Gradient-Based Bilevel Optimization for Principal–Agent Contract Design

Aarya Bookseller
Department of Computer Science
Texas A&M University
aaryabookseller@tamu.edu

Tomer Galanti
Department of Computer Science
Texas A&M University
galanti@tamu.edu

Korok Ray
Mays Business School
Texas A&M University
korok@mays.tamu.edu

Abstract

We study a bilevel *max–max* optimization framework for principal–agent contract design, where a principal selects incentives to maximize utility while anticipating the agent's best response. This setting, central in moral hazard and contract theory, arises in applications such as market design, delegated portfolio management, hedge fund fee design, and executive compensation. While some special cases admit closed-form solutions, realistic contracts involve nonlinear utilities, stochastic dynamics, and high-dimensional actions for which analytical solutions do not exist.

We remove this dependence on closed forms by introducing a scalable framework that solves general principal–agent problems without restrictive assumptions. Our approach adapts *machine learning* techniques for bilevel optimization—specifically, implicit differentiation with conjugate gradient (CG)—to compute hypergradients using only Hessian–vector products, avoiding explicit Hessian inversion and scaling to high-dimensional contracts. Applied to the classic Holmström–Milgrom model, the method recovers the exact analytical optimum and converges reliably from random initialization. Because it is matrix-free and problem-agnostic, it extends directly to complex, nonlinear principal–agent models, providing a new computational tool for contract design, specifically in financial markets.

1 Introduction

The design of incentive mechanisms is a central problem in economics, finance, and operations research [9, 27, 30, 24, 10, 26]. In many settings, a *principal* (e.g., an employer, regulator, firm, or portfolio manager) seeks to influence the actions of an *agent* (e.g., an employee, contractor, or service provider) whose choices directly affect the principal's payoff [17]. A key challenge is that the principal cannot directly dictate the agent's decision; instead, they must offer a *contract* specifying how the agent will be compensated based on observable outcomes [36, 20, 21]. The agent, upon observing the contract, chooses an action that maximizes their own utility, which may diverge from that of the principal.

This interaction leads naturally to a *bilevel optimization* problem in which both levels are maximization problems: the principal optimizes contract parameters in the outer problem, anticipating the agent's best-response in the inner problem [12]. Such models appear throughout the literature on *moral hazard* [20, 17, 21], mechanism design [31], and industrial organization.

39th/ Conference on Neural Information Processing Systems (NeurIPS 2025).

A canonical example is the *linear–quadratic principal–agent model* of Holmström and Milgrom [21], in which the principal offers a linear contract $t = (s, b)$, consisting of a fixed payment s and a performance-based incentive b. The agent chooses an effort level a that is costly to exert, and output is noisy. Under quadratic cost of effort and mean–variance preferences, this model admits a closed-form solution for $(s^\star, b^\star, a^\star)$ [21]. While this structure is analytically tractable, real-world contracts often involve nonlinear utilities, richer stochastic dynamics, and high-dimensional actions for which closed-form solutions do not exist [37].

When u_1 and u_2 are differentiable, a natural computational strategy is to use gradient-based optimization [13]. However, in bilevel settings the outer objective $u_1(a^\star(t), t)$ depends on the contract parameters t both directly and indirectly through the agent's optimal response $a^\star(t)$. Differentiating through the inner maximization requires computing *hypergradients* that involve inverting the Hessian of u_2 with respect to the agent's action, a computational bottleneck in high dimensions [16].

Contributions. We introduce a scalable, gradient-based framework for principal–agent contract design that adapts *implicit differentiation* with conjugate gradient (CG) [18, 39, 8, 16, 28]. Although CG-based implicit differentiation is a well-established tool in large-scale machine learning—with applications in meta-learning and hyperparameter optimization [29, 33]—it has not previously been applied to principal–agent models, where analytical derivations remain the norm. Our method computes hypergradients entirely through Hessian–vector products, eliminating the need for explicit Hessian inversion and scaling effectively to nonlinear utilities and high-dimensional contract spaces. We test our method on classic moral-hazard settings and show that, except in extreme cases, it recovers the exact closed-form optimum from random initialization, validating both its correctness and stability. More broadly, the framework extends seamlessly to settings where closed-form analysis is infeasible, offering a general-purpose computational tool that connects modern machine learning optimization techniques with long-standing challenges in contract theory [9, 36].

2 Problem Setup

We consider the problem of *contract design* in a principal–agent relationship [17, 20]. In this setting, a *principal* (e.g., an employer, a firm, or a regulator) wishes to design an incentive scheme that will influence the behavior of an *agent* (e.g., an employee, a contractor, or a service provider).

The interaction has four key features: **(a)** the principal cannot directly dictate the agent's decision a (e.g., the level of effort exerted); **(b)** instead, the principal offers a *contract* t specifying a fixed payment and performance-based incentives; **(c)** the agent observes t and chooses a to maximize their own utility $u_2(a, t)$; and **(d)** the principal's utility $u_1(a, t)$ depends on both the contract t and the resulting action a. This leads to a *bilevel max–max problem*:

$$\max_{t \in \mathbb{R}^m} \quad u_1\left(a^\star(t), t\right) \tag{1}$$

$$\text{s.t.} \quad a^\star(t) \in \arg\max_{a \in \mathbb{R}^n} \; u_2(a, t), \tag{2}$$

where t are contract parameters, a agent actions, and u_1, u_2 principal and agent utilities.

This type of bilevel max–max problem arises in *economics*, *finance*, and *contract theory*, where the goal is to design incentives that align an agent's actions with the principal's objectives. In asset management a principal (fund owner) may set a fixed fee s and performance–based rate b to maximize returns net of payouts, anticipating that the agent (portfolio manager) will choose a risk exposure a to maximize utility given market volatility σ, risk aversion r, and cost of risk-taking c.

In simple cases, the structure of u_1 and u_2 allows for a *closed-form solution*. However, outside such highly structured quadratic settings, closed-form solutions generally *do not exist*. Realistic principal–agent models may involve nonlinear production functions, multidimensional effort vectors, stochastic dynamics, or additional constraints. In these cases, (1)–(2) must be solved numerically, and the challenge becomes: ***How can we design efficient optimization algorithms to solve general bilevel max–max problems when no analytical solution is available?***

3 Method

When both the outer and inner objective functions are differentiable, a natural approach to solving the bilevel problem (1)–(2) is to apply gradient-based optimization. In this setting, the outer objective

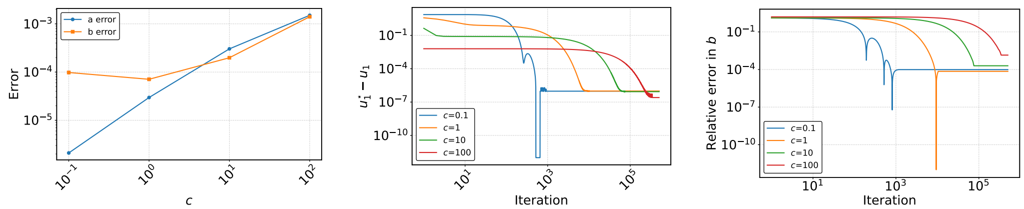

Figure 1: **Results for the Holmström–Milgrom linear–quadratic model with varying c. (Left)** Final relative errors in a and b. **(Middle)** Principal's utility gap during optimization. **(Right)** Relative error in b during optimization. Fixed parameters: $r = 1.0$, $\sigma = 0.1$; varied parameter: $c \in \{0.1, 1.0, 10.0, 100.0\}$.

