CONSTRAINED NONCONVEX STOCHASTIC OPTIMIZATION WITH ONE PROJECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Constrained nonconvex optimization has seen increasing application in modern machine learning, such as safe LLM alignment/finetuning, transfer learning and rank-constrained continual learning. The commonly used approach is Projected SGD which requires projection at every iteration. However, projection onto a functional constraint can cost substantially more than a stochastic first-order update. We study whether this operation can be deferred until the end of nonconvex stochastic optimization, and only do it once. For weakly convex, possibly nonsmooth objectives and regular convex constraints, we give a penalized proximal method that uses $\widetilde{O}(\epsilon^{-4})$ stochastic subgradients and one projection onto the constraint set. The returned point is exactly feasible and has a small Moreau-envelope stationarity measure; for smooth objectives, the same algorithm controls the projected-gradient mapping. For smooth nonconvex constraints, we assume a global lower bound on the infeasible constraint slope which permits arbitrary, possibly infeasible initialization. An exact-penalty variant attains the same stochastic-oracle order and returns an exactly feasible point within $O(\epsilon)$ of an ϵ-KKT point. All intermediate updates use only projections onto a simple Euclidean ball. These are oracle-complexity guarantees: the cost of the single terminal projection is separate, and no efficient projection algorithm for a general nonconvex set is assumed to follow from regularity.

1 INTRODUCTION

We consider stochastic optimization with a functional constraint,

$$\min_{x\in\mathbb{R}^d} f(x) \qquad \text{subject to} \qquad g(x) \le 0, \qquad \mathcal{C} := \{x : g(x) \le 0\}. \tag{1}$$

The objective f may be nonconvex and nonsmooth, and the constraint function g may be convex or nonconvex. In modern machine learning, this problem formulation is widely used in many application scenarios, such as transfer learning (Deng & Kpotufe, 2025; Kalan et al., 2026b), constrained LLM alignment (Zhang et al., 2025; Peng et al., 2025) and rank-constrained LLM finetuning (Hu et al., 2022; Zhang et al., 2023; Jang et al., 2024; Park et al., 2025). Nowadays, as the data amount and the number of parameters to optimize increase dramatically, a computationally efficient algorithm is needed to deploy large-scale model training.

Standard constrained stochastic optimization relies on projecting onto the feasible set after each update step. This method can work well for the simple constraint set such as l_2 unit ball or simplex, since the projection onto them can be computed efficiently. However, this approach becomes a severe bottleneck under general functional constraints as projection is itself an expensive optimization problem and repeatedly solving that sub-problem can dominate the total computational cost. This motivates a fundamental question:

Can we retain a nonconvex stationarity rate while projecting onto the constraint set only once, at the end?

For convex objectives and constraints, one-projection methods already show that the strict feasibility need not be maintained during optimization (Mahdavi et al., 2012; Yang et al., 2017). Extending this principle to nonconvex objectives requires a different conclusion and a different argument. A

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.

small objective gap is no longer the appropriate target, averaging outer iterates need not preserve stationarity, and projecting an arbitrary nearly stationary point of a penalty objective need not preserve the desired certificate. With nonconvex constraints, there is a further obstruction: the constraint violation can have stationary points outside the feasible set.

To bridge this gap, we propose a framework that separates stochastic progress from the final feasibility correction. Our algorithm approximately solves a sequence of strongly convex penalized proximal subproblems over a simple bounding ball, allowing intermediate iterates to remain infeasible. After selecting one outer iterate at random, we apply a single exact projection onto $\mathcal{C}$. The analysis identifies conditions under which this final correction yields an exactly feasible output with a stationarity guarantee for the original constrained problem.

Our results. We introduce the Penalized Proximal Subgradient Method with One Projection and analyze it for weakly convex, possibly nonsmooth objectives under two constraint settings. Under the assumptions specified for each setting, our stationarity guarantees hold with probability at least $2/3$ and require $\widetilde{O}(\epsilon^{-4})$ stochastic subgradient calls, for fixed problem parameters.

- *Convex constraints.* Starting from a supplied feasible point, the algorithm approximately solves softplus-penalized proximal subproblems and returns an exactly feasible output. For smooth objectives, Theorem 3.3 guarantees an ϵ-stationary point measured by the projected-gradient mapping. For nonsmooth objectives, Theorem 3.5 bounds the Moreau-envelope residual by ϵ. In both cases, the oracle complexity matches, up to logarithmic factors, the standard ϵ^{-4} stochastic-oracle complexity for weakly convex optimization (Davis & Grimmer, 2019; Davis & Drusvyatskiy, 2019).
- *Nonconvex constraints.* For smooth nonconvex g, we use an exact hinge penalty and impose a global infeasible-slope condition to rule out stationary points of the constraint violation outside the feasible set. This condition is inspired by boundary gradient lower bounds in convex constrained optimization (Mahdavi et al., 2012; Yang et al., 2017), but is stronger: it controls the slope throughout the ball's infeasible region and accounts for directions blocked by the ball's boundary. Under this condition and the remaining regularity assumptions, the algorithm allows any initial point in the ball and enforces no intermediate feasibility. Theorem 3.8 shows that one final projection returns an exactly feasible output within $O(\epsilon)$ of an ϵ-KKT point of (1), with the same $\widetilde{O}(\epsilon^{-4})$ stochastic subgradient complexity.

2 Related Work

Reducing projections. Mahdavi et al. (2012) introduced stochastic methods with a single final projection for convex optimization. Yang et al. (2017) developed a broader penalty-based theory of reduced projections and improved rates under constraint regularity. Our convex-constraint analysis uses this penalty principle inside proximal subproblems, but the outer target is stationarity of a nonconvex objective. Conditional-gradient methods provide a different way to avoid projections by using a linear minimization oracle (Jaggi, 2013). That oracle can itself be costly for a functional constraint; we instead use values and gradients of g during the iterations.

Weak convexity and constrained stationarity. Proximally guided stochastic subgradient methods solve weakly convex problems through strongly convex subproblems (Davis & Grimmer, 2019). Moreau-envelope stationarity and the ϵ^{-4} stochastic-oracle order are established tools in this setting (Davis & Drusvyatskiy, 2019). Our outer–inner organization builds on these ideas. For nonconvex feasible sets, Davis et al. (2025) analyze stochastic model-based methods over proximally smooth sets, including set approximations followed by retractions that restore feasibility each iteration. Their geometric framework is broader than our single smooth inequality. Our nonconvex result instead permits infeasible iterates throughout, at the price of a global slope condition and one terminal projection oracle.

Functional constraints and exact penalties. Ma et al. (2020) and Boob et al. (2019) regularize both objectives and constraints and solve convex constrained subproblems. The IQRC method of Ma et al. (2020) assumes an approximately feasible initial point and allows approximate feasibility;

it is not a method defined by one final exact projection. Its fully stochastic constraint-oracle model also differs from our exact access to g and ∇g. Jia & Grimmer (2022) study deterministic weakly convex functional constraints, including feasible approximate stationary solutions and Fritz–John guarantees without constraint qualification. More directly, Liu & Xu (2025) connect exact-penalty stationarity to constrained KKT conditions under a uniform Slater-type constraint qualification and give a stochastic method for expectation constraints. We do not claim a new exact-penalty principle. We use a related slope argument, accounting also for the bounding set's normal cone, to justify a single terminal feasibility correction and its effect on the reported output.

3 Main Results

Notations. All norms are Euclidean. Write $[u]_+ := \max\{u, 0\}$ and let $\mathcal{P}_D(x) := \arg\min_{y \in D} \|y - x\|$ denote the Euclidean projection, possibly set-valued. The symbols ∂f and $N_D = \partial \delta_D$ denote the limiting subdifferential and normal cone; δ_D is the extended real-valued indicator of D. A function is ℓ-weakly convex if adding $\ell \|\cdot\|^2/2$ makes it convex. In particular, an ℓ-smooth function is ℓ-weakly convex.

3.1 Penalized Proximal Subgradient Method with One Projection

Our method reduces the original nonconvex problem to a sequence of *strongly convex subproblems over a simple ball*. At outer iteration t, we form the penalized proximal objective

$$H_t(z) := f(z) + p(g(z)) + \frac{\beta}{2}\|z - x_t\|^2, \qquad \min_{z \in \mathcal{B}} H_t(z), \tag{2}$$

where $\mathcal{B} = \{z : \|z\| \le 1\}$ and p penalizes constraint violation. The quadratic term is chosen to dominate the negative curvature of both f and $p \circ g$, so that the *entire objective* H_t is m-strongly convex on $\mathcal{B}$, with $m > 0$. For convex g, the softplus penalty is convex, and $\beta = 2\ell$ gives $m = \ell$. For nonconvex g, we use the hinge penalty $p(u) = \lambda[u]_+$ and choose $\beta > \ell + \lambda L_g$, giving $m = \beta - \ell - \lambda L_g$. The following subsections specify the penalty parameters and assumptions for each setting.

Each subproblem is solved approximately by stochastic subgradient steps projected onto $\mathcal{B}$. Their average becomes the next outer iterate x_{t+1}. Strong convexity lets us choose a fixed inner budget K to obtain a prescribed function error

$$H_t(x_{t+1}) - \min_{z \in \mathcal{B}} H_t(z) \le \epsilon_{\text{in}} \tag{3}$$

with the probability required by the analysis. The algorithm does not need to evaluate the minimum or test this gap during its execution.

Both loops operate in $\mathcal{B}$ without enforcing feasibility in $\mathcal{C}$. After the outer loop, we select an iterate at random and project it onto $\mathcal{C}$ once. For convex constraints, the analysis bounds the infeasibility of each outer iterate. For nonconvex constraints, it instead shows that a selected iterate with a small proximal residual has a nearby feasible, approximately stationary witness. Thus intermediate iterates may be infeasible. Algorithm 1 describes the pseudocode.

For softplus, the penalty subgradient is simply $\nabla(p_\gamma \circ g)(z_s)$; for hinge, use the selection in Section 3.3. Together with (4), this makes h_s an unbiased stochastic subgradient of H, even when f is nonsmooth. The inner steps project only onto $\mathcal{B}$; the sole projection onto $\mathcal{C}$ appears in the outer algorithm's final return step.

3.2 Nonconvex Objective with a Convex Constraint Set

Assumption 3.1 (Shared assumptions for convex constraints)**.**

- The safeguard set is $\mathcal{B} = \{x : \|x\| \le 1\}$. The feasible set $\mathcal{C}$ is nonempty, closed, and contained in $\mathcal{B}$. A point $x_0 \in \mathcal{C}$ is supplied.
- The function f is ℓ-weakly convex and G_f-Lipschitz on an open convex neighborhood of $\mathcal{B}$, where $\ell, G_f > 0$. With fresh randomness at each call, the oracle satisfies

$$\mathbb{E}[\widetilde{v}(x;\xi) \mid \mathcal{F}] = v(x) \in \partial f(x), \qquad \|\widetilde{v}(x;\xi)\| \le G_f \quad \text{a.s.} \tag{4}$$

Algorithm 1 Penalized Proximal Subgradient Method with One Projection

Require: $x_0 \in \mathcal{B}$, $T, K \geq 2$, penalty p, index set $\mathcal{I}$, and $\beta, m > 0$ such that H_t is m-strongly convex on $\mathcal{B}$.

for $t = 0, \ldots, T-1$ **do**
  Form $H_t(z) = f(z) + p(g(z)) + \frac{\beta}{2}\|z - x_t\|^2$.
  $x_{t+1} \leftarrow \mathrm{PSM}(H_t, x_t)$ # approximately minimize H_t over $\mathcal{B}$.
end for
Sample J uniformly from the index set $\mathcal{I}$.
Return $\hat{x} \in \mathcal{P}_{\mathcal{C}}(x_J)$.
Subroutine $\mathrm{PSM}(H, x)$:
  Initialize $z_1 \leftarrow x$.
for $s = 1, \ldots, K$ **do**
  Set $\eta_s \leftarrow 2/[m(s+1)]$ and query $\widetilde{v}(z_s; \xi_s)$ with fresh ξ_s.
  Choose $h_s \in \widetilde{v}(z_s; \xi_s) + \beta(z_s - x) + \partial(p \circ g)(z_s)$.
  $z_{s+1} \leftarrow \mathcal{P}_{\mathcal{B}}(z_s - \eta_s h_s)$.
end for
  Return $K^{-1} \sum_{s=1}^{K} z_s$.

