NExT-Game 2026: New Frontiers in Game-Theoretic Learning – ICML 2026 Workshop

Non-Linear Strategic Classification Made Practical

Jack Geary JACK.GEARY@ED.AC.UK
Henry Gouk HENRY.GOUK@ED.AC.UK
Informatics Forum, University of Edinburgh, Edinburgh, United Kingdom

Abstract

Algorithmic developments in Strategic Classification have been mostly limited to linear classifiers in settings where the best response has a closed-form solution or can be easily approximated. While some work has explored the role of non-linear classifiers in strategic settings, progress in this direction is impeded by the computational intractability of the strategic behaviour. Addressing this, we present a novel method for approximating the best response by exploiting Lagrangian duality. By reformulating the strategic response as a constrained optimisation problem, we can construct a Lagrangian that is amenable to first order optimisation methods. This approach reproduces closed-form strategic behaviour in linear settings and can be straight-forwardly applied to non-linear settings. We show how the Implicit Function Theorem can be used in conjunction with our proposed response formulation during classifier learning to compute the total gradient of the loss. This connects the classifier parameters directly to the consequent strategic behaviour, yielding a novel training algorithm that can exploit this relationship. Experimental evaluation shows that the resulting models achieve improved strategic accuracy on common machine learning datasets.

1. Introduction

Those with a preference for a certain classification, and knowledge of the classifier that is being used, have an incentive to misrepresent their state to receive a favourable outcome [15]. This dynamic can arise in many sociotechnical situations, spanning from universities deciding which students to enrol as far as institutions determining how to distribute aid and resources in times of need. However the phenomenon is broader than classically considered sociotechnical situations: such strategic behaviour can arise in a broader range of applications than just those that deal with classifying individuals. While certain applications are well-served by simple linear models, in many cases deep neural networks have been shown to provide excellent predictive performance.

Strategic classification addresses settings where a *Learner* attempts to classify a population of *Agents* who strategically perturb their representations to optimise a utility known to both players [2, 15]. The Learner's goal is to develop a classifier that remains robust to these perturbations, thereby minimising the impact of manipulation. However, computing the optimal response is a significant challenge, as it is typically discontinuous and non-differentiable in most cases. Consequently, much of the existing literature focuses on linear classifiers, which can allow for closed-form expressions of the best response in some situations [6, 18, 19]. While this focus on linearity is advantageous for the interpretability required in sociotechnical scenarios, it limits the scope of research and the potential for applying these developments to less socially sensitive settings that could benefit from more flexible models.

To address this limitation we observe that the Agents' objective can be reformulated as a constrained optimisation problem. This admits the derivation the Lagrangian dual corresponding to the problem, which can be used as a surrogate method for approximating the Agents' best response. We show that in simple environments our Lagrangian dual-based method for computing the best response better approximates the expected best response than existing gradient-based approaches. Moreover, leveraging this novel method for computing responses to yield an expression for the total derivative of the strategic classification objective. This enables us to explore the utility of directly optimising the standard learning objective, and the development of more sophisticated strategically robust models.

In summary, we make the following contributions:

- We propose a novel method for approximating the best response to a given classifier based on the Lagrangian Dual of the Agents' objective. This method addresses key weaknesses in current response approximation methods, and can be directly applied in settings with linear or non-linear classifiers.
- We observe that the Lagrangian Dual response formulation provides a model for relating the learned parameters to the Agents' behaviour. We propose a novel training algorithm that uses the Implicit Function Theorem to exploit this relationship and demonstrate that the resulting models show improved strategic accuracy on various real world datasets.
- We identify a property of some response methods, *reputability*, indicating a method reliably identifies Agent states that are vulnerable to gaming. We prove that the agreement between reputable response methods and the true best response is monotonically related to the number of points the gamed by the response method, allowing for the comparison of different response methods in non-linear settings, without requiring a source of ground-truth gaming behaviour.

2. Strategic Classification and the Best Response

The classical classifier learning problem can be formulated as follows; for a feature space, $\mathcal{X}$, label space, $\mathcal{Y} = \{-1, 1\}$, and class of models $\mathcal{F}$, find a model, $f_\theta \in \mathcal{F}$, that achieves high accuracy according to the data distribution $\mathcal{D} = \mathcal{P}(\mathcal{X} \times \mathcal{Y})$.

In many realistic classification scenarios $\mathcal{D}$ is not representative of the data that a trained classifier will actually be deployed on [25, 27]. In strategic settings, each data point $(\mathbf{x}, \mathrm{y})$ may be associated with some Agent that can perturb their representation, $\mathbf{x}$. In particular, given that the Agent knows the classifier model, f_θ, they may be motivated to respond to the classifier by manipulating their state in order to obtain a positive classification, irrespective of their true label, y. This is typically modelled using an idealised best response function [15],

$$\Delta^*(\boldsymbol{x}, \theta) = \underset{\boldsymbol{z} \in \mathcal{X}}{\arg\max}\, f_\theta(\boldsymbol{z}) - c(\boldsymbol{x}, \boldsymbol{z}), \tag{1}$$

that trades off receiving a positive classification with some cost, $c(\boldsymbol{x}, \boldsymbol{z})$.

Strategic behaviour can, in practise, significantly undermine the performance of the resulting classifier [18]. To address this, algorithms have been designed to support training of classifiers that are robust to such misrepresentations [18, 24, 29]. In Strategic Classification, the objective is to minimise the empirical loss evaluated at the best response, $\Delta(x, \theta)$, which has lead to the Strategic

Empirical Risk Minimisation (SERM) approach [29],

$$\tilde{\theta} = \underset{\theta \in \Theta}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} l(\theta, \Delta^*(\mathbf{x}_i, \theta), \mathrm{y}_i) \tag{2}$$
$$\text{subject to} \quad \Delta^*(\boldsymbol{x}, \theta) = \underset{\boldsymbol{z} \in \mathcal{X}}{\arg\max}\, f_\theta(\boldsymbol{z}) - c(\boldsymbol{x}, \boldsymbol{z}).$$

This is a bilevel optimisation problem, where the $\arg\min$ over θ is referred to as the upper level and the $\arg\max$ over $\boldsymbol{z}$ as the lower level. Observe that computing $\tilde{\theta}$ relies heavily on the ability to solve the lower level problem, which can, for general $\mathcal{F}$ be intractable to solve. As a result, much of the work in the field is focussed on identifying settings and algorithms for which tractable solutions exist [15, 18, 24]. In the interest of saving space we defer a more comprehensive overview of previous literature in this area to Appendix A.