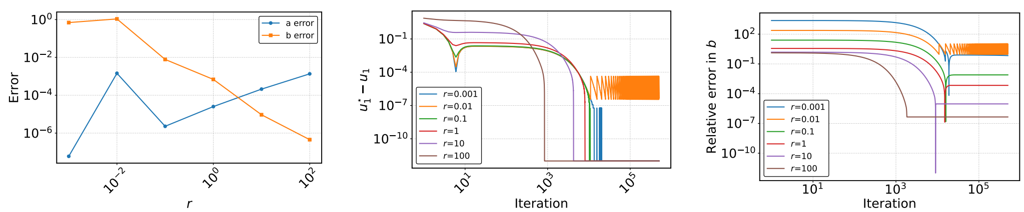

Figure 2: **Results with the Insurance Prevention model when varying r.** Fixed values: $c = 2.0$, $\sigma = 0.25$, $\ell = 1.0$. Varied: $r \in \{0.001, 0.01, 0.1, 1.0, 10.0, 100.0\}$.

$u_1(a^\star(t), t)$ depends on t both directly and indirectly through the agent's optimal response $a^\star(t)$. Thus, computing the gradient $\nabla_t u_1(a^\star(t), t)$ requires differentiating *through* the solution of the inner problem (2), which is typically unavailable in closed form and may itself require iterative optimization.

Theoretical derivation. Suppose $a^\star(t)$ denotes an exact maximizer of the inner problem. Then it satisfies the first-order optimality condition

$$\nabla_a u_2(a^\star(t), t) \;=\; 0. \tag{3}$$

By the implicit function theorem,

$$\frac{\mathrm{d}a^\star(t)}{\mathrm{d}t} \;=\; -\left[\nabla^2_{aa} u_2(a^\star(t), t)\right]^{-1} \nabla^2_{ta} u_2(a^\star(t), t). \tag{4}$$

Substituting this into the total derivative of u_1 yields the *hypergradient*:

$$\nabla_t u_1(a^\star(t), t) \;=\; \partial_t u_1(a^\star(t), t) - \partial_a u_1(a^\star(t), t)\left[\nabla^2_{aa} u_2(a^\star(t), t)\right]^{-1} \partial^2_{ta} u_2(a^\star(t), t). \tag{5}$$

Practical approximation. In practice, we do not have access to the exact optimum $a^\star(t)$. Instead, our algorithm computes an approximate best response $\tilde{a}(t)$ by running a limited number of gradient ascent steps on $u_2(\cdot, t)$. All quantities in (5) are therefore evaluated at $\tilde{a}(t)$ rather than $a^\star(t)$. Moreover, directly forming and inverting $\nabla^2_{aa} u_2$ is computationally infeasible for high-dimensional actions a. To avoid explicit inversion, we introduce an auxiliary vector v defined as the solution to

$$\left[-\nabla^2_{aa} u_2(\tilde{a}(t), t) + \lambda I\right] v \;=\; \nabla_a u_1(\tilde{a}(t), t), \tag{6}$$

where $\lambda > 0$ is a small damping parameter for numerical stability. Once v is obtained (approximately, using conjugate gradient), the hypergradient is estimated as

$$\text{hypergrad}_t \;=\; \nabla_t u_1(\tilde{a}(t), t) + \nabla^2_{ta} u_2(\tilde{a}(t), t)^\top v. \tag{7}$$

Practical implementation. The method is efficient because Hessian–vector products (HVPs) such as $\nabla^2_{aa} u_2\, v$ and $\nabla^2_{ta} u_2^\top v$ can be computed with reverse-on-forward automatic differentiation, avoiding explicit Hessian construction. We solve (6) approximately using T_{cg} iterations of conjugate gradient (CG) (see Alg. 2) [18, 39, 8, 28, 16], which requires only HVPs. The cost of each outer iteration is therefore: (i) the inner optimization for $a^\star(t)$, (ii) T_{cg} HVPs for solving (6), and (iii) one mixed HVP for $\nabla^2_{ta} u_2^\top v$. In practice, $T_{\text{cg}} \ll \dim(a)$ is sufficient for accurate hypergradients, and the method scales without forming or storing Hessians.

Algorithm 1 Bilevel Max–Max via Implicit Differentiation (CG) with Tolerances

Require: Utilities u_1, u_2; initial contract t_0; initial action a_0
Require: Inner steps T_{inner}, inner step size η_{inner}, inner tolerance ε_{in}
Require: Outer steps T_{outer}, outer step size η_{outer}
Require: Damping $\lambda \geq 0$, CG iterations T_{cg}, CG tolerance ε_{cg}

$t \leftarrow t_0, \quad a \leftarrow a_0$
for $k = 0$ to $T_{\text{outer}} - 1$ **do** ▷ outer loop: update contract t
  (Inner solve: approximate best response)
  for $j = 1$ to T_{inner} **do**
    $g_a^{(u_2)} \leftarrow \nabla_a u_2(a, t)$
    if $\|g_a^{(u_2)}\|_2 \leq \varepsilon_{\text{in}}$ **then break**
    $a \leftarrow a + \eta_{\text{inner}}\, g_a^{(u_2)}$ ▷ gradient *ascent* on $u_2(\cdot, t)$
  $\tilde{a}(t) \leftarrow a$ ▷ store approximate inner maximizer

  (Hypergradient via implicit differentiation + CG)
  $g_a^{(u_1)} \leftarrow \nabla_a u_1(\tilde{a}(t), t), \quad g_t^{(u_1)} \leftarrow \nabla_t u_1(\tilde{a}(t), t)$
  Define $hvp(v) \leftarrow -\nabla^2_{aa} u_2(\tilde{a}(t), t)\, v + \lambda v$ ▷ SPD operator for CG
  $v \leftarrow \text{CG}\big(\text{hvp},\, g_a^{(u_1)},\, T_{\text{cg}},\, \varepsilon_{\text{cg}}\big)$
  $m \leftarrow \nabla^2_{ta} u_2(\tilde{a}(t), t)^\top v$
  $\text{hypergrad}_t \leftarrow g_t^{(u_1)} + m$

  (Outer ascent step)
  $t \leftarrow t + \eta_{\text{outer}}\, \text{hypergrad}_t$
  (Warm start) $a \leftarrow \tilde{a}(t)$
return t, a ▷ final contract and induced (approximate) action

4 Experiments

Our goal is to test whether the proposed implicit differentiation method with conjugate gradient (CG) can reliably recover optimal contract parameters in bilevel max–max principal–agent problems. We benchmark performance in five canonical CARA–Normal linear–contract environments with known closed-form solutions: **(1)** the Holmström–Milgrom linear–quadratic benchmark [20], **(2)** insurance with prevention (self–protection)[14, 38], **(3)** imperfect performance measurement with a single noisy signal[26], **(4)** multiple noisy signals (two-signal aggregation)[26, 6], and **(5)** separable multitask contracting[22]. These settings allow us to compare learned contracts and induced actions against theoretical optima and to assess the numerical behavior of implicit differentiation in a controlled environment (see Appendix C.1).