Here x is measurable with respect to the past history $\mathcal{F}$.

- The function g is convex and continuously differentiable on that neighborhood. For constants $G_g, \sigma > 0$,
$$\|\nabla g(x)\| \leq G_g \quad (x \in \mathcal{B}), \qquad \|\nabla g(z)\| \geq \sigma \quad (z \in \mathcal{B},\ g(z) = 0).$$
Both $g(x)$ and $\nabla g(x)$ can be evaluated exactly.

The unit radius is a normalization; a known bounding ball can be translated and rescaled. The boundary condition implies a global error bound on $\mathcal{B}$, $\mathrm{dist}(x, \mathcal{C}) \leq [g(x)]_+/\sigma$, proved in Lemma B.1. It is a property of the chosen defining function, not merely of the feasible set.

Algorithm. At outer iteration t, define

$$F_t(x) := f(x) + \ell\|x - x_t\|^2, \qquad p_\gamma(u) := \gamma \log\bigl(1 + \exp(\lambda_0 u/\gamma)\bigr). \tag{5}$$

With $p = p_\gamma$ and $\beta = 2\ell$, the subproblem in (2) is $H_t = F_t + p_\gamma \circ g$. Since F_t is ℓ-strongly convex and $p_\gamma \circ g$ is convex, H_t is ℓ-strongly convex on $\mathcal{B}$. Set

$$\lambda_0 := \frac{2(G_f + 4\ell)}{\sigma}, \qquad \gamma := \frac{1}{K}. \tag{6}$$

Use Algorithm 1 with $p = p_\gamma$, $\beta = 2\ell$, $m = \ell$, and $\mathcal{I} = \{1, \ldots, T\}$.

Set $\Delta_f := f(x_0) - \min_{x \in \mathcal{B}} f(x)$ throughout this subsection. We state the two objective regimes separately because their stationarity certificates have different meanings.

Smooth objectives.

Assumption 3.2 (Smooth objective)**.** The function f is differentiable with an ℓ-Lipschitz gradient on the same open convex neighborhood of $\mathcal{B}$.

We use the projected-gradient mapping norm

$$\mathcal{G}_\ell(x) := \ell\|x - \mathcal{P}_{\mathcal{C}}(x - \nabla f(x)/\ell)\|. \tag{7}$$

At a feasible point this measures the displacement of a projected gradient step. It vanishes exactly when $0 \in \nabla f(x) + N_{\mathcal{C}}(x)$, the constrained first-order condition. A small value therefore bounds the change made by a projected gradient step from the returned point itself.

Theorem 3.3 (Smooth objective with convex constraints). *Suppose Assumptions 3.1 and 3.2 hold. For any $0 < \epsilon \leq 1$, run Algorithm 1 with $p = p_\gamma$, $\beta = 2\ell$, $m = \ell$, the penalty parameters in* (6)*, and $\mathcal{I} = \{1, \ldots, T\}$. There are choices*

$$T = O\left(1 + \frac{\ell\Delta_f}{\epsilon^2}\right), \qquad K = \widetilde{O}(\epsilon^{-2}), \tag{8}$$

specified in Appendix B.3, for which the output satisfies

$$\hat{x} \in \mathcal{C}, \qquad \Pr\{\mathcal{G}_\ell(\hat{x}) \le \epsilon\} \ge \frac{2}{3}.$$

For fixed problem parameters, the algorithm uses $TK = \widetilde{O}(\epsilon^{-4})$ stochastic gradient calls and exactly one projection onto $\mathcal{C}$.

Remark 3.4. The $\widetilde{O}(\epsilon^{-4})$ rate matches the optimal unconstrained smooth nonconvex exponent under the oracle models of Arjevani et al. (2023); Jin (2026), while using one final projection. This is a comparison of complexity exponents; a matching lower bound under all assumptions and oracle access of Theorem 3.3 is not established here.

Nonsmooth objectives. Under Assumption 3.1, f may be nonsmooth, so the gradient mapping need not be defined. Instead, we follow the standard convergence measure in nonsmooth nonconvex optimization Davis & Drusvyatskiy (2019) and define the Moreau envelope and its residual by

$$\begin{aligned} &\Phi_\ell(x) := \min_{y\in\mathcal{C}}\{f(y) + \ell\,\|y-x\|^2\}, \qquad Q_\ell(x) := \arg\min_{y\in\mathcal{C}}\{f(y) + \ell\,\|y-x\|^2\}, \\ &\mathcal{S}_\ell(x) := \|\nabla\Phi_\ell(x)\| = 2\ell\,\|x - Q_\ell(x)\|\,. \end{aligned} \tag{9}$$

The proximal point $y = Q_\ell(x)$ is unique. Its optimality condition gives

$$\|x-y\| = \frac{\mathcal{S}_\ell(x)}{2\ell}, \qquad \operatorname{dist}(0, \partial f(y) + N_{\mathcal{C}}(y)) \le \mathcal{S}_\ell(x). \tag{10}$$

Thus a small residual means that x is close to a feasible point with a small first-order residual. This nearby-point interpretation is useful for nonsmooth functions, whose subgradients need not vary continuously. It does not assert a small subgradient residual at the returned point itself.

Theorem 3.5 (**Nonsmooth objective with convex constraints**). *Suppose Assumption 3.1 holds. For any $0 < \epsilon \le 1$, run Algorithm 1 with $p = p_\gamma$, $\beta = 2\ell$, $m = \ell$, the penalty parameters in* (6)*, and $\mathcal{I} = \{1, \dots, T\}$. There are choices*

$$T = O\left(1 + \frac{\ell\Delta_f}{\epsilon^2}\right), \qquad K = \widetilde{O}(\epsilon^{-2}), \tag{11}$$

specified in Appendix B.3, for which the output satisfies

$$\hat{x} \in \mathcal{C}, \qquad \Pr\{\mathcal{S}_\ell(\hat{x}) \le \epsilon\} \ge \frac{2}{3}.$$

For fixed problem parameters, the algorithm uses $TK = \widetilde{O}(\epsilon^{-4})$ stochastic subgradient calls and exactly one projection onto $\mathcal{C}$.

Remark 3.6. The rate matches the standard $O(\epsilon^{-4})$ Moreau-envelope stationarity bound for weakly convex stochastic optimization (Davis & Grimmer, 2019; Davis & Drusvyatskiy, 2019), up to logarithms, with one final projection. In the smooth unconstrained case, a small Moreau residual also bounds the gradient norm up to a constant, connecting this measure to the smooth lower-bound benchmark. The same oracle-model qualification applies.

3.3 Nonconvex Objective with a Nonconvex Constraint Set

We now allow g to be nonconvex and x_0 to be infeasible. The following condition excludes infeasible stationary points of the constraint violation, including points where the boundary of $\mathcal{B}$ blocks further progress.

Assumption 3.7 (Nonconvex constraint and infeasible initialization)**.**

- The second item of Assumption 3.1 holds for the objective and its oracle. The set $\mathcal{C}$ is nonempty, closed, and contained in $\operatorname{int}\mathcal{B}$. The initial point $x_0 \in \mathcal{B}$ is arbitrary.
- The function g is continuously differentiable with an L_g-Lipschitz gradient on an open convex neighborhood of $\mathcal{B}$, where $L_g > 0$. Its values and gradients are available exactly. Write $G_g := \max_{x\in\mathcal{B}} \|\nabla g(x)\|$.

- There are $\sigma_{\mathrm{bd}}, \sigma_{\mathrm{out}} > 0$ such that

$$\|\nabla g(z)\| \geq \sigma_{\mathrm{bd}} \qquad (z \in \mathcal{B},\ g(z) = 0), \tag{12}$$

$$\operatorname{dist}(0, \nabla g(x) + N_{\mathcal{B}}(x)) \geq \sigma_{\mathrm{out}} \qquad (x \in \mathcal{B},\ g(x) > 0). \tag{13}$$

Condition (13) is a global sufficient condition for this guarantee, not a consequence of boundary regularity. It says that, within $\mathcal{B}$, as long as the constraint is not satisfied, one can always find a direction to decrease the constraint violation. It is stronger than simply requiring $\nabla g \neq 0$ on $g = 0$. Appendix C.5 gives a smooth counterexample to removing it from the present argument. Meanwhile, (12) and smoothness imply that $\mathcal{C}$ has a unique projection in the tube of radius $R := \sigma_{\mathrm{bd}}/L_g$; no separate geometric assumption is needed (Lemma C.1).

Algorithm: the same subroutine with an exact penalty. Consider $\Psi_\lambda(x) := f(x) + \lambda[g(x)]_+$ on $\mathcal{B}$ and choose

$$\lambda > \frac{G_f}{\sigma_{\mathrm{out}}}, \qquad \beta > \ell + \lambda L_g, \qquad m := \beta - \ell - \lambda L_g. \tag{14}$$

Run Algorithm 1 with $p(u) = \lambda[u]_+$, these β, m, and

$$\mathcal{I} = \{0, \dots, T-1\}. \tag{15}$$

For the composite penalty, select $\lambda \nabla g(z)$ when $g(z) > 0$ and 0 when $g(z) \leq 0$. At $g(z) = 0$, zero is a valid subgradient. Here $p \circ g$ is λL_g-weakly convex, so the inner objective $H_t(z) = \Psi_\lambda(z) + \frac{\beta}{2}\|z - x_t\|^2$ is m-strongly convex over the ball. The parameter choices are sequential and compatible: choose λ first and then β. Neither the inner nor the outer iterates are required to be feasible, and there is no feasibility-restoration phase before the final line of the algorithm.

Convergence measure. Our goal is an exactly feasible output $\hat{x}$ that is close to a near-stationary point of the *original* constrained problem:

$$\exists y \in \mathcal{C}: \quad \|\hat{x} - y\| \leq \delta, \qquad \operatorname{dist}(0, \partial f(y) + N_{\mathcal{C}}(y)) \leq \epsilon. \tag{16}$$

The two tolerances distinguish proximity from first-order accuracy. This is the same nearby-point interpretation underlying (10); it does not assert a small nonsmooth KKT residual at $\hat{x}$ itself. To prove it, we track the auxiliary penalty residual

$$Q_{\lambda,\beta}(x) := \operatorname*{arg\,min}_{y \in \mathcal{B}} \left\{ \Psi_\lambda(y) + \frac{\beta}{2}\|y - x\|^2 \right\}, \qquad \mathcal{R}_{\lambda,\beta}(x) := \beta \|x - Q_{\lambda,\beta}(x)\|. \tag{17}$$

The key is not merely to make a penalized problem stationary. Once this residual is below the positive margin $h_{\mathrm{feas}} := \lambda\sigma_{\mathrm{out}} - G_f$, its proximal witness $y = Q_{\lambda,\beta}(x)$ must be feasible. Its optimality condition then certifies original-problem stationarity, and projecting x changes its distance to that witness by at most a factor of two.