3. Evaluating Responses

The solution to the problem posed in Equation 2 relies heavily on the best response, Δ^*. While there are limited settings where has a tractable solution, in practise Δ^* generally can't be exactly computed and must be approximated. While previous work has explored methods to approximate Δ^* (e.g., [24]), the question of how to evaluate the quality of an approximate response remains open. This is particularly pertinent in the case of responses designed to respond to non-linear classifiers, where the true Δ^* can't be observed. To address this we propose a measure, *Agreement*, that can be used to quantify the rate of agreement between a response method, and the true best response.

3.1. Agreement

Let $\Delta : \mathcal{X} \times \Theta \to \mathcal{X}$ be an approximate solution to Δ^*. A classifier $f_\theta, \theta \in \Theta$ is gamed by Δ at a point $\boldsymbol{x} \in \mathcal{X}$ if $f_\theta(\Delta(\boldsymbol{x}, \theta)) \neq f_\theta(x)$. Let $I_S^\theta(\Delta) = \{\boldsymbol{x} \in S | f_\theta(\Delta(\boldsymbol{x}, \theta)) \neq f_\theta(\boldsymbol{x})\}$ be the set of points in S gamed by Δ. For a dataset S, the Agreement between two response methods Δ_1, Δ_2, $A_S^\theta(\Delta_1, \Delta_2) \in [0, 1]$, is given by the Jaccard Index between the two sets:

$$A_S^\theta(\Delta_1, \Delta_2) = \frac{|I_S^\theta(\Delta_1) \cap I_S^\theta(\Delta_2)|}{|I_S^\theta(\Delta_1) \cup I_S^\theta(\Delta_2)|}. \tag{3}$$

An Agreement of 0 implies that the methods did not agree at all, while an agreement of 1 indicates perfect alignment in what points should be gamed.

$A_S^\theta(\Delta, \Delta^*)$ can be seen as an estimate of how well Δ approximates Δ^* on S. In particular, for two response methods, Δ_1, Δ_2, if $A_S^\theta(\Delta_1, \Delta^*) > A_S^\theta(\Delta_2, \Delta^*)$ then Δ_1 is a better approximation of Δ^* than Δ_2 on S.

3.2. Agreement With an Unobservable Best Response

In cases when Δ^* can't be computed, to be able to compare response methods, it is necessary to derive a proxy measure to Agreement that does not directly rely on evaluating Δ^*. To derive such a measure we will rely on the following definition;

Definition 1 (Reputability) *A response method Δ is said to be Reputable if, for any parameter setting $\theta \in \Theta$, Δ gaming a point $\boldsymbol{x} \in \mathcal{X}$ implies that $\boldsymbol{x}$ is a point that would be gamed by the best response, Δ^*. Specifically:*

$$f_\theta(\Delta(\boldsymbol{x}, \theta)) \neq f_\theta(\boldsymbol{x}) \implies f_\theta(\Delta^*(\boldsymbol{x}, \theta)) \neq f_\theta(\boldsymbol{x}). \tag{4}$$

Observe that if Δ is reputable, then $|I_S^\theta(\Delta)|$ provides a lower bound on the number of points that are gameable under the unobservable Δ^*. As a consequence of this it can be shown that the reputability property is sufficient for $|I_S^\theta(\Delta)|$ to serve as an adequate proxy for comparing the agreement of two response methods.

Theorem 2 *For two reputable response methods, Δ_1, Δ_2, if $|I_S^\theta(\Delta_1)| > |I_S^\theta(\Delta_2)| \iff A_S^\theta(\Delta_1, \Delta^*) > A_S^\theta(\Delta_2, \Delta^*)$*

The proof of Theorem 2 is provided in Appendix B. A direct consequence of this theorem is that, for reputable responses, while it is not possible to determine what "true" gaming behaviour would be in the setting, it is possible to rank order the responses based on how well they approximate the behaviour by the number of points in S they succeed in gaming.

4. Computing the Best Response

For a linear model with a squared Euclidean cost,

$$c(\boldsymbol{x}, \boldsymbol{z}) = \frac{1}{2\epsilon}||\boldsymbol{x} - \boldsymbol{z}||_2^2 \tag{5}$$

the best response has a closed-form solution and can be computed exactly. In other cases, existing methods simply approximate the best response by optimising differentiable relaxations of the Agent's objective (see Appendix C for a more details). We improve upon this by reformulating the original objective in a way that makes it more conducive to gradient-based optimisation.

By explicitly addressing the motivation of the lower level objective as being to find a minimum cost response that successfully games the classifier, we can treat it as a constrained optimisation problem. We can write Equation 1 equivalently as,

$$\begin{aligned} \Delta^*(\mathbf{x}, \theta) = \underset{\mathbf{z} \in \mathcal{X}}{\arg\min} \quad & c(\mathbf{x}, \mathbf{z}), \\ \text{subject to} \quad & h(\mathbf{z}) \geq 0, \\ & c(\mathbf{x}, \mathbf{z}) \leq 2. \end{aligned} \tag{6}$$

In practice, one must constrain $h(\boldsymbol{z}) \geq \gamma$ for some small $\gamma > 0$ to ensure the inequality holds strictly. Observe that this formulation captures the same constraints as in Equation 1, but without depending on the interaction between the utility and the cost during the optimisation.