In each experiment, all environment parameters (e.g., c, σ) are fixed while a single parameter of interest (e.g., r) is varied. Contract parameters $b = t$ and the agent's action a are initialized independently from standard normal distributions.

We solve the bilevel problem (1)–(2) using Alg. 1. The inner problem is optimized by gradient ascent on u_2 with step size $\eta_{\text{inner}} = 5 \times 10^{-3}$ for up to $T_{\text{inner}} = 50$ iterations, stopping early when $\|\nabla_a u_2\|_2 \leq \varepsilon_{\text{inner}} = 10^{-4}$. The outer variables t are updated for $T_{\text{outer}} = 500{,}000$ iterations with step size $\eta_{\text{outer}} = 10^{-3}$, using implicit differentiation with CG ($T_{\text{cg}} = 20$, damping $\lambda = 10^{-4}$). Each inner solve is warm-started from the previous $\tilde{a}(t)$.

We evaluate performance using three metrics: the relative errors in the agent's action and contract parameter, $\text{error}_a := \frac{\|a - a^\star\|}{\|a^\star\|}$, $\text{error}_b := \frac{\|b - b^\star\|}{\|b^\star\|}$ and the principal's utility gap, $\Delta u_1 = u_1(a^\star, b^\star) - u_1(a, b)$, (a, b) are the learned action and contract parameters, and $(a^\star, b^\star)$ are their closed-form counterparts with $a^\star = a^\star(b^\star)$. Across all experiments, the algorithm consistently recovers the optimal contract parameter $b^\star$ and achieves a negligible utility gap, along with the corresponding optimal action $a^\star(b^\star)$. Performance degrades only in extreme cases (e.g., when $r \ll 0.1$, as in Fig. 3), which correspond to economically unrealistic regimes where the varied parameter (such as risk aversion r) takes on vanishingly small or excessively large values.

References

[1] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter. Differentiable convex optimization layers. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

[2] V. Aguirregabiria and P. Mira. Sequential estimation of dynamic discrete games. *Econometrica*, 75(1):1–53, 2007.

[3] V. Aguirregabiria and P. Mira. Dynamic discrete choice structural models: A survey. *Journal of Econometrics*, 156(1):38–67, 2010. Structural Models of Optimization Behavior in Labor, Aging, and Health.

[4] B. Amos and J. Z. Kolter. OptNet: Differentiable optimization as a layer in neural networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 136–145. PMLR, 06–11 Aug 2017.

[5] F. Bacchiocchi, M. Castiglioni, A. Marchesi, and N. Gatti. *Regret Minimization for Piecewise Linear Rewards: Contracts, Auctions, and Beyond*, page 1020. Association for Computing Machinery, New York, NY, USA, 2025.

[6] G. P. Baker. Incentive contracts and performance measurement. *Journal of Political Economy*, 100(3):598–614, 1992.

[7] J. F. Bard. Bilevel programming in management. In C. Floudas and P. Pardalos, editors, *Encyclopedia of Optimization*. Springer, Boston, MA, 2008.

[8] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. A. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Other Titles in Applied Mathematics. SIAM, Philadelphia, PA, 1994.

[9] P. Bolton and M. Dewatripont. *Contract Theory*. MIT Press, 2005.

[10] G. P. Cachon. Supply chain coordination with contracts. In *Handbooks in Operations Research and Management Science*, volume 11. Elsevier, 2003.

[11] P.-A. Chiappori and B. Salanié. Testing for asymmetric information in insurance markets. *Journal of Political Economy*, 108(1):56–78, 2000.

[12] B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Annals of Operations Research*, 153(1):235–256, Sept. 2007.

[13] J. Domke. Generic methods for optimization-based modeling. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, pages 318–326. PMLR, 2012.

[14] I. Ehrlich and G. S. Becker. Market insurance, self-insurance, and self-protection. *Journal of Political Economy*, 80(4):623–648, 1972.

[15] G.-L. Gayle and R. A. Miller. Identifying and testing models of managerial compensation. *The Review of Economic Studies*, 82(3):1074–1118, 2015.

[16] S. Gould, B. Fernando, A. Cherian, P. Anderson, R. S. Cruz, and E. Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *ArXiv*, abs/1607.05447, 2016.

[17] S. J. Grossman and O. D. Hart. An analysis of the principal-agent problem. *Econometrica*, 51(1):7–45, 1983.

[18] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[19] C.-J. Ho, A. Slivkins, and J. W. Vaughan. Adaptive contract design for crowdsourcing markets: bandit algorithms for repeated principal-agent problems. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14, page 359–376, New York, NY, USA, 2014. Association for Computing Machinery.

[20] B. Holmström. Moral hazard and observability. *The Bell Journal of Economics*, 10(1):74–91, 1979.

[21] B. Holmström and P. Milgrom. Aggregation and linearity in the provision of intertemporal incentives. *Econometrica*, 55(2):303–328, 1987.

[22] B. Holmström and P. Milgrom. Multitask principal-agent analyses: Incentive contracts, asset ownership, and job design. *Journal of Law, Economics, & Organization*, 7:24–52, 1991.

[23] V. J. Hotz and R. A. Miller. Conditional choice probabilities and the estimation of dynamic models. *The Review of Economic Studies*, 60(3):497–529, 1993.

[24] M. C. Jensen and W. H. Meckling. Theory of the firm: Managerial behavior, agency costs and ownership structure. *Journal of Financial Economics*, 3(4):305–360, 1976.

[25] S. M. Kakade, I. Lobel, and H. Nazerzadeh. Optimal dynamic mechanism design and the virtual-pivot mechanism. *Operations Research*, 61(4):837–854, 2013.

[26] J.-J. Laffont and D. Martimort. *The Theory of Incentives: The Principal–Agent Model*. Princeton University Press, 2002.

[27] E. P. Lazear and M. Gibbs. *Personnel Economics in Practice*. John Wiley & Sons, 3 edition, 2014.

[28] J. Lorraine, P. Vicol, and D. Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1540–1552. PMLR, 2020.

[29] D. Maclaurin, D. Duvenaud, and R. P. Adams. Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2113–2122. PMLR, 2015.

[30] P. Milgrom and J. Roberts. *Economics, Organization and Management*. Prentice Hall, 1992.

[31] R. B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[32] A. Pavan, I. Segal, and J. Toikka. Dynamic mechanism design: A myersonian approach. *Econometrica*, 82(2):601–653, 2014.

[33] A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[34] M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4334–4343. PMLR, 10–15 Jul 2018.