Theorem 3.8 (Nonsmooth objective with nonconvex constraints). *Suppose Assumption 3.7 holds and use* (14)–(15)*. Let* $\Delta_\lambda := \Psi_\lambda(x_0) - \min_{x \in \mathcal{B}} \Psi_\lambda(x)$*. For* $0 < \epsilon < \min\{h_{\mathrm{feas}}, \beta R, 1\}$*, there are choices*

$$T = O\left(1 + \frac{\beta \Delta_\lambda}{\epsilon^2}\right), \qquad K = \widetilde{O}(\epsilon^{-2}), \tag{18}$$

specified in Appendix C.4, with the following guarantee. The output is exactly feasible. With probability at least $2/3$*, the final projection is unique,* $\mathcal{R}_{\lambda,\beta}(x_J) \leq \epsilon$*, and there exist* $y \in \mathcal{C}$*,* $v \in \partial f(y)$*, and* $\mu \in [0, \lambda]$ *such that*

$$\|\hat{x} - y\| \leq \frac{2\epsilon}{\beta}, \qquad \|v + \mu \nabla g(y)\| \leq \epsilon, \qquad \mu g(y) = 0. \tag{19}$$

For fixed problem parameters, the algorithm uses $TK = \widetilde{O}(\epsilon^{-4})$ *stochastic subgradients and exactly one projection onto* $\mathcal{C}$*.*

Remark 3.9. Here we also achieve the standard ϵ^{-4} rate. The stochastic-oracle exponent is unchanged, although its constants now depend on the infeasible-slope margin and the penalty's weak convexity. Unlike the convex-constraint results above, this theorem does not require a feasible initial

point. It uses no projections onto local feasible-set models and does not keep the intermediate iterates feasible. It does, however, assume an oracle that returns an *exact nearest point* in the last line. Compactness ensures such a point exists on every outcome; uniqueness is guaranteed on the stated success event. Local uniqueness is not a computational algorithm for finding that point. Thus our bound counts one call to a potentially expensive projection oracle and does not claim polynomial-time nonconvex projection. For smooth f, replace v in (19) by $\nabla f(y)$ and we can keep the same rate.

4 Proof Overview

We explain the two different roles of the penalty and then show why the single final projection is sufficient. The complete proofs, including all parameter choices and probability bounds, are in the appendix.

Convex Constraints: Control Infeasibility Before Projecting. The inner objective is strongly convex even when f is nonsmooth. A self-contained stochastic subgradient bound gives inner function error $\widetilde{O}(1/K)$ (Lemma A.1). The softplus differs from $\lambda_0[g]_+$ by at most $\gamma \log 2$. Consequently, with a common inner error $e = \widetilde{O}(1/K)$ and $y_t = Q_\ell(x_t)$,

$$F_t(x_{t+1}) + \lambda_0[g(x_{t+1})]_+ \le F_t(y_t) + e. \tag{20}$$

Set $G_F := G_f + 4\ell$, a Lipschitz bound for F_t on the unit ball. The error bound for g and the choice $\lambda_0\sigma = 2G_F$ imply

$$d_{t+1} := \operatorname{dist}(x_{t+1}, \mathcal{C}) \le e/G_F, \qquad \|x_{t+1} - y_t\|^2 \le 2e/\ell. \tag{21}$$

The first estimate controls infeasibility without computing a projection. The second follows from strong convexity of the exact-penalty subproblem.

For the descent step, use $p_t = \mathcal{P}_{\mathcal{C}}(x_t)$ only as an analytical comparison point. Since y_t minimizes F_t over $\mathcal{C}$,

$$\ell \|x_{t+1} - x_t\|^2 \le f(x_t) - f(x_{t+1}) + G_f d_t + \ell d_t^2 + e. \tag{22}$$

Summing (22) and using (21) controls the average squared outer displacement. For smooth f, nonexpansiveness of the convex projection and Lipschitz continuity of ∇f transfer this control to $\mathcal{G}_\ell(p_{t+1})^2$. For nonsmooth f, Lipschitz continuity of Q_ℓ instead controls $\mathcal{S}_\ell(p_{t+1})^2$. Both bounds have the form $O(T^{-1} + e + e^2)$ for fixed problem parameters. Taking $T = O(\epsilon^{-2})$ and $e = O(\epsilon^2)$ proves Theorems 3.3 and 3.5 after random selection and one final projection.

Nonconvex Constraints: Make the Proximal Witness Feasible. We do not try to make every penalized subproblem an exact reformulation of a constrained proximal problem. Instead, Ψ_λ is a fixed weakly convex objective, and β makes its proximal subproblems strongly convex. Inexact proximal descent gives

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathcal{R}_{\lambda,\beta}(x_t)^2 \le \frac{4\beta\Delta_\lambda}{T} + 4\beta\left(1 + \frac{\beta}{m}\right)e. \tag{23}$$

This is the stochastic progress argument. The feasibility argument is separate. If a proximal witness $y = Q_{\lambda,\beta}(x)$ were infeasible, its optimality condition would read

$$\beta(x - y) = v + \lambda\nabla g(y) + n, \qquad v \in \partial f(y), \quad n \in N_{\mathcal{B}}(y).$$

Assumption 3.7 would then imply $\mathcal{R}_{\lambda,\beta}(x) \ge \lambda\sigma_{\text{out}} - G_f = h_{\text{feas}}$. Hence a smaller residual forces $y \in \mathcal{C}$. Because $\mathcal{C} \subset \operatorname{int}\mathcal{B}$, the artificial ball normal disappears, and the same optimality condition is an approximate KKT condition for (1).

Finally, for any nearest point $p \in \mathcal{P}_{\mathcal{C}}(x)$, $\|p - y\| \le \|p - x\| + \|x - y\| \le 2\|x - y\| = \frac{2\mathcal{R}_{\lambda,\beta}(x)}{\beta}$. Thus the final correction preserves proximity to the KKT witness without requiring a globally nonexpansive nonconvex projection. The small residual also places x in the unique-projection tube. Combining this observation with (23) proves Theorem 3.8.

5 EXPERIMENTS

We compare One Projection with Projected SGD on transfer learning and classification with abstention, reporting means and sample standard deviations over ten runs. Appendix D provides the experimental settings, projection procedures, and additional results.

5.1 TRANSFER LEARNING

To adapt to a target distribution while retaining source performance, we consider constrained empirical risk minimization (Hanneke & Kpotufe, 2024; Deng & Kpotufe, 2025):

$$\min_{\theta} R_T(\theta) \qquad \text{subject to} \qquad R_S(\theta) \le \tau_S,$$

where R_S, R_T are empirical logistic risks and $\tau_S = 1/\sqrt{n_S}$ for n_S source observations. We use a linear model on synthetic Gaussian data and a Softplus neural network on ClimSim rainfall data (Yu et al., 2023), giving convex and nonconvex reference problems, respectively.

Table 1 shows comparable target errors and logistic risks, with $6.9\times$ and $43.2\times$ lower mean runtime for One Projection. Runtime includes source-reference fitting and constrained target optimization. Final source risks satisfy the numerical tolerance; update budgets differ across methods (Appendix D.1). Figure 1 summarizes runtimes for both tasks.

Table 1: Transfer learning: target test performance and training runtime (mean $\pm$ sample standard deviation over ten runs). Update budgets and source-risk diagnostics are given in Appendix D.1.

Dataset	Method	Target error (%)	Target logistic risk	Runtime (s)
Gaussian	One Projection	10.00 ± 0.81	0.253 ± 0.030	9.74 ± 0.23
	Projected SGD	9.92 ± 0.66	0.252 ± 0.033	67.55 ± 17.90
ClimSim	One Projection	7.26 ± 1.31	0.333 ± 0.079	15.25 ± 2.12
	Projected SGD	7.13 ± 1.86	0.323 ± 0.073	658.91 ± 116.26

5.2 CLASSIFICATION WITH ABSTENTION

Following Kalan et al. (2026a), we learn scores s_0, s_1 that accept class j when $s_j(x) \ge 0$ and abstain on simultaneous acceptance. Using the hinge surrogate $\phi(u) = [1+u]_+$, we minimize ambiguity subject to class-specific error constraints:

$$\min_{\theta_0,\theta_1} \widehat{R}_{\mathrm{amb},\phi}(\theta_0,\theta_1) \qquad \text{subject to} \qquad \widehat{R}_{\phi,j}(\theta_j) \le \beta_j + \epsilon_j, \quad j \in \{0,1\}.$$

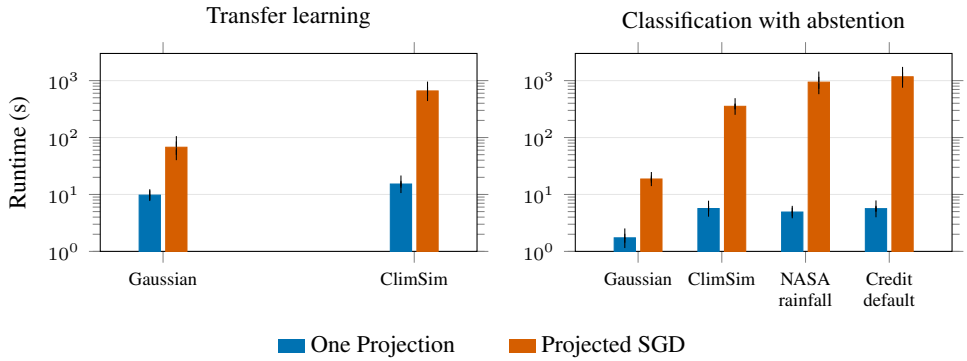

Figure 1: Training runtime over ten runs (mean $\pm$ sample standard deviation; logarithmic scale). Transfer includes source-reference fitting. Update budgets and optimizer-specific thresholds are given in Appendix D.

Here $\widehat{R}_{\mathrm{amb},\phi}$ is the empirical product surrogate, $\widehat{R}_{\phi,j}$ is the class-j surrogate error, and ϵ_j is a constraint slack. The objective is generally nonconvex; neural-network scores also yield nonconvex constraints.

We use linear scores on Gaussian data and neural networks on ClimSim, NASA POWER rainfall (Stackhouse Jr et al., 2021), and Give Me Some Credit (Kaggle, 2011). At 500 updates, One Projection attains similar errors with comparable or lower ambiguity, using 10.8–207.4× less runtime (Table 2). Figure 2 shows the ClimSim training curves. These compare validation-calibrated tradeoffs: some surrogate thresholds differ by optimizer, and nominal test-error targets are not uniformly met (Appendix D.2).

Table 2: Classification with abstention: test errors, ambiguity, and training runtime (mean ± sample standard deviation over ten runs). OP denotes One Projection and PSGD denotes Projected SGD. The validation-calibrated thresholds are listed in Table D.4.