The constrained optimisation problem proposed in Equation 6 is amenable to Karush-Kuhn-Tucker (KKT) solution methods [12, 17]. Under this formulation we can replace the objective in Equation 6 with the Lagrangian dual,

$$\mathcal{L}(\theta, \boldsymbol{x}, \boldsymbol{z}, \mu_1, \mu_2) = c(\boldsymbol{x}, \boldsymbol{z}) + \mu_1(h_\theta(\boldsymbol{z}) - \epsilon) + \mu_2(c(\boldsymbol{x}, \boldsymbol{z}) - 2), \tag{7}$$

where $\mu_1, \mu_2 \in \mathbb{R}$ are Lagrange multipliers. This is used to define an optimisation problem,

$$\Delta^{LD}(\mathbf{x}, \theta) = \arg\min_{\mathbf{z}} \max_{\mu_1,\mu_2} \quad \mathcal{L}(\theta, \mathbf{x}, \mathbf{z}, \mu_1, \mu_2) \quad \text{subject to} \quad \mu_1, \mu_2 \geq 0, \tag{8}$$

that can be solved with projected gradient ascent-descent. The projection function simply clamps the μ_1 and μ_2 Lagrange multipliers to ensure they remain non-negative. The Lagrangian dual solution converges to a locally optimal solution whilst enforcing the constraints. In the case where all local optima are global optima—e.g., if h_θ and c are convex in $\boldsymbol{z}$—this approach is guaranteed to compute the best response.

4.1. Post-Response Checks

Valid responses must satisfy two invariants, $f_\theta(\Delta(\mathbf{x}, \theta)) > 0$ and $c(\mathbf{x}, \mathbf{z}) < 2$. Approximate responses may violate these invariants in some instances. The Lagrangian Dual may violate one of these constraints if there is no feasible point. In practise, to avoid these issues, we manually check that both invariants are satisfied and return the original $\boldsymbol{x}$ if they are not. For the remainder of this work, unless stated otherwise, it can be assumed that the response methods evaluated have these post-response checks applied to them.

5. Training Strategically Robust Classifiers

The main complication in optimising the objective in Equation 2 is the need to backpropagate through the $\arg\max$ in the lower level of the problem. To overcome this, we show how the total derivative of the objective can be computed, facilitating the application of any of the gradient-based optimisation methods typically used for training deep networks [10, 13]. The total derivative for a single term of the summation in the strategic empirical risk can be decomposed as

$$\frac{dl}{d\theta} = \underbrace{\frac{\partial l}{\partial \theta}}_{\text{Direct Derivative}} + \underbrace{\frac{\partial l}{\partial \Delta}\frac{\partial \Delta}{\partial \theta}}_{\text{Indirect Derivative}} . \tag{9}$$

The direct derivative is computed via the backpropagation process for conventional training of neural networks, but with the strategically manipulated features used in place of the true features. The indirect derivative represents a correction that takes into account how the strategic response depends on θ. We explore two options for computing this term: (i) leveraging implicit gradient methods coupled matrix free linear algebra routines; and (ii) ignoring it completely, thus recovering a version of the REGD approach of [24] suitable for the strategic learning setting.

5.1. Computing the Indirect Derivative

Computing the indirect derivative is complicated by the presence of the $\arg\max$ and the indicator function present in the definition of the best response in Equation 1. The Lagrangian Dual formulation of the best response given in Equation 8 allows us to circumvent the indicator function, but actually further complicates the $\arg\max$ issue by introducing the maximisation over the Lagrange multipliers. We address this problem using so-called implicit gradient techniques that make use of the implicit function theorem and optimality conditions of the lower level problem to yield an expression for the gradient of Δ with respect to θ.

Table 1: Strategic Accuracies (%) realised for MLP model trained with different strategic training methods. Error bars represent 95% binomial confidence interval. Best accuracy in each comparison is emphasised.

Dataset	ERM	REGD	REGD-LD	TGD
Bank Customer Churn	80.75 ± 0.88	67.55 ± 1.05	81.30 ± 0.87	$\mathbf{85.50 \pm 0.79}$
default of credit card	81.55 ± 0.50	81.32 ± 0.50	82.23 ± 0.49	$\mathbf{82.53 \pm 0.49}$
Employee	74.87 ± 1.42	76.58 ± 1.39	76.80 ± 1.38	$\mathbf{76.91 \pm 1.38}$
GMSC	61.20 ± 0.84	56.42 ± 0.86	60.81 ± 0.84	$\mathbf{64.19 \pm 0.83}$
Houses	93.31 ± 0.39	93.19 ± 0.39	95.13 ± 0.34	$\mathbf{96.00 \pm 0.31}$
HR Analytics	78.84 ± 0.66	$\mathbf{80.25 \pm 0.64}$	78.71 ± 0.66	78.21 ± 0.67
mobile c36	62.18 ± 0.48	63.60 ± 0.47	$\mathbf{85.96 \pm 0.34}$	64.46 ± 0.47
statlog	$\mathbf{71.00 \pm 3.21}$	68.00 ± 3.30	69.50 ± 3.26	$\mathbf{71.00 \pm 3.21}$
Water Quality	60.11 ± 2.09	57.56 ± 2.11	58.29 ± 2.10	$\mathbf{62.30 \pm 2.07}$

Theorem 3 *The indirect derivative is given by*

$$\frac{\partial}{\partial \Delta}\frac{\partial \Delta}{\partial \theta} = -\begin{bmatrix} \frac{\partial l}{\partial \Delta} & 0 & 0 \end{bmatrix} \begin{bmatrix} \nabla^2_{\boldsymbol{zz}}\mathcal{L} & \nabla_{\boldsymbol{z}} h_\theta & \nabla_{\boldsymbol{z}} c \\ \mu_1 \nabla_{\boldsymbol{z}}^\top h_\theta & h_\theta - \epsilon & 0 \\ \mu_2 \nabla_{\boldsymbol{z}}^\top c & 0 & c-2 \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial \theta \partial \boldsymbol{z}} \\ \mu_1 \frac{\partial h_\theta}{\partial \theta} \\ 0 \end{bmatrix}.$$

In the interests of space, the proof of this Theorem is deferred to Appendix D. This approach to computing the total derivative of the strategic classification objective provides an ideal vehicle for experimentally determining the importance of indirect derivative. Table 1 presents the strategic accuracy of models trained with various training approaches on a sample of common machine learning datasets. Our results show that optimising the total gradient during training (TGD) results in improved robustness as compared to models trained with standard ERM, direct SERM optimisation with the REGD method proposed in [24], and strategic REGD optimsation when the Lagrangian response is used instead of the gradient-based response. A more thorough set of results, including a more detailed evaluation of Lagrangian and Gradient based responses can be found in Appendix E.