[35] J. Rust. Optimal replacement of gmc bus engines: An empirical model of harold zurcher. *Econometrica*, 55(5):999–1033, September 1987.

[36] B. Salanié. *The Economics of Contracts: A Primer*. MIT Press, 3 edition, 2017.

[37] Y. Sannikov. A continuous-time version of the principal–agent problem. *The Review of Economic Studies*, 75(3):957–984, 2008.

[38] S. Shavell. On moral hazard and insurance. *The Quarterly Journal of Economics*, 93(4):541–562, 1979.

[39] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, August 1994. Technical report.

[40] N. Williams. Persistent private information. *Econometrica*, 79(4):1233–1275, 2011.

[41] B. Zhu, S. Bates, Z. Yang, Y. Wang, J. Jiao, and M. I. Jordan. The sample complexity of online contract design. In *Proceedings of the 24th ACM Conference on Economics and Computation*, EC '23, page 1188, New York, NY, USA, 2023. Association for Computing Machinery.

Algorithm 2 Conjugate Gradient (CG) Solver

Require: `hvp`: function computing Hv for given v; right-hand side g; maximum iterations T_{cg}; tolerance tol

$v \leftarrow 0$
$r \leftarrow g - \texttt{hvp}(v)$ ▷ initial residual
$p \leftarrow r$
$rs\_old \leftarrow r^\top r$
for $k = 0$ to $T_{\text{cg}} - 1$ **do**
  $Hp \leftarrow \texttt{hvp}(p)$
  $\alpha \leftarrow rs\_old/(p^\top Hp)$
  $v \leftarrow v + \alpha p$
  $r \leftarrow r - \alpha Hp$
  if $\|r\|_2 < \text{tol}$ **then break**
  $rs\_new \leftarrow r^\top r$
  $\beta \leftarrow rs\_new/rs\_old$
  $p \leftarrow r + \beta p$
  $rs\_old \leftarrow rs\_new$
return v

A Conjugate Gradient

The *conjugate gradient* (CG) method is an iterative algorithm for solving large, sparse, symmetric positive-definite linear systems of the form

$$Hv = g,$$

where H is typically the Hessian of the inner objective with respect to the agent's actions, and g is a gradient vector from the outer objective. In our framework, CG is used to avoid explicitly forming or inverting H, instead relying solely on *Hessian–vector products* (HVPs), which are computed efficiently via automatic differentiation. This matrix-free property makes CG suitable for high-dimensional principal–agent problems, since each iteration requires only one HVP and a few vector operations.

Usage in our method. In (6), CG approximately solves

$$\left[-\nabla^2_{aa} u_2(a^\star(t), t) + \lambda I\right] v \;=\; \nabla_a u_1(a^\star(t), t),$$

returning v without explicitly computing or storing $\nabla^2_{aa} u_2$. The resulting v is then used to form the hypergradient in Eq. (5).

B Related Work

Principal–agent theory and contract design. Classic insight into incentive contracts arises from hidden–action models of moral hazard [20, 17, 21] and the broader mechanism–design tradition [31]. These foundations have been synthesized in comprehensive works spanning labor economics, industrial organization, management, and finance [9, 26, 36, 27, 30, 7, 19]. In the canonical CARA–Normal, linear–contract framework, the Holmström–Milgrom linear–quadratic model offers closed-form solutions under analytical tractability [21]. Subsequent extensions have preserved such tractability by incorporating features like multitask incentives [22], imperfect or multiple performance measures [6, 26], and insurance with prevention trade-offs [14, 38]. While these classical models provide sharp comparative statics, they rest on restrictive assumptions—quadratic costs, Gaussian uncertainty, and linear contracts—that limit their applicability in nonlinear or high-dimensional settings.

Computation in economics and finance. When analytical derivations are unavailable, economists have relied on problem-specific computational fixes. In dynamic contracting, tractability often comes from recursive formulations that reduce the principal's problem to continuation utilities [37, 40, 32, 25]. Structural estimation methods likewise design bespoke algorithms for particular incentive models—ranging from nested fixed-point methods [35] and conditional choice probabilities [23] to sequential and simulation-based estimators for dynamic games [2, 3, 11, 15]. More recently, online learning perspectives recast contract design as a bandit problem, yielding adaptive discretization and regret-minimization guarantees [19, 41, 5]. Each of these approaches has advanced understanding in specific settings, but all share the same limitation: they depend on handcrafted derivations or strong structure, leaving open the question of how to compute contracts in more general nonlinear or high-dimensional environments.

Bilevel optimization. Abstractly, principal–agent models can be cast as bilevel *max–max* programs where the principal's optimization anticipates the agent's best response. Bilevel optimization has a long tradition in operations research [12] and has become central in machine learning—powering hyperparameter tuning, meta-learning, data reweighting, and differentiable programming [28, 33, 34, 4, 1]. Early methods differentiated through the inner solver by "unrolling" optimization steps [13], which is memory-intensive and prone to truncation bias. More scalable approaches apply implicit differentiation to the inner optimality conditions, reducing the task to solving linear systems involving Hessians or Jacobians [16, 28]. Despite their success in machine learning, such bilevel optimization tools have not been systematically applied to contract design.

Implicit differentiation and matrix-free methods. Modern implicit differentiation techniques avoid explicit Hessian inversion by combining automatic differentiation for Hessian–vector products (HVPs) with Krylov-style solvers like conjugate gradient (CG) [18, 8, 39]. This "matrix-free" recipe underpins recent advances in scalable hypergradient methods for meta-learning, hyperparameter optimization, and differentiable programming [29, 33, 16, 28]. Bringing this toolkit into principal–agent problems fills the gap left by recursive, structural, and bandit approaches: it provides a general-purpose, model-agnostic computational method that requires only gradient and HVP oracles.

Positioning and novelty. n this work, we adapt well-established implicit-differentiation techniques to the distinctive *max–max* structure of principal–agent problems, introducing ascent directions appropriate for inner maximizations and adaptive damping for numerical stability. Our framework (i) reproduces known analytic solutions in classical settings, (ii) scales to nonlinear utilities and high-dimensional contract spaces, and (iii) requires only gradient evaluations and Hessian–vector product oracles, without ever forming or inverting the Hessian explicitly. In this way, we bring a computational tool widely adopted in machine learning into contract design, showing that it provides a flexible and general alternative to existing specialized methods.

C Additional Results

To expand on the main-text experiments, we conducted a wide range of experiments with the settings outlined below.

C.1 Settings

We present a set of canonical hidden-action models in which a risk–neutral principal offers a linear contract to a risk–averse agent. In each environment, the agent privately chooses an action (e.g., effort or prevention) that influences an output or loss variable, which is observed imperfectly or with noise. Contracts take the form $t = (s, b, \ldots)$, where s is a fixed payment, and b (and possibly other coefficients) link compensation to the observed performance measure(s). The principal and the agent both have exponential utility with Constant Absolute Risk Aversion (CARA) preferences and face normally distributed shocks, yielding tractable certainty–equivalent formulations.