Dataset	Method	Class 0 error (%)	Class 1 error (%)	Ambiguity (%)	Runtime (s)
Gaussian	OP	12.84 ± 2.60	13.89 ± 2.59	11.10 ± 2.28	1.73 ± 0.31
	PSGD	12.99 ± 2.50	14.04 ± 2.24	11.15 ± 2.53	18.69 ± 1.30
ClimSim	OP	9.35 ± 0.43	5.00 ± 1.00	1.72 ± 0.44	5.66 ± 0.61
	PSGD	9.00 ± 0.86	5.09 ± 1.73	2.95 ± 1.03	353.63 ± 42.62
NASA rainfall	OP	9.68 ± 0.42	9.55 ± 1.74	9.17 ± 0.76	4.91 ± 0.18
	PSGD	9.80 ± 0.41	10.74 ± 2.84	8.62 ± 1.75	940.30 ± 223.84
Credit default	OP	16.75 ± 2.27	21.20 ± 2.54	14.95 ± 2.89	5.63 ± 0.70
	PSGD	17.80 ± 2.04	20.44 ± 2.96	16.05 ± 2.95	1167.85 ± 237.30

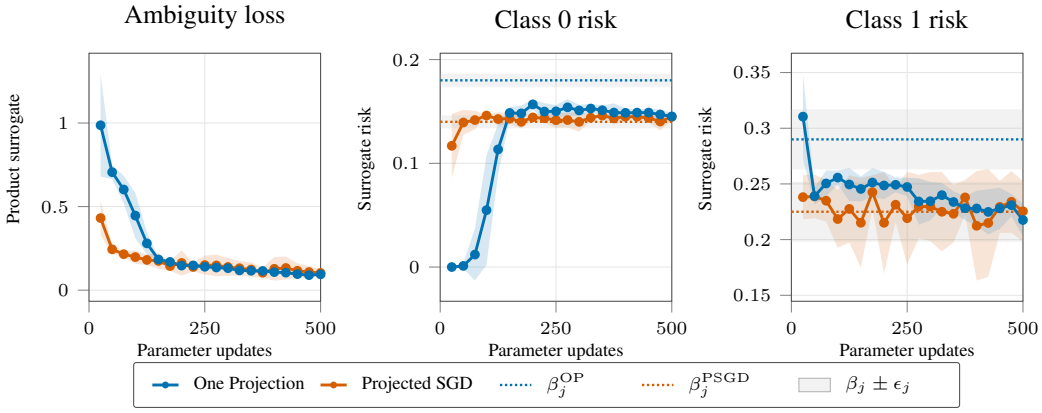

Figure 2: **ClimSim rainfall abstention.** Training surrogate losses (mean ± one standard deviation over ten runs). Dotted lines mark the validation-calibrated class-j surrogate thresholds β_j for each optimizer; gray bands show $\beta_j \pm \epsilon_j$, with constraint slack ϵ_j. Calibration and checkpoint details are in Appendix D.2.

6 Conclusion

In this paper we propose a proximal penalized subgradient method with one projection that can solve nonconvex optimization with convex or nonconvex constraints efficiently. Our rates match the best known rates in unconstrained nonconvex optimization. These results extend the existing literature to the nonconvex regime, and have significant applications in modern machine learning, such as transfer learning and safe LLM finetuning. The experiments on transfer learning and classification with abstention show that our method can achieve similar performance compared to projected SGD, while saving significant amount of runtime.

AI USE STATEMENT

In this work, we used generative AI tools for assisting in the writing of proofs and implementing methods. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, interpreting results. Assisting with translation is not applicable to this work. Additionally, we used generative AI tools for editing a research paper to improve readability. We have reviewed all AI-assisted work. The polished draft was checked and LLM-generated code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Yossi Arjevani, Yair Carmon, John C. Duchi, Dylan J. Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 199(1–2): 165–214, 2023. doi: 10.1007/s10107-022-01822-7. URL https://doi.org/10.1007/s10107-022-01822-7.

Digvijay Boob, Qi Deng, and Guanghui Lan. Stochastic first-order methods for convex and nonconvex functional constrained optimization. arXiv:1908.02734, 2019. URL https://arxiv.org/abs/1908.02734.

Damek Davis and Dmitriy Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019. doi: 10.1137/18M1178244. URL https://doi.org/10.1137/18M1178244.

Damek Davis and Benjamin Grimmer. Proximally guided stochastic subgradient method for nonsmooth, nonconvex problems. *SIAM Journal on Optimization*, 29(3):1908–1930, 2019. doi: 10.1137/17M1151031. URL https://doi.org/10.1137/17M1151031.

Damek Davis, Dmitriy Drusvyatskiy, and Zhan Shi. Stochastic optimization over proximally smooth sets. *SIAM Journal on Optimization*, 35(1):157–179, 2025. doi: 10.1137/20M1320225. URL https://doi.org/10.1137/20M1320225.

Yuyang Deng and Samory Kpotufe. Mixed-sample sgd: an end-to-end analysis of supervised transfer learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 92917–92954. Curran Associates, Inc., 2025. doi: 10.52202/085713-3108. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/85d4d3ba2d10ca59a3cff87d5692474f-Paper-Conference.pdf.

Steve Hanneke and Samory Kpotufe. Adaptive sample aggregation in transfer learning. *arXiv preprint arXiv:2408.16189*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Martin Jaggi. Revisiting Frank–Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 427–435. PMLR, 2013. URL https://proceedings.mlr.press/v28/jaggi13.html.

Uijeong Jang, Jason D. Lee, and Ernest K. Ryu. LoRA training in the NTK regime has no spurious local minima. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21306–21328. PMLR, 2024. URL https://proceedings.mlr.press/v235/jang24d.html.

Zhichao Jia and Benjamin Grimmer. First-order methods for nonsmooth nonconvex functional constrained optimization with or without Slater points. arXiv:2212.00927, 2022. URL https://arxiv.org/abs/2212.00927.

Jikai Jin. A tight lower bound for smooth nonconvex stochastic optimization with bounded gradient noise. arXiv:2608.09004, 2026. URL https://arxiv.org/abs/2608.09004.

Kaggle. Give Me Some Credit. Kaggle competition dataset, 2011. URL https://www.kaggle.com/competitions/GiveMeSomeCredit. Accessed September 25, 2026.

Mohammadreza M. Kalan, Yuyang Deng, and Sanaz Hamidi. Classification with abstention under class-conditional error constraints. 2026a. URL https://arxiv.org/abs/2609.22632.

Mohammadreza Mousavi Kalan, Yuyang Deng, Eitan J. Neugut, and Samory Kpotufe. Neyman-pearson classification under both null and alternative distributions shift. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=pHckxhmBlI.

Wei Liu and Yangyang Xu. A SPIDER-type stochastic subgradient method for expectation-constrained nonconvex nonsmooth optimization. arXiv:2501.19214v2, 2025. URL https://arxiv.org/abs/2501.19214v2.

Runchao Ma, Qihang Lin, and Tianbao Yang. Quadratically regularized subgradient methods for weakly convex optimization with weakly convex constraints. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6554–6564. PMLR, 2020. URL https://proceedings.mlr.press/v119/ma20d.html.

Mehrdad Mahdavi, Tianbao Yang, Rong Jin, Shenghuo Zhu, and Jinfeng Yi. Stochastic gradient descent with only one projection. In *Advances in Neural Information Processing Systems*, volume 25, 2012. URL https://papers.neurips.cc/paper_files/paper/2012/hash/c52f1bd66cc19d05628bd8bf27af3ad6-Abstract.html.

JuneYoung Park, Minjae Kang, Seongbae Lee, Haegang Lee, Seongwan Kim, and Jaeho Lee. Riemannian optimization for LoRA on the stiefel manifold. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 20971–20985, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.1143. URL https://aclanthology.org/2025.findings-emnlp.1143/.

Xiyue Peng, Hengquan Guo, Jiawei Zhang, Dongqing Zou, Ziyu Shao, Honghao Wei, and Xin Liu. Enhancing safety in reinforcement learning with human feedback via rectified policy optimization. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2120.

Paul W Stackhouse Jr, Bradley Macpherson, Madison Broddle, Chequel McNeil, Jason Barnett, Colleen Mikovitz, and Taiping Zhang. The prediction of worldwide energy resources (power) project. In *United Nations Climate Change Conference, Community of Practice 26*, 2021.

Tianbao Yang, Qihang Lin, and Lijun Zhang. A richer theory of convex constrained optimization with reduced projections and improved rates. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3901–3910. PMLR, 2017. URL https://proceedings.mlr.press/v70/yang17f.html.

Sungduk Yu, Walter Hannah, Liran Peng, Jerry Lin, Mohamed Aziz Bhouri, Ritwik Gupta, Björn Lütjens, Justus C. Will, Gunnar Behrens, Julius J. M. Busecke, et al. Climsim: A large multi-scale dataset for hybrid physics–ml climate emulation. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0968. URL https://papers.neurips.cc/paper_files/paper/2023/hash/45fbcc01349292f5e059a0b8b02c8c3f-Abstract-Datasets_and_Benchmarks.html.

Botong Zhang, Shuo Li, Ignacio Hounie, Osbert Bastani, Dongsheng Ding, and Alejandro Ribeiro. Alignment of large language models with constrained learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1042.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lq62uWRJjiY.

A A High-Probability Inner-Solver Bound

We first give a bound that applies to both penalty choices. All probability statements can be conditioned on the history before an inner run. This is important because the next proximal objective depends on the preceding stochastic iterates.

Lemma A.1 (Strongly convex stochastic subgradient subroutine)**.** *Let H be m-strongly convex on a neighborhood of $\mathcal{B}$, where $m > 0$. Suppose $\widetilde{h}_s$ is a stochastic subgradient at $z_s \in \mathcal{B}$ with $\mathbb{E}[\widetilde{h}_s \mid \mathcal{F}_s] \in \partial H(z_s)$ and $\left\|\widetilde{h}_s\right\| \le G$ almost surely, for some $G > 0$. Starting at any $z_1 \in \mathcal{B}$, set*

$$z_{s+1} = \mathcal{P}_{\mathcal{B}}(z_s - \eta_s \widetilde{h}_s), \qquad \eta_s = \frac{2}{m(s+1)}, \qquad \bar{z} = \frac{1}{K}\sum_{s=1}^{K} z_s.$$

For $0 < \delta < 1$, with probability at least $1 - \delta$,

$$H(\bar{z}) - \min_{z \in \mathcal{B}} H(z) \le U_K(m, G, \delta) := \frac{m + (G^2/m)\{\log(K+1) + 2\log(1/\delta)\}}{K}. \tag{24}$$

Proof. Let u minimize H over $\mathcal{B}$, put $D_s = \|z_s - u\|$, and write $h_s = \mathbb{E}[\widetilde{h}_s \mid \mathcal{F}_s]$ and $X_s = \left\langle h_s - \widetilde{h}_s, z_s - u \right\rangle$. Nonexpansiveness of the ball projection and strong convexity imply

$$\begin{aligned} H(z_s) - H(u) &\le \langle h_s, z_s - u\rangle - \frac{m}{2} D_s^2 \\ &\le \frac{D_s^2 - D_{s+1}^2}{2\eta_s} - \frac{m}{2} D_s^2 + \frac{\eta_s G^2}{2} + X_s. \end{aligned}$$

Summing with the stated stepsize and dropping the negative terminal term gives

$$\sum_{s=1}^{K} (H(z_s) - H(u)) \le \frac{m}{4} D_1^2 - \frac{m}{4}\sum_{s=1}^{K} D_s^2 + \frac{G^2}{m}\sum_{s=1}^{K}\frac{1}{s+1} + \sum_{s=1}^{K} X_s. \tag{25}$$

Conditionally on $\mathcal{F}_s$, X_s is centered and its range has length at most $2GD_s$. The conditional bounded-variable exponential inequality gives

$$\mathbb{E}[\exp(\theta X_s) \mid \mathcal{F}_s] \le \exp(\theta^2 G^2 D_s^2/2).$$

Iterating conditional expectations and applying Markov's inequality shows that, with probability at least $1 - \delta$,

$$\sum_{s=1}^{K} X_s \le \frac{\theta G^2}{2}\sum_{s=1}^{K} D_s^2 + \frac{\log(1/\delta)}{\theta}.$$

Set $\theta = m/(2G^2)$. The squared-distance sum cancels the negative sum in (25). Since $D_1 \le 2$ and $\sum_{s=1}^{K}(s+1)^{-1} \le \log(K+1)$, division by K and convexity of H prove (24). □

There is no differentiability requirement on H. In particular, neither the small softplus smoothing parameter nor the kink of the hinge changes this subgradient bound. The function error is $\widetilde{O}((m + G^2/m)/K)$, rather than a smooth-gradient bound involving the penalty's smoothness constant.