We defer the empirical work performed to support our theoretical results to Appendix E

6. Discussion & Conclusion

As technology advances, the need to develop non-linear classifier models that are robust to strategic behaviour will grow. The complexities associated with accurately approximating best response behaviour have impeded the development of approaches for training non-linear models that can be robust to strategic behaviour. In this work we have demonstrated how the Agents' objective in strategic classification can be reformulated as a constrained optimisation problem which is amenable to Lagrangian dual optimisation. We proposed a novel strategic training algorithm, TGD, which incorporated the total gradient into the training loop, and experimentally demonstrated that models trained with this algorithm consistently outperformed other approaches on strategic accuracy. We proposed Agreement as a novel metric for evaluating approximations of the best response and proved that, under minor conditions, the number of points gamed by different approximations could be used as an appropriate proxy to Agreement, preserving rank ordering per this metric.

Acknowledgements

This work was funded by NatWest Group via the Centre for Purpose-Driven Innovation in Banking. This project was supported by the Royal Academy of Engineering under the Research Fellowship programme.

References

[1] Mark Braverman and Sumegha Garg. The role of randomness and noise in strategic classification. *arXiv preprint arXiv:2005.08377*, 2020.

[2] Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *KDD*, 2011.

[3] Lee Cohen, Saeed Sharifi-Malvajerdi, Kevin Stang, Ali Vakilian, and Juba Ziani. Bayesian strategic classification. In *NeurIPS*, 2024.

[4] Edwige Cyffers, Muni Sreenivas Pydi, Jamal Atif, and Olivier Cappé. Optimal classification under performative distribution shift. *Advances in Neural Information Processing Systems*, 37:68144–68160, 2024.

[5] Justin Domke. Generic methods for optimization-based modeling. In *AISTATS*, 2012.

[6] Itay Eilat, Ben Finkelshtein, Chaim Baskin, and Nir Rosenfeld. Strategic classification with graph neural networks. *arXiv preprint arXiv:2205.15765*, 2022.

[7] Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *ICML*, 2018.

[8] Credit Fusion and Will Cukierski. Give me some credit. https://kaggle.com/competitions/GiveMeSomeCredit, 2011. Kaggle.

[9] Boyan Gao, Henry Gouk, Hae Beom Lee, and Timothy M Hospedales. Meta mirror descent: Optimiser learning for fast convergence. *arXiv preprint arXiv:2203.02711*, 2022.

[10] Boyan Gao, Henry Gouk, Yongxin Yang, and Timothy Hospedales. Loss function learning for domain generalization by implicit gradient. In *ICML*, 2022.

[11] Jack Geary and Henry Gouk. Strategic classification with randomised classifiers. *arXiv preprint arXiv:2502.01313*, 2025.

[12] Geoff Gordon and Ryan Tibshirani. Karush-kuhn-tucker conditions. *Optimization*, 10 (725/36):725, 2012.

[13] Stephen Gould, Basura Fernando, Anoop Cherian, and Peter Anderson. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.

[14] Michael Großhans, Christoph Sawade, Michael Brückner, and Tobias Scheffer. Bayesian games for adversarial regression problems. In *ICML*, 2013.

[15] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pages 111–122, 2016.

[16] Zachary Izzo, Lexing Ying, and James Zou. How to learn when data reacts to your model: performative gradient descent. In *ICML*, 2021.

[17] Harold William Kuhn and Albert William Tucker. Linear inequalities and related systems.(am-38), 1957.

[18] Sagi Levanon and Nir Rosenfeld. Strategic classification made practical. In *ICML*, 2021.

[19] Sagi Levanon and Nir Rosenfeld. Generalized strategic classification and the case of aligned incentives. In *ICML*, 2022.

[20] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *AISTATS*, 2020.

[21] John P Miller, Juan C Perdomo, and Tijana Zrnic. Outside the echo chamber: Optimizing the performative risk. In *International Conference on Machine Learning*, pages 7710–7720. PMLR, 2021.

[22] Mehrnaz Mofakhami, Ioannis Mitliagkas, and Gauthier Gidel. Performative prediction with neural networks. In *AISTATS*, 2023.

[23] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *ICML*, 2016.

[24] Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *ICML*, 2020.

[25] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. MIT Press, 2022.

[26] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *NeurIPS*, 2019.

[27] Elan Rosenfeld. *Understanding, Formally Characterizing, and Robustly Handling Real-World Distribution Shift*. PhD thesis, Carnegie Mellon University, 2023.

[28] Elan Rosenfeld and Nir Rosenfeld. One-shot strategic classification under unknown costs. *arXiv preprint arXiv:2311.02761*, 2023.

[29] Ravi Sundaram, Anil Vullikanti, Haifeng Xu, and Fan Yao. Pac-learning for strategic classification. *Journal of Machine Learning Research*, 24(192):1–38, 2023.

[30] Benyamin Trachtenberg and Nir Rosenfeld. Strategic classification with non-linear classifiers. *arXiv preprint arXiv:2505.23443*, 2025.

[31] Joaquin Vanschoren. houses. https://www.openml.org/d/823, 2014. OpenML.

[32] Han-Jia Ye, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, and De-Chuan Zhan. A closer look at deep learning methods on tabular datasets. *arXiv preprint arXiv:2407.00956*, 2024.

[33] Guangzheng Zhong, Yang Liu, and Jiming Liu. Nonlinear performative prediction. *arXiv preprint arXiv:2509.01139*, 2025.

[34] Nicolas Zucchet and Joao Sacramento. Beyond backpropagation: bilevel optimization through implicit differentiation and equilibrium propagation. *Neural Computation*, 34(12):2309–2346, 2022.

Camera-Ready Edits Summary

Reviewer feedback did not suggest any revisions to the paper. An Acknowledgements section has been added to the camera-ready version of this document which was not included in the original submission. Otherwise, no revisions were made to content or figures of this paper prior to camera-ready submission.

