K = 8$ candidates to match this paper's existing Best-of-8 convention for Flow, not anything intrinsic to MISO (its own illustrative example uses $K = 2$). We

report all three loss variants for transparency, with `miso-mix`, the method's own recommended combined objective, designated the primary comparison row in Table 9 *before* inspecting results, consistent with every other single-configuration baseline in this paper.

Table 12 gives the full three-variant comparison. `miso-mix` and `miso-wta` perform similarly. `miso-mix` is a narrow winner in 2 of 3 families and is the row reported in the main text. Because all $K = 8$ candidates come from a single forward pass, MISO's per-instance time is dominated by one MLP evaluation rather than 8 sequential passes.

Table 12: **MISO loss-variant ablation.** Best-of-8, mean (std.) over five seeds on held-out VI instances. `miso-mix` is the pre-registered primary row reported in Table 9. n=50, k=16's label-generation protocol is given in Appendix D.5.

Case	Variant	Mean gap	Med gap	P90 gap	Succ@1e-3	Time (ms/w)
n=5, k=7	miso-wta	0.239 (0.016)	0.201 (0.024)	0.503 (0.014)	0.09 (0.01)	0.006 (0.000)
	miso-pd	0.786 (0.011)	0.711 (0.004)	1.356 (0.037)	0.01 (0.00)	0.006 (0.000)
	miso-mix	0.232 (0.017)	0.188 (0.019)	0.503 (0.032)	0.10 (0.03)	0.006 (0.000)
n=8, k=12	miso-wta	0.725 (0.028)	0.721 (0.039)	1.236 (0.048)	0.04 (0.01)	0.008 (0.000)
	miso-pd	1.466 (0.008)	1.410 (0.018)	2.079 (0.036)	0.00 (0.00)	0.008 (0.000)
	miso-mix	0.709 (0.030)	0.701 (0.037)	1.224 (0.041)	0.04 (0.01)	0.008 (0.000)
n=50, k=16	miso-wta	2.490 (0.025)	2.567 (0.029)	3.351 (0.031)	0.02 (0.01)	0.082 (0.000)
	miso-pd	2.817 (0.035)	2.771 (0.032)	3.536 (0.046)	0.00 (0.00)	0.082 (0.000)
	miso-mix	2.490 (0.027)	2.561 (0.011)	3.388 (0.033)	0.02 (0.00)	0.082 (0.000)

L2WS. Learning to Warm-Start (Sambharya et al., 2024) trains a network mapping problem parameters to a warm start for a fixed-point operator, then applies a fixed number of that operator's iterations, with a loss on the distance between the resulting iterate and the ground-truth solution (their supervised, `fixed_k` loss – one of the two loss families the paper proposes; we use this one since our setting always has a labeled solution, unlike their unsupervised residual-only variant). The official released code (`github.com/stellatogrp/l2ws`) targets Douglas-Rachford/SCS splitting for quadratic and conic programs, for which the bimatrix-game family has no off-the-shelf analogue; it is, however, naturally posed as a VI solved by an alternating mirror-map fixed-point iteration, so we port L2WS's loss construction and training procedure – a warm-start network followed by K unrolled, differentiable steps of a fixed-point operator, trained end-to-end – onto our own already-tuned entropic-OMD update (`solve_omd_entropic`) as the operator. The warm-start network is a 4-layer, 1024-hidden-unit ReLU MLP producing a simplex point per player via a differentiable simplex projection; we unroll K=10 entropic-OMD steps at training time (matching the operator's own already-tuned step size for each game family) and train for 150 epochs on the same 12,000-row training set used for the other baselines. At evaluation we report both the raw warm start (`L2WS (raw)`) and the warm start refined by 10 further OMD steps (`L2WS (refined)`, the row shown in Table 9).

DiOpt. DiOpt (Ding et al., 2025) trains a conditional diffusion policy, then improves it via iterative self-distillation: draw several candidate solutions from the current model for each training condition, evaluate each with the true objective, keep the best, and continue training the denoiser on that self-generated best candidate rather than only the original label. The released code (`github.com/wadx2019/diopt`, MIT licensed) is built around an RL-style policy/critic loop (states, actions, a learned Q-function, a replay buffer) for continuous-control and retargeting tasks; we port the core mechanism – supervised denoising pretraining followed by iterative best-of-K self-distillation – onto our one-shot regression setting, selecting the best candidate by the exact Nash-gap objective (`exploitability_bimatrix`) rather than a learned critic, since it is available in closed form here. The denoiser is the same conditional-DDPM architecture used elsewhere in this project (16 diffusion timesteps); we pretrain for 200 epochs with standard denoising, then run 50 epochs of self-distillation with K=8 candidates per condition, on the same 12,000-row training set as the other baselines.