B Proofs for Convex Constraints

Throughout this section Assumption 3.1 holds. Define

$$G_F := G_f + 4\ell, \qquad G_H := G_F + \lambda_0 G_g, \qquad e_K(\delta) := U_K(\ell, G_H, \delta) + \frac{\log 2}{K}. \tag{26}$$

Thus $\lambda_0\sigma - G_F = G_F > 0$ under (6). This positive margin is an explicit consequence of the parameter choice, not an additional assumption.

B.1 Constraint Error Bound and Exact Penalization

Lemma B.1 (Error bound induced by the convex defining function)**.** *For every $x \in \mathcal{B}$,*

$$\operatorname{dist}(x, \mathcal{C}) \le \frac{[g(x)]_+}{\sigma}. \tag{27}$$

Proof. The claim is immediate for $x \in \mathcal{C}$. Otherwise let $p = \mathcal{P}_{\mathcal{C}}(x)$. Compactness of $\mathcal{C}$ gives existence, and convexity gives uniqueness. The point p lies on $g = 0$. Because $\nabla g(p) \neq 0$, the normal cone is $N_{\mathcal{C}}(p) = \{\alpha \nabla g(p) : \alpha \ge 0\}$. The projection condition gives $x - p = \alpha \nabla g(p)$ for some $\alpha > 0$. Convexity then yields

$$g(x) \ge g(p) + \langle \nabla g(p), x - p \rangle = \|\nabla g(p)\| \, \|x - p\| \ge \sigma \operatorname{dist}(x, \mathcal{C}).$$

□

Lemma B.2 (Inner accuracy, infeasibility, and proximal accuracy)**.** *Fix $x_t \in \mathcal{B}$, let $y_t = Q_\ell(x_t)$, and run the inner routine with the convex settings. With conditional probability at least $1 - \delta$,*

$$F_t(x_{t+1}) + \lambda_0 [g(x_{t+1})]_+ \le F_t(y_t) + e_K(\delta), \tag{28}$$

$$\operatorname{dist}(x_{t+1}, \mathcal{C}) \le e_K(\delta)/G_F, \tag{29}$$

$$\|x_{t+1} - y_t\|^2 \le 2 e_K(\delta)/\ell. \tag{30}$$

Proof. The function F_t is ℓ-strongly convex. Since p_γ is convex and nondecreasing and g is convex, $H_t := F_t + p_\gamma \circ g$ is also ℓ-strongly convex. The stochastic subgradient used in Algorithm 1 has norm at most

$$G_f + 2\ell \, \|z_s - x_t\| + \lambda_0 G_g \le G_f + 4\ell + \lambda_0 G_g = G_H.$$

Apply Lemma A.1 and compare the minimum of H_t to $H_t(y_t)$. The elementary bounds

$$\lambda_0 [u]_+ \le p_\gamma(u) \le \lambda_0 [u]_+ + \gamma \log 2$$

and $\gamma = 1/K$ prove (28).

For completeness, the exact-penalty objective $P_t(z) := F_t(z) + \lambda_0 [g(z)]_+$ has the same unique minimizer y_t over $\mathcal{B}$ as F_t does over $\mathcal{C}$. Indeed, for any $z \in \mathcal{B}$, set $p = \mathcal{P}_{\mathcal{C}}(z)$ and $d = \|z - p\|$. The function F_t is G_F-Lipschitz on $\mathcal{B}$, so Lemma B.1 gives

$$\begin{aligned} P_t(z) &\ge F_t(p) - G_F d + \lambda_0 \sigma d \\ &\ge F_t(y_t) + G_F d. \end{aligned} \tag{31}$$

This proves exactness and, together with (28), (29). Finally, P_t is ℓ-strongly convex. At its minimizer over the convex set $\mathcal{B}$, quadratic growth gives

$$\frac{\ell}{2} \|x_{t+1} - y_t\|^2 \le P_t(x_{t+1}) - P_t(y_t) \le e_K(\delta),$$

proving (30). □

B.2 Descent and Stationarity Transfer

All projections $p_t = \mathcal{P}_{\mathcal{C}}(x_t)$ and proximal points below are proof objects; only the selected terminal projection is computed. Write $d_t = \|x_t - p_t\|$ and $s_t = \|x_{t+1} - x_t\|$.

Lemma B.3 (Outer descent with infeasible iterates)**.** *Suppose the conclusions of Lemma B.2 hold in every outer iteration with a common error bound e. Then*

$$\frac{1}{T} \sum_{t=0}^{T-1} s_t^2 \le \frac{\Delta_f}{\ell T} + \frac{1 + G_f/G_F}{\ell} e + \frac{e^2}{G_F^2}. \tag{32}$$

Proof. Dropping the nonnegative penalty in (28), using $F_t(y_t) \le F_t(p_t)$, and using Lipschitz continuity of f give

$$\begin{aligned} f(x_{t+1}) + \ell s_t^2 &\le F_t(y_t) + e \\ &\le f(p_t) + \ell d_t^2 + e \\ &\le f(x_t) + G_f d_t + \ell d_t^2 + e. \end{aligned}$$

This proves (22), including its quadratic distance term. Now sum over t, use $d_0 = 0$ and $d_t \le e/G_F$ for $t \ge 1$, and use $f(x_T) \ge \min_{\mathcal{B}} f$. □

Lemma B.4 (Properties of the convex constrained proximal map)**.** *The point $Q_\ell(x)$ is unique and Q_ℓ is 2-Lipschitz. Moreover, Φ_ℓ is differentiable with $\nabla\Phi_\ell(x) = 2\ell(x - Q_\ell(x))$, and* (10) *holds.*

Proof. The function $f + \delta_{\mathcal{C}}$ is proper, lower semicontinuous and ℓ-weakly convex. Compactness gives existence of the proximal minimizer; strong convexity gives uniqueness. For $q_i = Q_\ell(x_i)$, the optimality conditions and monotonicity of the subdifferential of $f + \delta_{\mathcal{C}} + \ell \|\cdot\|^2/2$ give

$$\langle 2\ell(x_1 - q_1) - 2\ell(x_2 - q_2), q_1 - q_2 \rangle \ge -\ell \|q_1 - q_2\|^2 .$$

Consequently, $\|q_1 - q_2\|^2 \le 2\langle x_1 - x_2, q_1 - q_2\rangle$, which proves the Lipschitz bound. Differentiating the minimum with its unique continuous minimizer yields the stated envelope gradient. The proximal optimality condition is

$$2\ell(x - Q_\ell(x)) \in \partial f(Q_\ell(x)) + N_{\mathcal{C}}(Q_\ell(x)).$$

The sum rule is valid since f is locally Lipschitz and weakly convex on a neighborhood of $\mathcal{C}$; equivalently it is the convex sum rule after adding $\ell \|\cdot\|^2/2$. This proves (10). □

Lemma B.5 (The final projection preserves a stationarity certificate)**.** *Under the event in Lemma B.3,*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathcal{S}_\ell(p_{t+1})^2 \le \frac{48\ell\Delta_f}{T} + (72 + 48G_f/G_F)\ell e + \frac{156\ell^2 e^2}{G_F^2}, \tag{33}$$

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathcal{G}_\ell(p_{t+1})^2 \le \frac{12\ell\Delta_f}{T} + (162 + 12G_f/G_F)\ell e + \frac{39\ell^2 e^2}{G_F^2}, \tag{34}$$

where the second statement additionally assumes f is ℓ-smooth.

Proof. Put $E_t = \|x_{t+1} - y_t\|$, so $E_t^2 \le 2e/\ell$. For the nonsmooth measure, Lemma B.4 gives

$$\begin{aligned} \mathcal{S}_\ell(p_{t+1}) &\le 2\ell \|p_{t+1} - y_t\| + 4\ell \|x_t - p_{t+1}\| \\ &\le 4\ell s_t + 2\ell E_t + 6\ell d_{t+1}. \end{aligned}$$

Squaring with $(a+b+c)^2 \le 3(a^2+b^2+c^2)$, averaging, and applying (32) proves (33).

For the smooth measure, nonexpansiveness of the convex projection and ℓ-Lipschitz continuity of ∇f show that $\mathcal{G}_\ell$ is 3ℓ-Lipschitz on $\mathcal{B}$. Proximal optimality also gives

$$y_t = \mathcal{P}_{\mathcal{C}}\left(y_t - \frac{\nabla f(y_t) + 2\ell(y_t - x_t)}{\ell}\right), \qquad \mathcal{G}_\ell(y_t) \le 2\ell \|y_t - x_t\| .$$

It follows that

$$\mathcal{G}_\ell(p_{t+1}) \le 2\ell s_t + 5\ell E_t + 3\ell d_{t+1}.$$

The same squared-sum bound proves (34). □

B.3 Shared Schedule and Proofs of the Convex-Constraint Theorems

Theorems 3.3 and 3.5 use the same schedule. Choose

$$T := \max\left\{2, \left\lceil \frac{1000\ell\Delta_f}{\epsilon^2} \right\rceil\right\}, \qquad \delta_{\text{in}} := \frac{1}{6T}, \\ e_* := \min\left\{\frac{\epsilon^2}{4000\ell}, \frac{G_F\epsilon}{60\ell}\right\}. \tag{35}$$

Take any $K \geq 2$ such that $e_K(\delta_{\text{in}}) \leq e_*$. This is an explicit scalar inequality involving only the specified constants. For example, double K from 2 until it holds. There is such a choice with

$$K = \widetilde{O}\left(1 + \left(1 + \ell + \frac{G_H^2}{\ell}\right)\max\left\{\frac{\ell}{\epsilon^2}, \frac{\ell}{G_F\epsilon}\right\}\right). \tag{36}$$

The logarithmic factors include $\log T$ from δ_{in}.

Condition on the history at the start of each inner run and apply Lemma B.2. Let $\mathcal{E}_{\text{in}}$ be the event that all T inner guarantees hold with error at most e_*. A union bound gives $\Pr(\mathcal{E}_{\text{in}}) \geq 1 - T\delta_{\text{in}} = 5/6$.