We review the Holmström–Milgrom linear–quadratic benchmark and several extensions capturing distinct informational and technological frictions: moral hazard in insurance with prevention, imperfect performance measurement, and relative performance evaluation. For each environment, we state the utility functions, solve for the agent's best response, impose the participation constraint, and derive closed-form optimal contract parameters $(a^\star, b^\star, s^\star, \ldots)$.

Holmström–Milgrom (linear–quadratic baseline). In this setting, a risk–neutral principal hires a risk–averse agent to produce output; the agent chooses effort a (cost $\frac{c}{2}a^2$), output is measured with noise $y = a + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, and pay is linear in measured performance, $w = s + by$. The principal's and agent's certainty–equivalent utilities are

$$\begin{aligned} u_1(a, t; r, \sigma, c) &= a - \tfrac{1}{2} r b^2 \sigma^2 - \tfrac{1}{2} c a^2, \\ u_2(a, t; r, \sigma, c) &= s + ba - \tfrac{1}{2} r b^2 \sigma^2 - \tfrac{1}{2} c a^2, \end{aligned} \tag{8}$$

where $r > 0$ is the agent's risk aversion, $c > 0$ is the effort–cost coefficient, and $\sigma > 0$ measures output variability.

Imposing the participation constraint $u_2(a^\star, t) = U_{\text{res}}$ yields the closed-form optimal contract:

$$\begin{aligned} b^\star &= \frac{1}{1 + rc\sigma^2}, \\ a^\star &= \frac{b^\star}{c}, \\ s^\star &= U_{\text{res}} - \Big[b^\star a^\star - \tfrac{1}{2} r (b^\star)^2 \sigma^2 - \tfrac{1}{2} c (a^\star)^2\Big]. \end{aligned} \tag{9}$$

As r or σ^2 rise, incentives become more costly and $b^\star$ decreases.

Insurance with prevention (self–protection). In this setting, a risk–neutral insurer offers coverage to a risk–averse insured who faces a baseline loss $\ell > 0$ and a zero-mean loss shock $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ with volatility $\sigma > 0$. Unobservable prevention a lowers expected loss linearly (from ℓ to $\ell - a$) at cost $\frac{c}{2} a^2$ with $c > 0$. The realized loss is $\tilde{L} = (\ell - a) + \varepsilon$. The contract is linear in the realized loss: the insured pays a fixed premium s and receives indemnity $b\tilde{L}$, so they bears the residual loss $(1 - b)\tilde{L}$. The principal (insurer) and agent (insured) certainty–equivalent utilities are

$$\begin{aligned} u_1(a, t; r, \sigma, c, \ell) &= -(\ell - a) - \tfrac{1}{2} r (1 - b)^2 \sigma^2 - \tfrac{1}{2} c a^2, \\ u_2(a, t; r, \sigma, c, \ell) &= -(1 - b)(\ell - a) - s - \tfrac{1}{2} r (1 - b)^2 \sigma^2 - \tfrac{1}{2} c a^2, \end{aligned} \tag{10}$$

where $r > 0$ is the insured's risk aversion, $c > 0$ is the prevention–cost coefficient, $\sigma > 0$ is the loss volatility, and $\ell > 0$ is baseline loss.

Imposing $u_2(a^\star, t) = U_{\text{res}}$ yields the optimal contract:

$$\begin{aligned} b^\star &= \frac{rc\sigma^2}{1 + rc\sigma^2}, \\ a^\star &= \frac{1}{c(1 + rc\sigma^2)}, \\ s^\star &= -U_{\text{res}} - (1 - b^\star)(\ell - a^\star) - \tfrac{1}{2} r (1 - b^\star)^2 \sigma^2 - \tfrac{1}{2} c (a^\star)^2. \end{aligned} \tag{11}$$

As risk or risk aversion r increases, coverage increases ($b^\star \uparrow$) and prevention decreases ($a^\star \downarrow$), reflecting moral hazard.

Imperfect performance measurement (CARA–Normal). We study a hidden-action setting in which performance is observed only through a noisy signal $z = \alpha a + \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and sensitivity $\alpha \in \mathbb{R}$. The contract is parameterized by $t = (s, b)$, where s is a fixed transfer and b is the slope on the signal. The principal's and agent's certainty–equivalent utilities are

$$\begin{aligned} u_1(a, t; r, \sigma, c, v) &= va \; - \; \tfrac{1}{2} r b^2 \sigma^2 \; - \; \tfrac{1}{2} c a^2, \\ u_2(a, t; r, \sigma, c, \alpha) &= s + b(\alpha a) \; - \; \tfrac{1}{2} r b^2 \sigma^2 \; - \; \tfrac{1}{2} c a^2, \end{aligned} \tag{12}$$

where $r > 0$ is risk aversion, $c > 0$ is the effort–cost coefficient, $\sigma > 0$ is signal noise, α is signal informativeness, and $v > 0$ is the principal's marginal value of effort.

Imposing the participation constraint $u_2(a^\star, t) = U_{\text{res}}$ yields the closed-form optimal contract:

$$\begin{aligned} b^\star &= \frac{v\alpha}{v\alpha^2 + rc\sigma^2}, \\ a^\star &= \frac{\alpha b^\star}{c}, \\ s^\star &= U_{\text{res}} - \Big[b^\star \alpha a^\star - \tfrac{1}{2} r (b^\star)^2 \sigma^2 - \tfrac{1}{2} c (a^\star)^2\Big]. \end{aligned} \tag{13}$$

As $\alpha \to 0$ (uninformative signal), incentives vanish ($b^\star \to 0$); as α rises or σ^2 falls, the optimal slope increases.

Relative performance (peer benchmark, CARA–Normal). Two agents face a common shock and idiosyncratic noise. For agent i, $y_i = a + \eta + \varepsilon_i$ and $y_j = a_{\text{peer}} + \eta + \varepsilon_j$, where $\varepsilon_i, \varepsilon_j \sim \mathcal{N}(0, \sigma^2)$ are

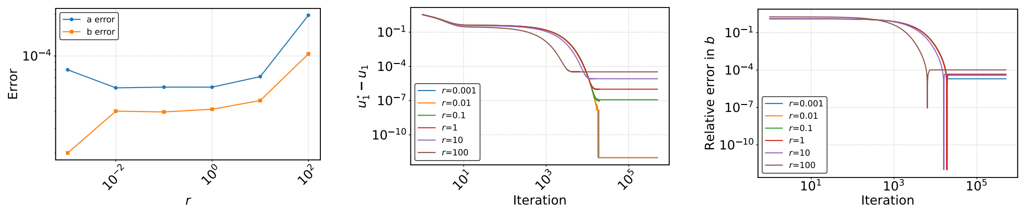

Figure 3: **Results with the Holmström–Milgrom linear–quadratic model when varying r.** Fixed values: $c = 2.0$, $\sigma = 0.1$. Varied: $r \in \{0.001, 0.01, 0.1, 1.0, 10.0, 100.0\}$.