Appendix A. Related Work

Building off of the setting established by [2, 14], Strategic Classification was originally formulated by Hardt et al. [15]. This models the classifier learning problem as an interaction between a Learner attempting to construct a classifier, and Agents that have perfect awareness of the classifier and can pay perturb their state before being classified. It is assumed that Agents receive positive utility from positive classification, and that the costs the Agents can pay to change their state is known to both the Agents and the Learner. The resulting optimisation problem is generally computationally intractable; in order to empirically evaluate their algorithm, Hardt et al. make the simplifying assumption that the learned classifier is a linear SVM model [15].

Subsequent works in this area have relaxed some of the assumptions made in the original paper; [3] weaken the assumption that the Agents know the Learner's classifier, [19] assume that Agents can have a preference for positive or negative classification, [28] examines the consequences when the Agents' cost function is unknown to the Learner, and [1, 11, 29] explore the setting where there is randomness or uncertainty in the classifier learned by the Learner. However, all of these relaxations are considered exclusively in the setting where the classifier being learned is a linear model, and the results are primarily demonstrated on small-scale toy datasets.

Trachtenberg and Rosenfeld [30] explicitly explore strategic behaviour for non-linear classifiers. However, their principal motivation is on demonstrating the possible effects of strategic behaviour, and are able to realise this experimentally without having to explicitly compute the best response. [18] also consider applications of their approach to non-linear classifiers, in the form of recurrent neural network models applied to time series data. However, their solution approach still necessarily relies on model convexity in the inputs, which is not generally true in non-linear settings.

A central motivation for the widespread adoption of the linear classifier in Strategic Classification research is its theoretical simplicity, and that its best response has a closed form definition. This makes it possible to learn strategically robust linear classifiers without actually having to solve the bilevel optimisation problem inherent to the standard problem formulation. This is often used in Strategic Classification literature (e.g., [6, 18, 19]). While this facilitates practical experimentation, it limits results to linear models. Some work has been done to weaken this restriction; [19] in particular, propose a loss function, the Strategic Hinge loss, which can be used when optimising linear models without explicitly computing the best response. However, in general there has been little progress for non-linear models.

Computing the total derivative for optimisation is a commonly used tool in bilevel optimisation [34]. The implicit function theorem is widely used to compute the total derivative in these settings [13, 23, 34]. This approach has been applied in hyperparameter optimization [7, 20] meta-learning [9, 10, 26], and differentiable programming [5], providing scalable alternatives to costly unrolled optimization. In contrast with these settings, where it is assumed that the lower level objective is unconstrained, in this work the lower level problem is itself subject to constraints. In fields such as Performative Prediction, recent work has explored using the total derivative in model learning [4, 16, 21, 22, 33]. However, this setting provides no access to the underlying response model; these approaches rely on zeroth order estimation methods to approximate the total derivative.

Appendix B. Proof of Theorem 2

Proof Without loss of generality, assume every point gamed by Δ_2 is also gamed by Δ_1:

$$f_\theta(\Delta_2(\boldsymbol{x},\theta)) \neq f_\theta(\boldsymbol{x}) \implies f_\theta(\Delta_1(\boldsymbol{x},\theta)) \neq f_\theta(\boldsymbol{x}). \tag{10}$$

Therefore,

$$I_S^\theta(\Delta_2) \subseteq I_S^\theta(\Delta_1) \tag{11}$$

$$\implies I_S^\theta(\Delta_2) \cap I_S^\theta(\Delta^*) \subseteq I_S^\theta(\Delta_1) \cap I_S^\theta(\Delta^*). \tag{12}$$

$|I_S^\theta(\Delta_1)| > |I_S^\theta(\Delta_2)| \implies \exists \boldsymbol{x}' \in S$ s.t $f_\theta(\Delta_1(\boldsymbol{x}',\theta)) \neq f_\theta(\boldsymbol{x}')$ and $f_\theta(\Delta_2(\boldsymbol{x}',\theta)) = f_\theta(\boldsymbol{x}')$. Since Δ_1 is reputable, $f_\theta(\Delta_1(\boldsymbol{x}',\theta)) \neq f_\theta(\boldsymbol{x}') \implies f_\theta(\Delta^*(\boldsymbol{x}',\theta)) \neq f_\theta(\boldsymbol{x}')$. Therefore $\boldsymbol{x}' \in I_S^\theta(\Delta_1) \cap I_S^\theta(\Delta^*)$ and $\boldsymbol{x}' \notin I_S^\theta(\Delta_2) \cap I_S^\theta(\Delta^*)$. By definition, it follows that $A_S^\theta(\Delta_1, \Delta^*) > A_S^\theta(\Delta_2, \Delta^*)$.

To prove equivalence we consider the fact that, for a reputable response Δ, $I_S^\theta(\Delta_1) \cup I_S^\theta(\Delta^*) = I_S^\theta(\Delta^*)$. Therefore

$$A_S^\theta(\Delta_1, \Delta^*) > A_S^\theta(\Delta_2, \Delta^*) \tag{13}$$

$$\implies |I_S^\theta(\Delta_1) \cap I_S^\theta(\Delta^*)| > |I_S^\theta(\Delta_2) \cap I_S^\theta(\Delta^*)| \tag{14}$$

$$\implies |I_S^\theta(\Delta_1)| > |I_S^\theta(\Delta_2)| \tag{15}$$

■

Appendix C. Best Response Approximations

C.1. Linear Response

For a linear model with Euclidean cost, the best response has a closed-form solution and can be computed exactly. As a result, Equation 2 collapses to a single level problem;

$$\Delta(\boldsymbol{x},\theta) = \begin{cases} \boldsymbol{x} - \theta \frac{h_\theta(\boldsymbol{x})}{\|\theta\|_2} & \text{if } -2 \leq h_\theta(\boldsymbol{x}) < 0 \\ \boldsymbol{x} & \text{otherwise.} \end{cases} \tag{16}$$

where, for simplicity we have let $\epsilon = 1$.

C.2. Gradient Response

Gradient-based methods are a common approach for solving optimisation problems such as Equation 2. Such methods involve using gradient ascent to solve the lower level maximisation, and gradient descent to solve the upper level minimisation. Perdomo et al. [24], propose an iterative gradient-based method that they show can be used to approximate the best response in Strategic Classification problems. However, we note that, since f is a binary classifier, it is not differentiable, and so gradient methods cannot be directly applied to the problem as presented in Equation 2.