Proof of Theorem 3.3. On $\mathcal{E}_{\text{in}}$, (34), (35), and $G_f/G_F \leq 1$ give

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathcal{G}_\ell(p_{t+1})^2 \leq \left(\frac{12}{1000} + \frac{174}{4000} + \frac{39}{3600}\right)\epsilon^2 < \frac{\epsilon^2}{6}.$$

Conditionally on any such trajectory, uniform independent selection of $J \in \{1, \ldots, T\}$ and Markov's inequality imply $\Pr\{\mathcal{G}_\ell(p_J) \leq \epsilon \mid \text{trajectory}\} \geq 5/6$. The unconditional success probability is at least $25/36 > 2/3$. The actual output $\hat{x} = p_J$ is always feasible and requires just one projection onto $\mathcal{C}$. The schedule gives $TK = \widetilde{O}(\epsilon^{-4})$ stochastic gradient calls for fixed problem parameters. □

Proof of Theorem 3.5. On the same event $\mathcal{E}_{\text{in}}$, (33) and (35) give

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathcal{S}_\ell(p_{t+1})^2 \leq \left(\frac{48}{1000} + \frac{120}{4000} + \frac{156}{3600}\right)\epsilon^2 < \frac{\epsilon^2}{6}.$$

Uniform independent selection of J and Markov's inequality again give $\Pr\{\mathcal{S}_\ell(p_J) \leq \epsilon\} \geq 25/36 > 2/3$. The output is exactly feasible, and (10) supplies the nearby-stationarity interpretation. The same schedule and final projection yield the stated stochastic subgradient and projection counts. □

The gap Δ_f need not be computed: any known upper bound may replace it in the schedule, and $\Delta_f \leq 2G_f$ on the unit ball. No evaluation of the stationarity measure or of the proof-only points p_t, y_t is needed to run the algorithm.

C Proofs for Nonconvex Constraints

Throughout this section Assumption 3.7 holds. Set

$$\rho_\lambda := \ell + \lambda L_g, \qquad G_\lambda := G_f + \lambda G_g, \qquad G_H := G_\lambda + 2\beta, \qquad m = \beta - \rho_\lambda > 0.$$

The constants G_H, m in this section refer to the nonconvex configuration.

C.1 Geometry of the Terminal Projection

Lemma C.1 (A projection tube derived from g)**.** *Let $R = \sigma_{\text{bd}}/L_g$. For every $p, y \in \mathcal{C}$ and $n \in N_{\mathcal{C}}(p)$,*

$$\langle n, y - p\rangle \leq \frac{\|n\|}{2R}\|y - p\|^2. \tag{37}$$

Consequently, $\mathcal{P}_{\mathcal{C}}(x)$ is unique whenever $\operatorname{dist}(x, \mathcal{C}) < R$. In particular, $\mathcal{C}$ is R-proximally smooth.

Proof. At an interior point of $\mathcal{C}$ the normal cone is $\{0\}$. At a boundary point p, the nonzero gradient condition yields the smooth inequality normal formula

$$N_{\mathcal{C}}(p) = \{\alpha \nabla g(p) : \alpha \geq 0\}.$$

Indeed, the local tangent cone is the halfspace $\{u : \langle \nabla g(p), u \rangle \leq 0\}$, whose polar is this normal ray. Since $g(p) = 0$ and $g(y) \leq 0$, the smooth lower bound along the segment in $\mathcal{B}$ gives

$$g(y) \geq \langle \nabla g(p), y - p \rangle - \frac{L_g}{2} \|y - p\|^2, \qquad \langle \nabla g(p), y - p \rangle \leq \frac{L_g}{2} \|y - p\|^2.$$

Multiply by α and use $\alpha \leq \|n\| / \sigma_{\mathrm{bd}}$ to obtain (37).

To see uniqueness directly, suppose p_1, p_2 are projections of x, with common distance $d < R$. Then $n_i = x - p_i \in N_{\mathcal{C}}(p_i)$. Applying (37) twice and adding yields

$$\|p_1 - p_2\|^2 \leq \frac{d}{R} \|p_1 - p_2\|^2.$$

Thus $p_1 = p_2$. Projections exist by compactness, so this proves the unique-projection tube characterization of proximal smoothness. □

C.2 Penalty Residual and Original-Problem Stationarity

The function $[g]_+$ is L_g-weakly convex because

$$[g(x)]_+ + \frac{L_g}{2} \|x\|^2 = \max\left\{ g(x) + \frac{L_g}{2} \|x\|^2, \frac{L_g}{2} \|x\|^2 \right\}$$

is convex. Thus Ψ_λ is ρ_λ-weakly convex, and $Q_{\lambda,\beta}(x)$ is well-defined and unique. Since f is Lipschitz on a neighborhood of $\mathcal{B}$, all of its limiting subgradients on $\mathcal{B}$ have norm at most G_f.

Lemma C.2 (A small penalty residual has a feasible KKT witness)**.** *For $x \in \mathcal{B}$, write $y = Q_{\lambda,\beta}(x)$ and $r = \mathcal{R}_{\lambda,\beta}(x)$. If $r < h_{\mathrm{feas}}$, then $y \in \mathcal{C}$ and there are $v \in \partial f(y)$ and $\mu \in [0, \lambda]$ with*

$$\beta(x - y) = v + \mu \nabla g(y), \qquad \mu g(y) = 0. \tag{38}$$

Every $p \in \mathcal{P}_{\mathcal{C}}(x)$ then satisfies

$$\mathrm{dist}(x, \mathcal{C}) \leq \frac{r}{\beta}, \qquad \|p - y\| \leq \frac{2r}{\beta}, \qquad \mathrm{dist}(0, \partial f(y) + N_{\mathcal{C}}(y)) \leq r. \tag{39}$$

If additionally $r < \beta R$, the projection is unique.

Proof. Convex subdifferential calculus after a quadratic shift, or the equivalent weakly convex calculus, gives

$$\beta(x - y) = v + \lambda\theta \nabla g(y) + n, \qquad v \in \partial f(y), \quad n \in N_{\mathcal{B}}(y), \tag{40}$$

where $\theta = 1$ for $g(y) > 0$, $\theta = 0$ for $g(y) < 0$, and $\theta \in [0, 1]$ for $g(y) = 0$. If $g(y) > 0$, then $N_{\mathcal{B}}(y)$ is a cone and

$$r = \|v + \lambda \nabla g(y) + n\| \geq \lambda \,\mathrm{dist}(0, \nabla g(y) + N_{\mathcal{B}}(y)) - G_f \geq h_{\mathrm{feas}},$$

a contradiction. Thus $y \in \mathcal{C} \subset \mathrm{int}\, \mathcal{B}$ and $n = 0$. Set $\mu = \lambda\theta$. This proves (38), including nonnegativity and exact complementarity. The boundary normal formula from Lemma C.1 shows that $\mu \nabla g(y) \in N_{\mathcal{C}}(y)$, proving the stationarity claim.

Finally $\mathrm{dist}(x, \mathcal{C}) \leq \|x - y\| = r/\beta$. For any nearest point p, the triangle inequality gives $\|p - y\| \leq \|p - x\| + \|x - y\| \leq 2r/\beta$. Lemma C.1 gives uniqueness when $r < \beta R$. □

C.3 Inexact Proximal Descent

Lemma C.3 (Average penalty residual)**.** *Suppose every inner run returns $x_{t+1} \in \mathcal{B}$ such that*

$$H_t(x_{t+1}) - H_t(y_t) \leq e, \qquad H_t(z) := \Psi_\lambda(z) + \frac{\beta}{2} \|z - x_t\|^2, \qquad y_t = Q_{\lambda,\beta}(x_t). \tag{41}$$

Then we have

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{R}_{\lambda,\beta}(x_t)^2 \leq \frac{4\beta\Delta_\lambda}{T} + 4\beta \left(1 + \frac{\beta}{m}\right) e. \tag{42}$$

Proof. Strong convexity and (41) imply $\|x_{t+1} - y_t\|^2 \le 2e/m$. Since $H_t(y_t) \le H_t(x_t) = \Psi_\lambda(x_t)$, we also have

$$\Psi_\lambda(x_{t+1}) + \frac{\beta}{2}\|x_{t+1} - x_t\|^2 \le \Psi_\lambda(x_t) + e.$$

Sum this inequality and use the definition of Δ_λ to obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\|x_{t+1} - x_t\|^2 \le \frac{2\Delta_\lambda}{\beta T} + \frac{2e}{\beta}.$$

Finally,

$$\mathcal{R}_{\lambda,\beta}(x_t)^2 \le 2\beta^2\|x_t - x_{t+1}\|^2 + 2\beta^2\|x_{t+1} - y_t\|^2.$$

Averaging concludes the proof. □

C.4 Proof of Theorem 3.8

Choose

$$\begin{aligned} T &:= \max\left\{2, \left\lceil \frac{48\beta\Delta_\lambda}{\epsilon^2}\right\rceil\right\}, \qquad \delta_{\text{in}} := \frac{1}{6T}, \\ e_* &:= \frac{\epsilon^2}{48\beta(1+\beta/m)}. \end{aligned} \tag{43}$$

Take $K \ge 2$ such that $U_K(m, G_H, \delta_{\text{in}}) \le e_*$. Doubling K from 2 until this scalar inequality holds is sufficient and gives

$$K = \widetilde{O}\left(1 + \left(m + \frac{G_H^2}{m}\right)\frac{\beta(1+\beta/m)}{\epsilon^2}\right). \tag{44}$$

The actual inner update is an unbiased stochastic subgradient step for H_t on $\mathcal{B}$, with norm bounded by $G_f + \lambda G_g + 2\beta = G_H$. At $g = 0$, its chosen hinge coefficient zero is admissible. Lemma A.1, conditional on the past of each run, and a union bound give (41) for all t with probability at least $5/6$. On this event Lemma C.3 and (43) yield

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathcal{R}_{\lambda,\beta}(x_t)^2 \le \frac{\epsilon^2}{12} + \frac{\epsilon^2}{12} = \frac{\epsilon^2}{6}.$$

The independent uniform index $J \in \{0, \dots, T-1\}$ therefore satisfies $\mathcal{R}_{\lambda,\beta}(x_J) \le \epsilon$ with conditional probability at least $5/6$. As before, the unconditional success probability is at least $25/36 > 2/3$. The strict target restrictions $\epsilon < h_{\text{feas}}$ and $\epsilon < \beta R$ allow us to apply Lemma C.2, which gives (19) and unique terminal projection. Regardless of this event, $\mathcal{C}$ is compact and nonempty, so any exact nearest point returned by the oracle is feasible. This proves Theorem 3.8.

A known upper bound on Δ_λ can replace it in the schedule; in particular $\Delta_\lambda \le 2G_\lambda$. No proximal residual or distance to $\mathcal{C}$ is evaluated during the iterations. The guarantee is a nearby-stationarity certificate, not a claim that the unknown witness y or its multiplier is explicitly returned.

C.5 Why Boundary Regularity Alone Does Not Suffice

Consider the one-dimensional problem with $f(x) = 0$, $\mathcal{B} = [-1, 1]$, and

$$g(x) = \left(x^2 - \frac{1}{4}\right)^2 - \frac{1}{64}.$$

Its feasible set is $\mathcal{C} = \{x : 1/8 \le x^2 \le 3/8\} \subset (-1, 1)$, with two components. The gradient is Lipschitz on a neighborhood of $\mathcal{B}$, and it is nonzero at all four boundary points. Nevertheless, $g(0) = 3/64 > 0$ and $g'(0) = 0$. Starting Algorithm 1 at $x_0 = 0$ with the exact zero objective oracle leaves every inner and outer iterate at zero. Indeed, if $\beta > \ell + \lambda L_g$, the penalized proximal objective is strongly convex and has derivative zero at zero, so its exact minimizer is also zero. Consequently the penalty residual is zero there, but its proximal witness is infeasible.