All three baselines are trained and evaluated on the same 5 seeds, training-set size, and held-out evaluation protocol as MISO (§D.4 above), including on n=50, k=16 (label-generation protocol in Appendix D.5). Results are given in Table 9 and Section 5.

D.5 Large-Game Label-Generation Protocol

This section explains why this family's per-sample accuracy behaves differently under ODE integration than the other two families, and gives the protocol used to generate its labels.

Why does per-sample accuracy degrade under ODE integration? Sampling the flow model's ODE trajectory at intermediate integration times $t \in [0, 1]$ for n=50, k=16 shows that the mean Nash gap of the *partial* sample *increases* monotonically from t=0 (pure noise) to t=1 (the final, fully-integrated sample) – the opposite of the trend observed for n=5, k=7 and n=8, k=12. This is not a discretization artifact: 500 and 2000 Euler steps (instead of the default 70) give the same final gap. The true equilibria in this family are overwhelmingly near-pure-strategy (median mixed-strategy support size 1 of 50 actions), making this large-game setting substantially more difficult for the present learned flow. We therefore report the behavior as an empirical limitation of the current model and label-generation pipeline.

Label generation. Training and test labels are generated with a two-step entropic Mirror-Prox solver, step size 0.5 (matching the already-tuned OGDA/EG/OMD step for this family), 10,000 iterations, and 16 restarts, keeping only instances where the best-of-restarts result had exploitability ≤ 0.001 (keep rate $\approx 90\%$). We generate a 30,000-row dataset from the underlying game family (base payoff matrices A_0, B_0, A_i, B_i) using 120-way CPU parallelism. The resulting labels are predominantly near-pure-strategy (median support size 1), consistent with the explanation above.

D.6 ODE Sampler-Step Ablation Protocol

The flow sampler uses an explicit Euler discretization of the learned ODE. To separate model quality from numerical integration error, we evaluate each trained flow model with sampler steps in $\{20, 40, 70, 140\}$ while holding the number of terminal samples fixed at $M = 8$. Table 13 reports mean (std.) over five random evaluation/sample seeds. Increasing the step count beyond the default 70 gives little additional gap improvement relative to its runtime cost; the main gains in Table 9 therefore come from sampling and selection, not from an unusually fine ODE discretization. **Timing caveat.** This ablation was run on CPU, whereas the timings in Table 9 were measured on GPU; consequently the absolute Time (ms/w) figures here (e.g., 2.037ms at 70 steps for n=5, k=7) are not directly comparable to the GPU timing for the corresponding Best-of-8 row in Table 9 (0.162ms). Only the *within-table* trend across step counts is meaningful here.

Faster equal-time sampler (Heun's method). The Euclidean-CFM "equal-time to OMD" rows in Table 9 use a 10-step Heun (2nd-order Runge–Kutta) integrator in place of 70-step Euler, plus a fully vectorized (batched, no per-instance Python loop) exploitability scorer, both drop-in replacements requiring no retraining – the same learned velocity field defines the ODE regardless of integrator. The 70-step Euler single-draw means reported in Table 9 are 0.453, 1.552, and 7.583 for n=5, k=7, n=8, k=12, and n=50, k=16, respectively. The Heun rows use a larger equal-time candidate budget, so their reported means (0.008, 0.030, and 0.505) are best-of-M_{eq} quantities rather than single-draw comparisons. This distinction is why we do not report an unpaired single-draw Heun-versus-Euler improvement here. Because a correctly-timed per-draw cost must include the cost of scoring each candidate (not only sampling it), we verified that the original per-instance Python scoring loop – not the ODE sampler – was in fact the dominant cost at this batch size; only after vectorizing it does the cheaper sampler translate into more draws fitting in OMD's wall-clock budget. Both changes together increase M at the fixed OMD-matched budget by roughly an order of magnitude ($M \approx 600 \rightarrow 5700$ for n=5, k=7 and n=8, k=12; $\approx 600 \rightarrow 3200$ for n=50, k=16, whose 100-dimensional joint-strategy vector makes each draw costlier), which is what produces the improved equal-time statistics reported in Table 9 relative to a 70-step-Euler-with-unvectorized-scoring implementation of the same protocol.