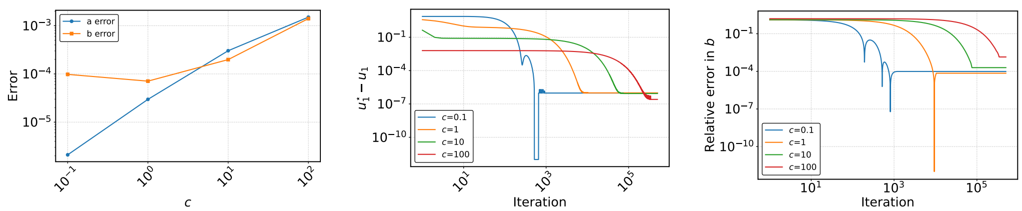

Figure 4: **Results with the Holmström–Milgrom linear–quadratic model when varying c.** Fixed values: $r = 1.0$, $\sigma = 0.1$. Varied: $c \in \{0.1, 1.0, 10.0, 100.0\}$.

independent and $\eta \sim \mathcal{N}(0, \tau^2)$ is common. The contract is $t = (s, b, d)$, with intercept s, own–slope b, and peer–slope d on $w_i = s + b\, y_i + d\, y_j$. The principal's and agent's certainty–equivalent utilities are

$$\begin{aligned} u_1(a, t; r, \sigma, \tau, c, v) &= va \;-\; \tfrac{1}{2}r\big[(b^2{+}d^2)(\sigma^2{+}\tau^2) + 2bd\,\tau^2\big] \;-\; \tfrac{1}{2}ca^2, \\ u_2(a, t; r, \sigma, \tau, c, a_{\text{peer}}) &= s + ba + da_{\text{peer}} \;-\; \tfrac{1}{2}r\big[(b^2{+}d^2)(\sigma^2{+}\tau^2) + 2bd\,\tau^2\big] \;-\; \tfrac{1}{2}ca^2. \end{aligned} \tag{14}$$

Imposing $u_2(a^\star, t) = U_{\text{res}}$ and optimizing yields

$$\begin{aligned} d^\star &= -\,b^\star \frac{\tau^2}{\sigma^2 + \tau^2}, \\ \sigma^2_{\text{eff}} &= \frac{\sigma^2(\sigma^2 + 2\tau^2)}{\sigma^2 + \tau^2}, \\ b^\star &= \frac{v}{v + rc\sigma^2_{\text{eff}}}, \\ a^\star &= \frac{b^\star}{c}, \\ s^\star &= U_{\text{res}} - \Big[b^\star a^\star + d^\star a_{\text{peer}} - \tfrac{1}{2}r\, b^{\star 2}\sigma^2_{\text{eff}} - \tfrac{1}{2}c\, a^{\star 2}\Big]. \end{aligned} \tag{15}$$

As the common-shock variance τ^2 rises, the optimal peer loading becomes more negative ($|d^\star| \uparrow$), residual risk per unit incentive increases ($\sigma^2_{\text{eff}} \uparrow$), and the optimal own–slope $b^\star$ declines.

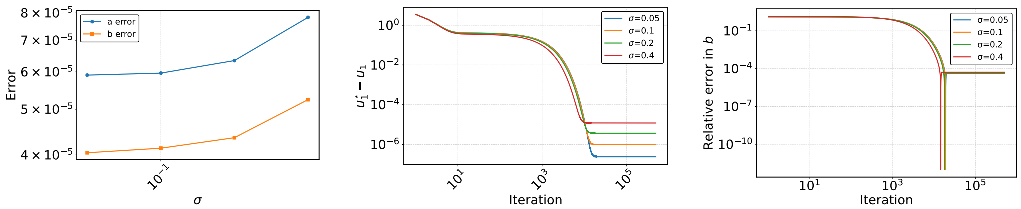

Figure 5: **Results with the Holmström–Milgrom linear–quadratic model when varying σ.** Fixed values: $r = 1.0$, $c = 2.0$. Varied: $\sigma \in \{0.05, 0.1, 0.2, 0.4\}$.

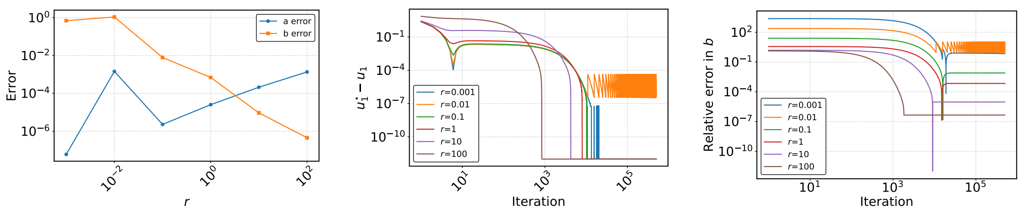

Figure 6: **Results with the Insurance Prevention model varying r. Fixed values: $c = 2.0$, $\sigma = 0.25$, and $\ell = 1.0$. Varied: $r \in \{0.001,\ 0.01,\ 0.1,\ 1.0,\ 10.0,\ 100.0\}$.**

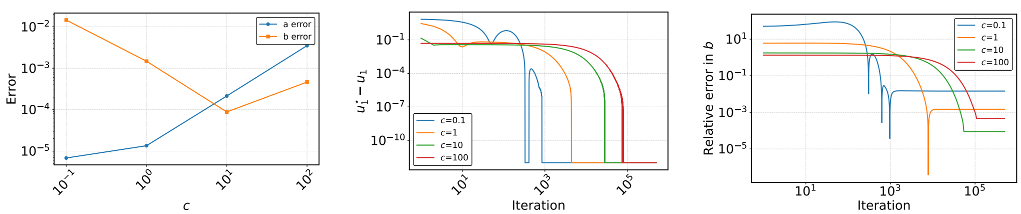

Figure 7: **Results with the Insurance Prevention model when varying c.** Fixed values: $r = 1.0$, $\sigma = 0.25$, $\ell = 1.0$. Varied: $c \in \{0.1, 1.0, 10.0, 100.0\}$.

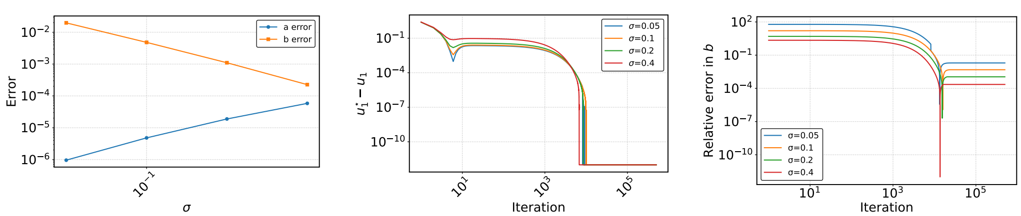

Figure 8: **Results with the Insurance Prevention model when varying σ.** Fixed values: $r = 1.0$, $c = 2.0$, $\ell = 1.0$. Varied: $\sigma \in \{0.05, 0.1, 0.2, 0.4\}$.