This example does not show that every first-order method fails, nor does it say that the final projected point in this particular constant-objective problem is nonstationary. It shows precisely why boundary regularity cannot justify the implication from a small penalty residual to a feasible proximal witness. The global condition (13) excludes this obstruction. It is a substantive restriction, especially for disconnected or complicated nonconvex feasible regions.

Why not require every proximal subproblem to be feasible? For nonconvex g, requiring every penalized proximal subproblem to have a feasible minimizer can couple the penalty coefficient to β, while strong convexity simultaneously requires $\beta > \ell + \lambda L_g$. Those demands need not be compatible. Our proof imposes no such per-subproblem exactness requirement. It first fixes $\lambda > G_f/\sigma_{\mathrm{out}}$, then fixes $\beta > \ell + \lambda L_g$, and only forces feasibility of a proximal witness once its residual is below h_{feas}. No compatibility condition between an inner exact-penalty margin and the proximal curvature is needed.

D Additional Experimental Details

D.1 Transfer Learning

Data and objective. Table D.1 lists the source and target training sample sizes and model architectures. For a labeled sample D and a real-valued score s_θ, the empirical logistic risk is

$$R_D(\theta) = \frac{1}{|D|} \sum_{(x,y)\in D} \log\big(1 + \exp(-(2y-1)s_\theta(x))\big), \qquad y \in \{0,1\}.$$

We minimize R_T subject to $R_S \le \tau_S$, with $\tau_S = 1/\sqrt{n_S}$. This is a cap on the raw empirical source risk, not an excess-risk constraint of the form $R_S \le \inf_\theta R_S(\theta) + \tau_S$; no assumption that the minimum source logistic risk equals zero is needed to describe these experiments.

Table D.1: Transfer-learning data and models. Counts are class 0/class 1 in the target and source training samples.

Dataset	Target	Source	Variables	Model
Gaussian	40/40	1000/1000	10	Linear with intercept
ClimSim rainfall	100/100	950/950	124	[8] Softplus MLP

Budgets and reporting. Both methods are evaluated over ten paired test seeds. The numbers of constrained target updates differ across methods and are reported in Table D.2; they exclude the source-reference fitting stage. The recorded runtime includes both source-reference fitting and constrained target optimization. Speedups are ratios of mean runtimes, not means of per-seed ratios. Figure 1 in the main text summarizes these runtime comparisons. The ClimSim final source risk rounds to 0.02295, while the displayed source-risk limit rounds to 0.02294. The unrounded residual is within the configured feasibility tolerance of 10^{-5}; all locked runs were therefore recorded as numerically feasible.

Table D.2: Transfer-learning update budgets and final empirical source risks. Source risks are means $\pm$ sample standard deviations over ten runs; zeros in the reported standard deviations reflect rounding.

Dataset	Method	Updates	Source-risk cap	Final source risk
Gaussian	One Projection	3,000	0.02236	0.022 ± 0.001
	Projected SGD	500	0.02236	0.022 ± 0.000
ClimSim	One Projection	1,000	0.02294	0.02295 ± 0.00000
	Projected SGD	100	0.02294	0.02295 ± 0.00000

Synthetic Gaussian data. The synthetic experiment used 10-dimensional Gaussian observations with identity covariance. The target distributions were

$$X_{T,0} \sim N(\mathbf{0}, I), \qquad X_{T,1} \sim N(\mathbf{1}, I),$$

and the source distributions were

$$X_{S,0} \sim N(-2\mathbf{1}, I), \qquad X_{S,1} \sim N(\mathbf{1}, I).$$

Thus, class 1 had the same distribution in both domains, whereas source class 0 was shifted relative to target class 0. Each seed used 40 target-training and 1,000 source-training observations per class. The target validation and test sets each contained 500 observations per class. We used a linear score with an intercept.

ClimSim rainfall data. We constructed domains by grouping consecutive location identifiers into blocks of four. The target domain comprised locations 104–107 (derived group 26), and the source domain comprised locations 108–111 (derived group 27). Heavy rain was defined as a rain rate above the 95th percentile of nonzero training-period rainfall. This threshold was fixed before constructing the validation and test labels. The data were divided chronologically, and preprocessing was fitted using the union of the source and target training samples only.

The experiment used deliberately balanced samples rather than the natural rainfall prevalence. Each seed contained 100 target-training and 950 source-training observations per class, 200 target-validation observations per class, and 400 target-test observations per class. The model was a one-hidden-layer $[8]$ Softplus MLP receiving the 124 climate variables.

Optimization settings. For Gaussian data, the shared source-reference model was fitted for 1,000 updates with learning rate 0.05. One Projection used learning rate 0.5, 3,000 target-gradient updates, and minibatches of size 32; Projected SGD used learning rate 1.0, 500 updates, and the same batch size. The linear source-risk projection was solved as a convex numerical projection.

For ClimSim, the shared source-reference model was fitted for 200 updates with learning rate 0.003. One Projection used learning rate 0.05, 1,000 target-gradient updates, and minibatches of size 40; Projected SGD used learning rate 0.3, 100 updates, and the same batch size. Neural source-risk projections used the augmented-Lagrangian solver with feasibility tolerance 10^{-5}. All reported timing experiments ran on the CPU.

Training curves. Figures 3 and 4 show target validation risk, target training risk, and source risk against cumulative updates. The two methods are shown only over their respective update budgets.

Points show means over ten runs and shaded regions show one sample standard deviation. One Projection points are outer-loop outputs plotted at their cumulative gradient-step counts, while Projected SGD points are stored evaluation checkpoints. The final One Projection point includes the terminal feasibility projection. The dotted horizontal line denotes the maximum permitted empirical source risk.

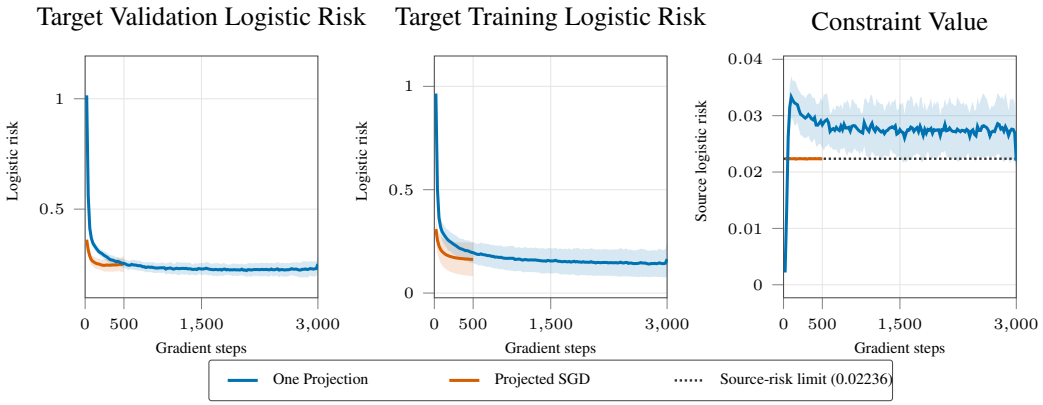

Figure 3: **Synthetic Gaussian Transfer.** Mean target validation risk, target training risk, and empirical source-risk constraint value across ten paired test seeds. Lines show means and shaded regions show one sample standard deviation. The final One Projection value is evaluated after its single feasibility projection. Lines connect every recorded One Projection outer iterate and every stored Projected SGD evaluation value.

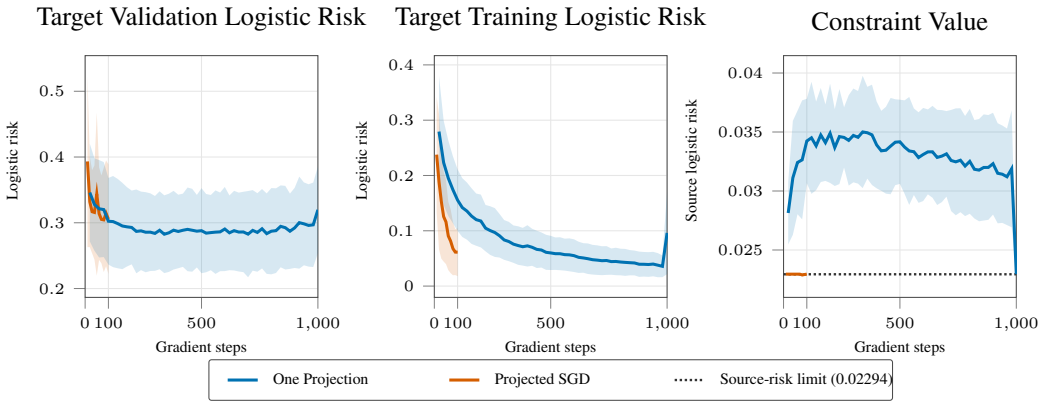

Figure 4: **ClimSim Rainfall Transfer.** Mean target validation risk, target training risk, and empirical source-risk constraint value across ten paired test seeds. Lines show means and shaded regions show one sample standard deviation. The final One Projection value is evaluated after its single feasibility projection. Lines connect every recorded One Projection outer iterate and every stored Projected SGD evaluation value.

D.2 Classification with Abstention

Empirical problem and metrics. Let $\{(x_i, y_i)\}_{i=1}^n$ be the training sample, $n_j = |\{i : y_i = j\}|$, and $\theta = (\theta_0, \theta_1)$. With $\phi(u) = [1 + u]_+$, we use

$$\widehat{R}_{\mathrm{amb},\phi}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \phi(s_0(x_i;\theta_0))\phi(s_1(x_i;\theta_1)),$$

$$\widehat{R}_{\phi,j}(\theta_j) = \frac{1}{n_j}\sum_{i:y_i=j} \phi(-s_j(x_i;\theta_j)), \qquad j \in \{0,1\}.$$

The empirical feasible set requires $\widehat{R}_{\phi,j}(\theta_j) \le \beta_j + \epsilon_j$ for both classes, retaining the calibration slacks described below. On the test sample, class-j error is the fraction of class-j observations for which $s_j < 0$, and ambiguity is the fraction for which both scores are nonnegative. These indicator metrics differ from the surrogate quantities optimized during training.

Calibration and evaluation. All abstention experiments use 500 parameter updates. Model architecture, feature representation, learning rate, and surrogate thresholds (β_0, β_1) were selected using validation data. The thresholds were calibrated separately for each optimizer to give comparable error and ambiguity rates, then held fixed for evaluation over ten test seeds. Test data were not used for model selection or hyperparameter tuning. Table D.3 gives the nominal error targets α_j, and Table D.4 reports the chosen surrogate thresholds and accuracy among decided observations. Here β_j denotes a surrogate threshold, not the proximal regularization parameter in Algorithm 1.

Table D.3: Abstention datasets and nominal class-conditional error targets $\alpha_0 = \alpha_1 = \alpha$. Counts refer to the full experimental samples; architectures and feature processing are described in the text.