Instead, the response method proposed in [24] computes the approximation to a relaxation of the problem in Equation 2, where the binary classifier, f_θ, is replaced in the lower objective with the corresponding $h_\theta \in \mathcal{H}$ (such that $f_\theta(x) = sgn(h_\theta(x))$), where $\mathcal{H}$ is a class of differentiable models. This effectively reformulates the best response objective given in Equation 1 as

$$\Delta^{GD}(\boldsymbol{x}, \theta) = \underset{\boldsymbol{z} \in \mathcal{X}}{\arg\max}\, h_\theta(\boldsymbol{z}) - c(\boldsymbol{x}, \boldsymbol{z}). \tag{17}$$

As a consequence of this relaxation, one of the constraints on the best response is relaxed; in the best response of Equation 1, solutions trade off maximising $f_\theta(\boldsymbol{z})$, minimising the cost $c(\boldsymbol{x}, \boldsymbol{z})$, and constraining the cost to be less than two ($c(\boldsymbol{x}, \boldsymbol{z}) \leq 2$). This last constraint arises from the interaction between the utility and the cost terms in the optimisation; f is binary and the maximum the utility can be improved by is two. If the cost to realise this exceeds two then the Agent would realise a higher objective by letting $\boldsymbol{z} = \boldsymbol{x}$. This constraint does not necessarily apply in the case where f_θ is replaced with h_θ. In particular, we observe that in the linear case with the squared Euclidean cost, the gradient response converges to

$$\boldsymbol{z} = \boldsymbol{x} + \epsilon\theta \tag{18}$$

which, depending on the value for θ may wildly over- or under-estimate true gaming behaviour. This is evident from Figure 1 (centre-right) where the Gradient Response only identifies a subset of the negative points as being vulnerable to gaming and, as such, is a poor approximation of the best response.

Appendix D. Proof of Theorem 3

Proof Denote the concatenation of the primal and dual variables of the Lagrangian Dual response by $\boldsymbol{w} = (\boldsymbol{z}, \mu_1, \mu_2)$. Consider the residual of this problem,

$$F(\theta, \boldsymbol{x}, \boldsymbol{w}) = \begin{bmatrix} \nabla_{\boldsymbol{z}} \mathcal{L}(\theta, \boldsymbol{x}, \boldsymbol{z}, \mu_1, \mu_2) \\ \mu_1(h_\theta(\boldsymbol{z}) - \epsilon) \\ \mu_2(c(\boldsymbol{x}, \boldsymbol{z}) - 2) \end{bmatrix}. \tag{19}$$

Let $\boldsymbol{w}^*(\theta, \boldsymbol{x})$ be the solution to the KKT problem for a given θ and $\boldsymbol{x}$. Then we can define the implicit function,

$$F(\theta, \boldsymbol{x}, \boldsymbol{w}^*(\theta, \boldsymbol{x})) = 0. \tag{20}$$

By the implicit function theorem,

$$0 = \frac{dF}{d\theta} \tag{21}$$

$$0 = \frac{\partial F}{\partial \theta} + \frac{\partial F}{\partial \boldsymbol{w}^*} \frac{\partial \boldsymbol{w}^*}{\partial \theta}, \tag{22}$$

where we have omitted the point where the derivatives are evaluated for readability. Rearranging, we obtain the expression for the indirect derivative,

$$\frac{\partial \boldsymbol{w}^*}{\partial \theta} = -\left(\frac{\partial F}{\partial \boldsymbol{w}^*}\right)^{-1} \frac{\partial F}{\partial \theta}. \tag{23}$$

The result follows from computing the derivatives for elements of F. ■

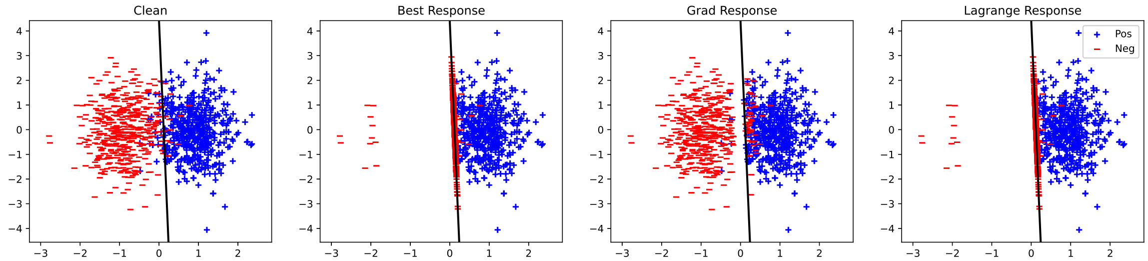

Figure 1: Various responses to a linear classifier trained on the Gaussians dataset (negative points marked with red '-', positive points with blue '+'): (Left) No Response; (Middle-Left) Best Response; (Middle-Right) Gradient Response; (Right) Lagrangian Response

Appendix E. Experimental Results

We experimentally investigate the utility of our proposed Lagrangian Dual response from two perspectives. First, we evaluate how effectively it can be used to respond to classifiers that have not necessarily been trained to be robust to this specific type of strategic response. Second, we investigate the efficacy of the two strategic training techniques discussed in Section 5: directly optimising the total derivative (TGD), or using a version of REGD specialised for strategic settings. Details on model architectures and training procedures are given in Appendix F.

Experiments in this section are performed on a suite of datasets compiled primarily from a subset of the datasets in the TALENT collection ([32]). The set also contains the Housing dataset ([31]), as used in [4], and the Give Me Some Credit dataset ([8]) as used in [22, 24]. Further details about the specific datasets used in our experiments can be found in Appendix G.

E.1. Responding to Linear Models

To demonstrate the effect this response has on the data distribution, we consider the two dimensional Gaussian dataset proposed in [19]; a simple two dimensional dataset containing points sampled i.i.d from $S = \{(\boldsymbol{x}_i, 1) \sim \mathcal{N}(\mu_1, 1)\}_{i=1}^n \cup \{(\boldsymbol{x}_i, -1) \sim \mathcal{N}(\mu_{-1}, 1)\}_{i=1}^n, \mu_{-1} < \mu_1$. We train a linear support vector machine (SVM) on the dataset, without taking any effort to make the classifier strategically robust. We consider three methods for computing strategic responses under the Euclidean cost: the exact closed-form solution, the Gradient response (Equation 17), and the Lagrangian Dual response (Equation 8). For all three approaches we use the squared Euclidean cost.