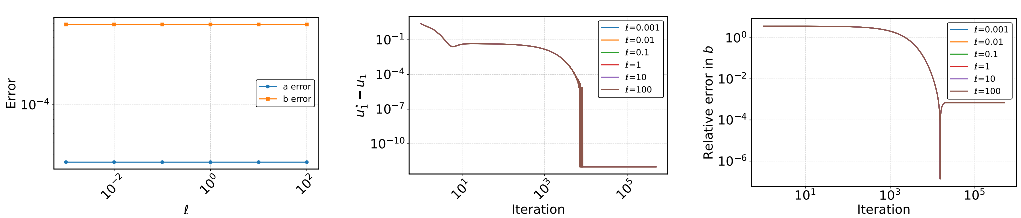

Figure 9: **Results with the Insurance Prevention model when varying ℓ.** Fixed values: $r = 1.0$, $c = 2.0$, $\sigma = 0.25$. Varied: $\ell \in \{0.001, 0.01, 0.1, 1.0, 10.0, 100.0\}$.

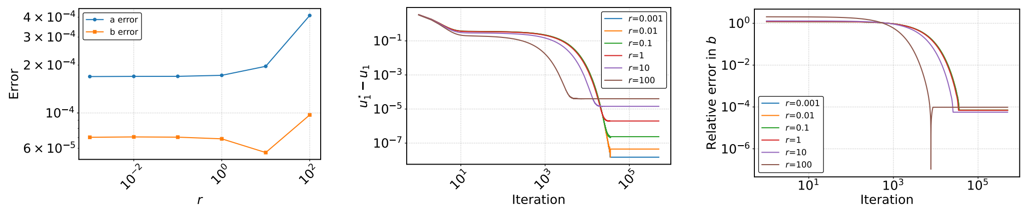

Figure 10: **Results with the Imperfect Signal model when varying r.** Fixed values: $c = 2.0$, $\sigma = 0.1$, $\alpha = 0.7$, $v = 1.0$. Varied: $r \in \{0.001, 0.01, 0.1, 1.0, 10.0, 100.0\}$.

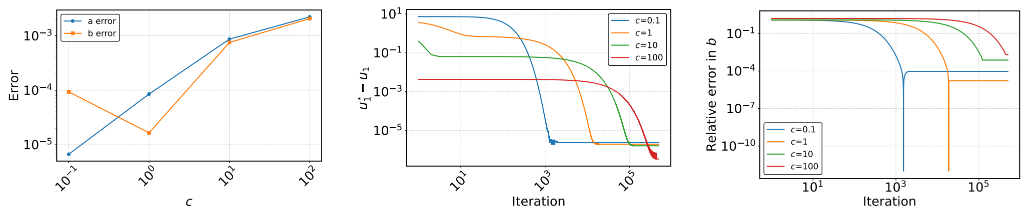

Figure 11: **Results with the Imperfect Signal model when varying c.** Fixed values: $r = 1.0$, $\sigma = 0.1$, $\alpha = 0.7$, $v = 1.0$. Varied: $c \in \{0.1, 1.0, 10.0, 100.0\}$.

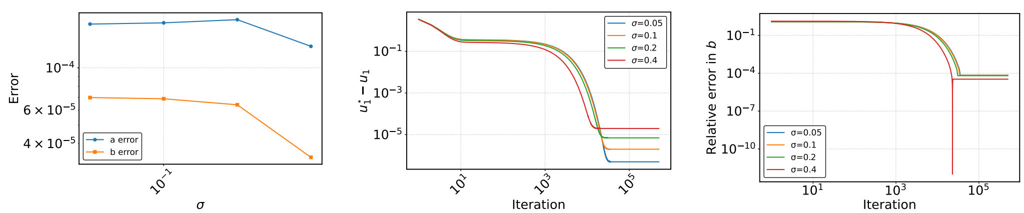

Figure 12: **Results with the Imperfect Signal model when varying σ.** Fixed values: $r = 1.0$, $c = 2.0$, $\alpha = 0.7$, $v = 1.0$. Varied: $\sigma \in \{0.05, 0.1, 0.2, 0.4\}$.

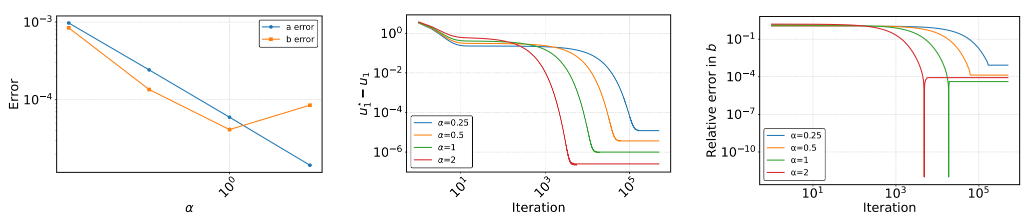

Figure 13: **Results with the Imperfect Signal model when varying α.** Fixed values: $r = 1.0$, $c = 2.0$, $\sigma = 0.1$, $v = 1.0$. Varied: $\alpha \in \{0.25, 0.5, 1.0, 2.0\}$.

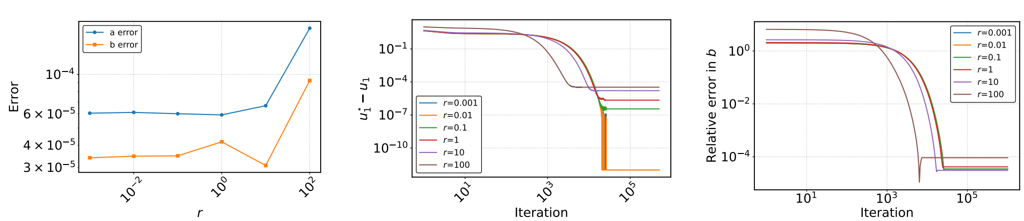

Figure 14: **Results with the Multitask Separable model when varying r.** Fixed values: $U_{\text{res}} = 0.0$, c_i, σ_i, and v_i at baseline values. Varied: $r \in \{0.001, 0.01, 0.1, 1.0, 10.0, 100.0\}$.

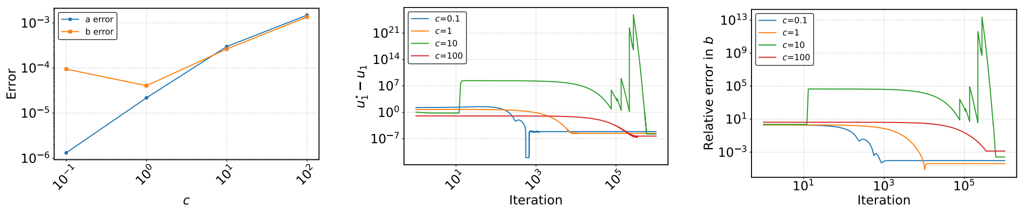

Figure 15: **Results with the Multitask Separable model when varying c_i.** Fixed values: $r = 1.0$, $U_{\text{res}} = 0.0$, σ_i and v_i at baseline values. Varied: $c_i \in \{0.1, 1.0, 10.0, 100.0\}$ for each task i.