Dataset	Observations	Class 0/class 1	Raw variables	α
Gaussian	2,000	1,000/1,000	8	0.15
ClimSim rainfall	11,373	10,825/548	124	0.05
NASA rainfall	49,302	47,073/2,229	6	0.10
Credit default	23,864	21,236/2,628	11	0.20

The surrogate thresholds β_j were calibrated on validation data to move the class-conditional errors toward the targets α_j. Training enforced

$$\widehat{R}_{\phi,j}(\theta_j) \le \beta_j + \epsilon_j, \qquad \epsilon_j = \frac{C}{\sqrt{n_j}}.$$

The shared slack constant C was selected using validation data and fixed at $C = 0.5$ for the reported experiments. Satisfying this surrogate constraint does not guarantee that the held-out class-conditional error is at most α_j. In particular, both methods have about 9% class-0 test error on ClimSim despite the nominal 5% target; the class-1 errors on credit default are also slightly above the nominal 20% target. Moreover, the different calibrated thresholds used by the two optimizers on ClimSim and NASA mean that they are compared at similar predictive tradeoffs rather than on identical constrained problems.

Numerical implementation. For the One Projection updates, the two residuals are combined as

$$g_{\max}(\theta) = \max_{j\in\{0,1\}} \{\widehat{R}_{\phi,j}(\theta_j) - \beta_j - \epsilon_j\}.$$

The projection routines enforce the two inequalities individually. For linear scores, Euclidean projection onto the empirical feasible set is a convex problem and is solved to numerical tolerance. For neural networks, an augmented-Lagrangian solver approximately minimizes $\frac{1}{2}\|\theta - \bar{\theta}\|^2$ subject to the empirical constraints. It seeks a feasible point near $\bar{\theta}$ but does not certify a globally nearest feasible point. One Projection uses independent minibatches to estimate the ambiguity objective and each class-conditional constraint gradient. Projected SGD uses an objective minibatch followed by a full-training-set feasibility projection after every parameter update. Thus these experiments evaluate a practical implementation: the hinge and maximum constraints can be nonsmooth, constraint

Table D.4: Abstention calibration and supplementary test accuracy. Thresholds are selected on validation data; accuracy is reported as mean ± sample standard deviation over ten runs.

Dataset	Method	(β_0, β_1)	Decided accuracy (%)
Gaussian	One Projection	$(0.300, 0.300)$	84.95 ± 2.14
	Projected SGD	$(0.300, 0.300)$	84.77 ± 1.96
ClimSim	One Projection	$(0.180, 0.290)$	91.69 ± 0.60
	Projected SGD	$(0.140, 0.225)$	91.96 ± 0.98
NASA rainfall	One Projection	$(0.260, 0.210)$	89.35 ± 0.39
	Projected SGD	$(0.260, 0.280)$	89.46 ± 0.44
Credit default	One Projection	$(0.425, 0.400)$	79.77 ± 1.73
	Projected SGD	$(0.425, 0.400)$	78.78 ± 1.58

estimates are stochastic, and the neural-network projection is approximate. The exact constraint-oracle and projection assumptions of the nonconvex-constraint theorem are not verified by these experiments.

Synthetic Gaussian. For each seed, we generated 2,000 observations from two balanced eight-variable Gaussian classes, with class separation 2.0, identity-scaled covariance, and 2% label noise. The split contains 1,200 training, 400 validation, and 400 test observations. We use two independent linear scores with intercepts.

ClimSim rainfall. The 11,373 observations were split chronologically into 6,844 training, 2,257 validation, and 2,272 test observations. Class 1 denotes a rain rate above the training-set 95th percentile. Each score is an independent $[32, 16]$ Softplus MLP with 248 inputs: 124 climate variables and their squares.

NASA rainfall. The 49,302 observations were divided using a seeded, stratified 60/20/20 random split: 29,581 training, 9,860 validation, and 9,861 test observations. The available data artifact contains no dates, so the split is not chronological. Each score is an independent $[8]$ Softplus MLP with 27 degree-two features derived from six NASA variables.

Credit default. The 23,864 labeled observations were divided using a seeded, stratified 60/20/20 random split: 14,318 training, 4,772 validation, and 4,774 test observations. Default is class 1. Each score is an independent $[16, 8]$ ReLU MLP with 11 inputs after a Yeo–Johnson transformation fitted on the training data.

Optimization and timing details. All experiments used minibatches of size 64 and 500 parameter updates. One Projection used $T = 20$ outer iterations and $K = 25$ inner iterations, with proximal coefficient $\ell = 0.1$, penalty strength 5.0, and the last outer iterate. Its joint parameter-ball radius was 2 for the linear Gaussian model and 10 for the MLP models. The selected One Projection/Projected SGD learning rates were $0.05/0.05$ for Synthetic Gaussian, $0.05/0.05$ for ClimSim, $0.05/0.03$ for NASA, and $0.10/0.03$ for credit default.

The linear Gaussian experiment used the exact empirical-risk projector with feasibility tolerance 10^{-5}, at most 30 bisection steps, and at most 500 dual steps. The MLP experiments used the approximate augmented-Lagrangian projector with learning rate 0.01, 150 inner iterations per multiplier round, six multiplier rounds, initial penalty 1, penalty growth factor 5, maximum penalty 10^6, 25 feasible-refinement iterations, feasibility tolerance 10^{-5}, and boundary-refinement tolerance 10^{-4}.

Runs were deterministic and CPU-only, using an Intel Core i7-7820X, Python 3.11.9, and PyTorch 2.13.0. Reported training runtime begins with the optimizer's parameter updates and ends after its final training projection. It includes all bounding-set and feasible-set projections, but excludes data loading and preprocessing, feasible initialization, post-training evaluation, figure generation, and artifact serialization.

Additional training curves. The ClimSim curves appear in Figure 2 in the main text. Figures 5, 6, and 7 below give the corresponding results for Gaussian, NASA rainfall, and credit default data. All four figures show the product surrogate objective and class-conditional surrogate risks. Points and colored bands report the mean and one sample standard deviation across ten runs. One Projection checkpoints are outer-loop outputs, with the last point evaluated after the terminal projection; Projected SGD is recorded at the corresponding cumulative update counts. Dotted lines show the validation-calibrated thresholds β_j, separately for each optimizer where these differ. Gray bands show $\beta_j \pm \epsilon_j$ with $\epsilon_j = 0.5/\sqrt{n_j}$; they are reference ranges, not confidence intervals. The enforced empirical constraint is $\widehat{R}_{\phi,j} \leq \beta_j + \epsilon_j$. One Projection permits intermediate constraint violation and applies the feasible-set projection only at the end.

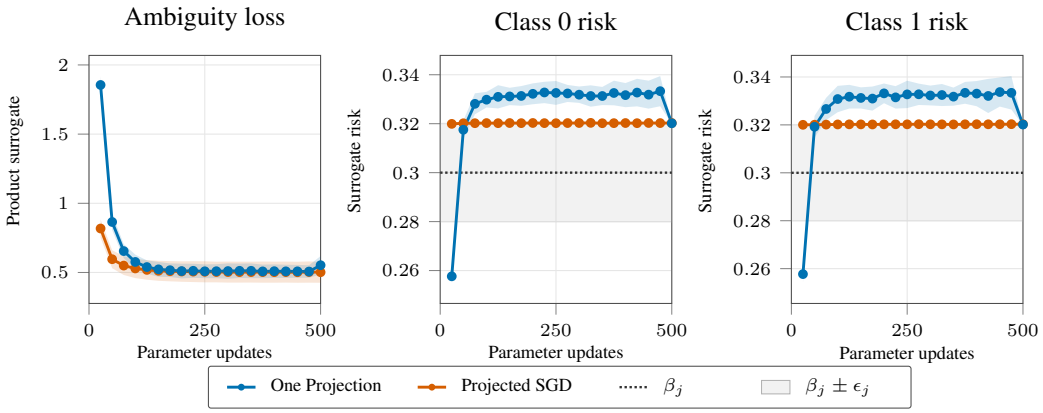

Figure 5: **Synthetic Gaussian Abstention.** Training product surrogate loss and class-specific surrogate risks. Points show means and colored bands show one sample standard deviation over ten runs. One Projection is recorded at outer-loop outputs, with its final point after projection; Projected SGD is recorded at the corresponding cumulative update counts.

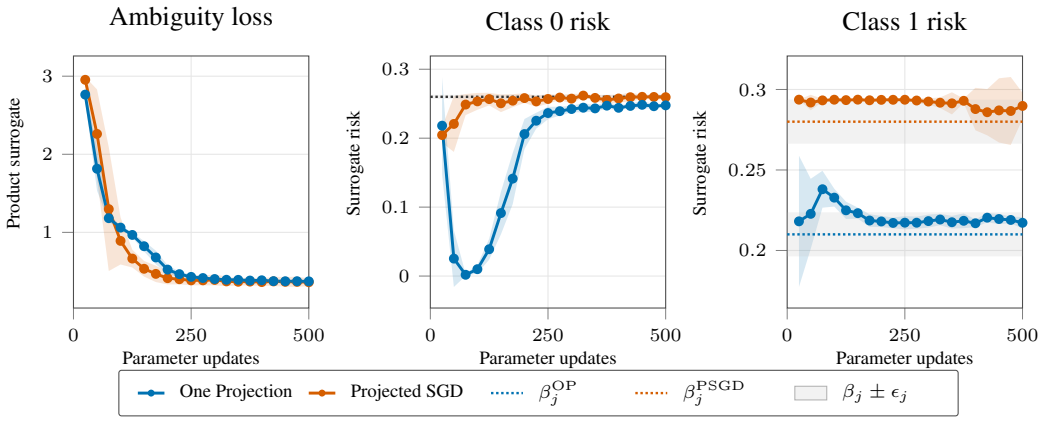

Figure 6: **NASA Rainfall Abstention.** Training product surrogate loss and class-specific surrogate risks. Points show means and colored bands show one sample standard deviation over ten runs. One Projection is recorded at outer-loop outputs, with its final point after projection; Projected SGD is recorded at the corresponding cumulative update counts.

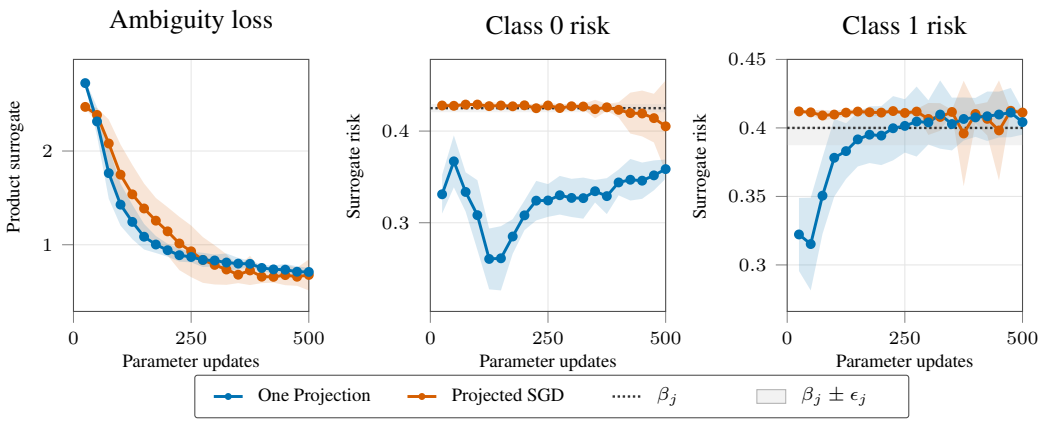

Figure 7: **Credit Default Abstention.** Training product surrogate loss and class-specific surrogate risks. Points show means and colored bands show one sample standard deviation over ten runs. One Projection is recorded at outer-loop outputs, with its final point after projection; Projected SGD is recorded at the corresponding cumulative update counts.