Figure 1 provides a visual comparison of these response methods on the Gaussians dataset. The Left pane shows the unperturbed dataset along with the location of the SVM decision boundary. Comparing with the optimal response (Middle-Left), we observe that the Gradient response (Middle-Right) is susceptible to two types of errors: (i) some points move too far over the decision boundary, meaning they incur more cost than is necessary; and (ii) some points that could game the model are not manipulated. In contrast to this, our Lagrangian Dual respons (Right) is able to exactly replicate the behaviour of the true best response.

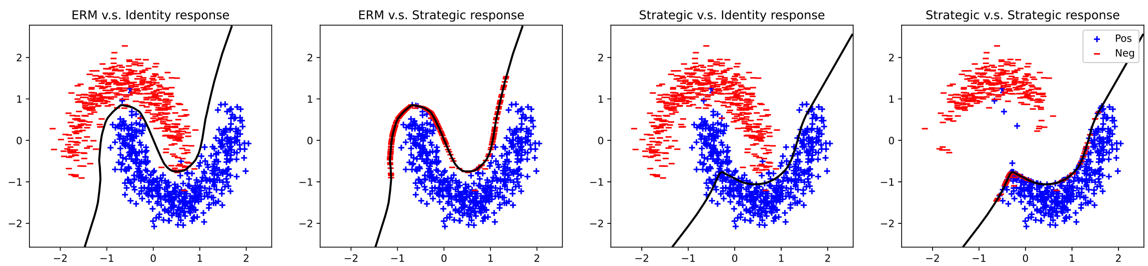

Figure 2: Various responses to an MLP classifier trained on the twin moons dataset (negative points marked with red '-', positive points with blue '+'): (Left) ERM with no response; (Middle-Left) ERM with strategic response; (Middle-Right) REGD-LD with no response; (Right) REGD-LD with strategic response

Table 2: Proportion (%) of points gamed for linear and MLP models trained on clean datasets under Gradient (Δ^{GD}) and Lagrangian (Δ^{LD}) response models. Error bars represent 95% binomial confidence interval. Larger value in each comparison is emphasised.

	Linear		**MLP**	
Dataset	Δ^{GD}	Δ^{LD}	Δ^{GD}	Δ^{LD}
Bank Customer Churn	3.05 ± 0.38	$\mathbf{24.45 \pm 0.96}$	0.40 ± 0.14	$\mathbf{12.80 \pm 0.75}$
default of credit card	1.27 ± 0.14	$\mathbf{6.02 \pm 0.31}$	1.67 ± 0.17	$\mathbf{4.70 \pm 0.27}$
Employee	0.00 ± 0.00	$\mathbf{0.32 \pm 0.19}$	$\mathbf{17.29 \pm 1.24}$	15.57 ± 1.19
GMSC	3.83 ± 0.33	$\mathbf{24.92 \pm 0.75}$	4.97 ± 0.38	$\mathbf{20.85 \pm 0.70}$
Houses	1.45 ± 0.19	$\mathbf{10.76 \pm 0.48}$	1.31 ± 0.18	$\mathbf{9.54 \pm 0.46}$
HR Analytics	3.76 ± 0.31	$\mathbf{8.77 \pm 0.46}$	4.38 ± 0.33	$\mathbf{13.26 \pm 0.55}$
mobile c36	34.98 ± 0.47	$\mathbf{37.64 \pm 0.48}$	0.00 ± 0.00	$\mathbf{28.75 \pm 0.44}$
statlog	1.50 ± 0.86	$\mathbf{32.00 \pm 3.30}$	1.50 ± 0.86	$\mathbf{25.50 \pm 3.08}$
Water Quality	0.00 ± 0.00	$\mathbf{16.76 \pm 1.59}$	0.73 ± 0.36	$\mathbf{12.75 \pm 1.42}$

E.2. Responding to Non-Linear Models

To qualitatively investigate whether our Lagrangian Dual response is able to respond to non-linear models, we consider two MLP classifiers; one trained with ERM, and another trained using the strategic REGD method [24] trained on the twin moons dataset (Figure 3).

Figure 2 (Left) visualises the ERM decision boundary and the unperturbed dataset. While this is optimal in absence of strategic behaviour, all of the negative points are close to the decision boundary, and our strategic response is therefore able to move them to lie on the positive side of the decision boundary (Middle-Left). When the model is trained with strategic REGD (Middle-Right), accuracy on the unperturbed data points is poor, but when the strategic behaviour is applied in the Right pane, we see that all the positive points are classified correctly but only a subset of the negatively labelled points successfully game the classifier.

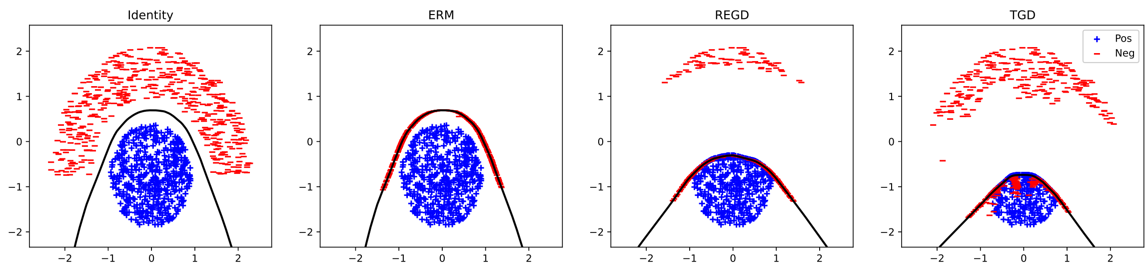

Figure 3: Results of Strategic Training on an MLP on the ball-and-disk dataset (negative points marked with red '-', positive points with blue '+'): (Left) ERM model; (Middle-Left) ERM model with Strategic response; (Middle-Right) REGD model ([24]) with Strategic Response; (Right) TGD model (Ours) with Strategic response.