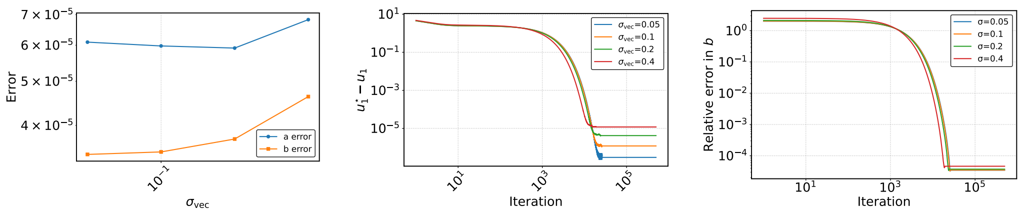

Figure 16: **Results with the Multitask Separable model when varying σ_i.** Fixed values: $r = 1.0$, $U_{\text{res}} = 0.0$, c_i and v_i at baseline values. Varied: $\sigma_i \in \{0.05, 0.1, 0.2, 0.4\}$ for each task i.

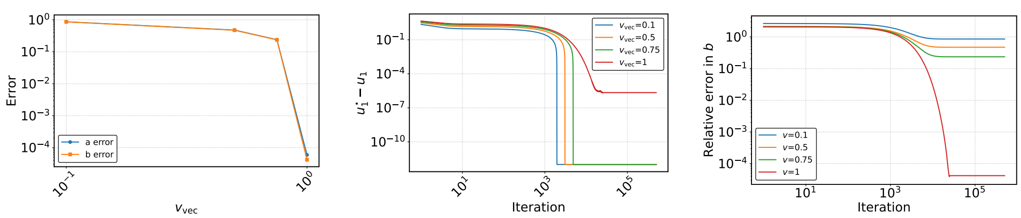

Figure 17: **Results with the Multitask Separable model when varying v_i.** Fixed values: $r = 1.0$, $U_{\text{res}} = 0.0$, c_i and σ_i at their baseline values. Varied: $v_i \in \{0.10, 0.5, 0.75, 1.0\}$ (for each task i).

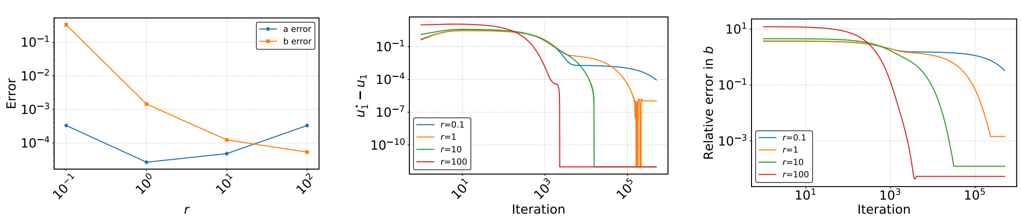

Figure 18: **Results with the Two Signals model when varying r.** Fixed values: $c = 2.0$, $v = 1.0$, $\sigma_1 = 0.15$, $\sigma_2 = 0.2$. Varied: $r \in \{0.1, 1.0, 10.0, 100.0\}$.

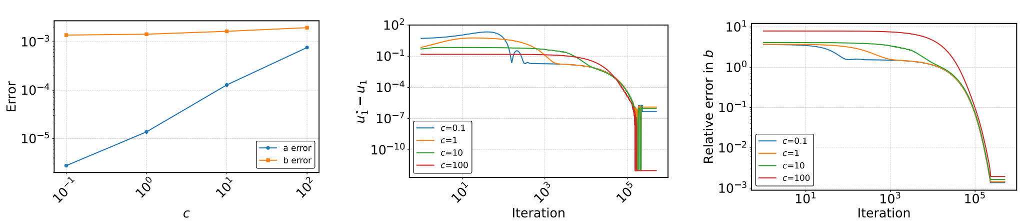

Figure 19: **Results with the Two Signals model when varying c.** Fixed values: $r = 1.0$, $v = 1.0$, $\sigma_1 = 0.15$, $\sigma_2 = 0.2$. Varied: $c \in \{0.1, 1.0, 10.0, 100.0\}$.

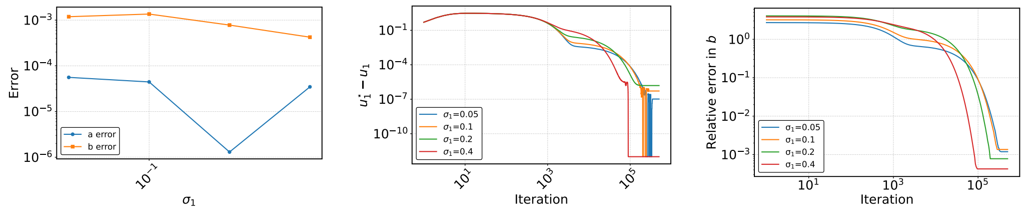

Figure 20: **Results with the Two Signals model when varying σ_1.** Fixed values: $r = 1.0$, $c = 2.0$, $v = 1.0$, $\sigma_2 = 0.2$. Varied: $\sigma_1 \in \{0.05, 0.1, 0.2, 0.4\}$.

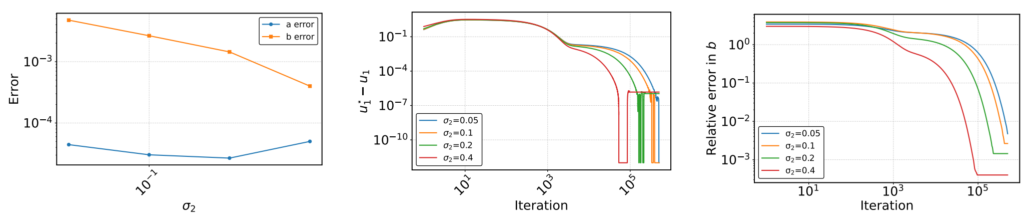

Figure 21: **Results with the Two Signals model when varying σ_2.** Fixed values: $r = 1.0$, $c = 2.0$, $v = 1.0$, $\sigma_1 = 0.15$. Varied: $\sigma_2 \in \{0.05, 0.1, 0.2, 0.4\}$.

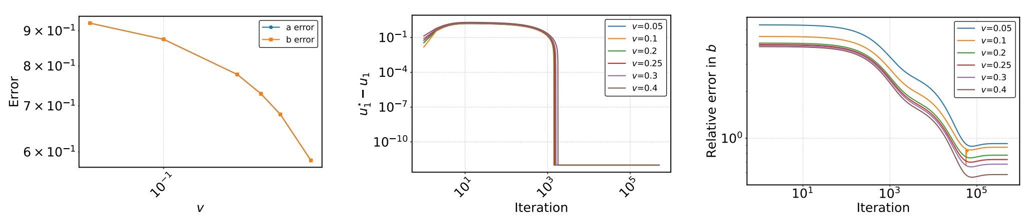

Figure 22: **Results with the Two Signals model when varying v.** Fixed values: $r = 1.0$, $c = 2.0$, $\sigma_1 = 0.15$, $\sigma_2 = 0.2$. Varied: $v \in \{0.05, 0.10, 0.20, 0.25, 0.30, 0.40\}$.