E Additional Related Work and Discussion

Shared landscape. Learning-based optimization amortizes repeated solution by predicting solutions or update rules (Amos, 2023), while recent diffusion- and flow-based methods learn conditional distributions that can represent multiple solutions (Janner et al., 2022; Li et al., 2024; Liang & Chen,

Table 13: **Euler ODE-step ablation.** Best-of-8 flow samples on 256 held-out test parameters; mean (std.) over five random evaluation/sample seeds. Timed on CPU; not directly comparable to the GPU timings in Table 9 (see text).

Case	Steps	Mean gap	P90 gap	Time (ms/w)
n=5, k=7	20	0.092 (0.006)	0.259 (0.020)	0.884 (0.081)
	40	0.086 (0.007)	0.253 (0.030)	1.262 (0.093)
	70	0.084 (0.007)	0.245 (0.013)	2.037 (0.064)
	140	0.080 (0.006)	0.241 (0.015)	3.843 (0.199)
n=8, k=12	20	0.476 (0.033)	1.098 (0.074)	0.798 (0.092)
	40	0.453 (0.017)	1.073 (0.057)	1.376 (0.045)
	70	0.455 (0.026)	1.038 (0.039)	2.081 (0.069)
	140	0.444 (0.018)	1.045 (0.057)	3.851 (0.038)
n=50, k=16	20	3.621 (0.053)	5.674 (0.107)	1.068 (0.049)
	40	3.650 (0.075)	5.738 (0.109)	1.551 (0.082)
	70	3.732 (0.058)	5.880 (0.035)	2.383 (0.076)
	140	3.735 (0.121)	5.880 (0.126)	4.236 (0.052)

2024). Within this literature, Liang & Chen (2024) is closest to our work: they use flow matching to learn parameter-conditioned distributions over solutions of nonconvex optimization problems, then draw and select candidates at inference time. We share this generative paradigm, and therefore our contribution is not flow matching itself. The additional issues arise from extending it to general PVIs and from analyzing a model learned from finite data.

Beyond scalar-objective optimization. Liang & Chen (2024) study parameterized constrained optimization, where a scalar objective orders candidates and objective suboptimality defines solution quality. *A general VI operator need not be the gradient of any scalar potential* (Facchinei & Pang, 2003). Consequently, objective and suboptimality need not exist. This distinction includes equilibrium models such as our general-sum bimatrix games, whose players need not share a common potential. It also means that a suboptimality result cannot simply be relabeled as a VI guarantee: correctness must instead be formulated directly through feasibility and a VI gap or stationarity measure.

VI-specific correctness. A PVI does not canonically define a probability distribution over its solutions. Our offline labeling process chooses a conditional target $p(\cdot \mid w)$, and the theory assumes that it is supported on the solution set $\mathcal{S}(w)$. Theorems 4.1–4.3 first transfer flow-matching error to Wasserstein distance between this chosen target and the generated distribution, and then use PVI regularity to control distance to the feasible set and the absolute VI gap. These latter transfers replace objective-suboptimality analysis and are specific to the PVI formulation. At nonzero learning error, the conclusions are approximate feasibility and stationarity, rather than exact solution recovery.

From population error to a learned finite-data model. Liang & Chen (2024)'s main guarantee propagates a population vector-field approximation term, together with discretization and target-mass terms, to a best-of-m objective guarantee. It does not provide a finite-N bound connecting the empirical training of the learned network to that population term. This connection matters because our estimator $\theta_{\mathcal{D}}$ depends on the random sample used for training. Under the boundedness, Lipschitz, and covering-complexity conditions in Section 4, uniform generalization controls its population CFM risk. The resulting total error $\mathcal{E}_{N,\delta}$ (cf. Theorem 4.1) combines the best population CFM risk in the model class, including any irreducible CFM floor, the empirical optimization error, and a finite-sample generalization term. Flow stability then converts this quantity into Wasserstein error, and PVI regularity converts Wasserstein error into feasibility and stationarity guarantees for unseen parameters, with high probability over both the training data and the test parameter. Thus, the new statistical bridge is from empirical training to the population error required by the PVI analysis.