E.3. Strategic Responses on Real Data

Theorem 2 presents a means of comparing the relative performance of different response methods in both linear and non-linear settings. However,to make use of this result we assert the following proposition (proof in Appendix H);

Proposition 4 *Any response, Δ, whose output satisfies to the post-response checks specified in Section 4.1 is a reputable response.*

Per Theorem 2, reputable responses that game more points are better approximations of Δ^*. We compare the performance of Δ^{LD} with that of Δ^{GD} by measuring the proportions of points they game when responding to linear and MLP models trained with ERM on each of our experiment datasets (Table 2). The results show that, in the linear case, Δ^{LD} consistently games more points than Δ^{GD}. Similarly, in the MLP case Δ^{LD} consistently outperforms Δ^{GD} in all but one of the datasets. These results indicate that our Lagrangian response, Δ^{LD}, is reliably a better approximation of Δ^* than the Gradient response.

E.4. Strategic Training

It remains to be shown that Δ^{LD} can improve performance on strategic training. Δ^{LD} can be incorporated into training by plugging it directly the strategic REGD training process proposed in [24]. An alternative is to train a model with a gradient descent-based algorithm, optimising the total derivative instead of the direct derivative (Section 5). In the following results we investigate the consequences of both these approaches.

E.4.1. VISUALISING STRATEGIC TRAINING

The impact of using the total gradient (TGD) instead of the direct gradient (REGD) can be observed in Figure 3. This depicts the results of training an MLP model on a two dimensional dataset comprised of a positive ball surrounded on one side by an equal number of negative points, with the two clusters separated by a narrow margin. The Left pane depicts ERM training without any consideration of the strategic behaviour, which is extremely vulnerable to strategic behaviour (Middle-Left). The Middle-Right pane presents the results from strategic REGD training. The learned decision

Table 3: Strategic Accuracies (%) realised for MLP model trained with different strategic training methods. Strategic accuracies computed with respect to ensemble response evaluated on the respective test datasets. Error bars represent 95% binomial confidence interval. Best accuracy in each comparison is emphasised.

Dataset	ERM	REGD	REGD-LD	TGD
Bank Customer Churn	80.75 ± 0.88	67.55 ± 1.05	81.30 ± 0.87	$\mathbf{85.50 \pm 0.79}$
default of credit card	81.55 ± 0.50	81.32 ± 0.50	82.23 ± 0.49	$\mathbf{82.53 \pm 0.49}$
Employee	74.87 ± 1.42	76.58 ± 1.39	76.80 ± 1.38	$\mathbf{76.91 \pm 1.38}$
GMSC	61.20 ± 0.84	56.42 ± 0.86	60.81 ± 0.84	$\mathbf{64.19 \pm 0.83}$
Houses	93.31 ± 0.39	93.19 ± 0.39	95.13 ± 0.34	$\mathbf{96.00 \pm 0.31}$
HR Analytics	78.84 ± 0.66	$\mathbf{80.25 \pm 0.64}$	78.71 ± 0.66	78.21 ± 0.67
mobile c36	62.18 ± 0.48	63.60 ± 0.47	$\mathbf{85.96 \pm 0.34}$	64.46 ± 0.47
statlog	$\mathbf{71.00 \pm 3.21}$	68.00 ± 3.30	69.50 ± 3.26	$\mathbf{71.00 \pm 3.21}$
Water Quality	60.11 ± 2.09	57.56 ± 2.11	58.29 ± 2.10	$\mathbf{62.30 \pm 2.07}$

boundary does demonstrate improved robustness compared to the ERM result, but is still vulnerable to gaming by the majority of the negative points. In contrast, the model resulting from TGD training (Right) demonstrates considerably greater robustness to gaming behaviour. In particular, we observe that the learner exploits awareness of the gaming potential of positive and negative points, resulting in a decision-boundary with greater distance from the negative points than realised by either other training method.

E.4.2. QUANTITATIVE EVALUATION

Strategic accuracy is used to evaluate the effectiveness of the proposed training methods; an MLP with 2 hidden layers and 128 hidden dimensions per hidden layer is trained with each method. The accuracy of the resulting models is evaluated at test time on strategically perturbed data. In order to produce perturbed data that best approximates Δ^*, we ensemble over the results of applying Δ^{GD} and Δ^{LD} to the data with the post-response checks (Section 4.1). As any point that would be gamed by either response is gamed by the ensemble, this ensemble is still reputable, and is at least as good an approximation to Δ^* as either method individually.

Table 3 compares four training processes: ERM with no awareness of strategic behaviour; strategic REGD with the gradient response; strategic REGD with the Lagrangian dual response (REGD-LD); and TGD computed with the Lagrangian Dual response. TGD consistently achieves higher strategic accuracy as compared with the other training methods, demonstrating that incorporating the total gradient into the model training procedure has a significant impact on the resulting model's robustness to strategic behaviour. Comparing the results from REGD with the gradient response and REGD with the Lagrangian dual response, we observe that, while REGD-LD reasonably consistently outperformed the REGD approach, the inclusion of the more accurate response in training does not account for the entirety of the improvement observed in TGD.

Appendix F. Experimental Detail

In this section we provide greater detail about the experiments run in the main paper. All of the experiments were run on python, using standard pytorch packages to train all the models. The code to reproduce all the experimental results will be released after review. All results are from experiments run on a single GeForce RTX 2080 Ti. However, the models used in these experiments are of a size that they could have been run on most standard laptops.

F.1. Model Training

Two classes of models were used for these experiments; Linear models and Multi-Layer Perceptron (MLP) models. The Linear models were standard linear models of the form $W\mathbf{x} + \mathbf{b}$, where W, the model weights and $\mathbf{b}$, the bias, are learnt during training. The MLP models all had 2 hidden layers, with each hidden layer having 128 dimensions and a final linear layer mapping to a scalar output. A ReLU activation is used between each model layer.

Models were trained on datasets with a 60:20:20 training:validation:test split. Training was run for a maximum of 50 epochs, and early stopping was performed by evaluating the strategic accuracy on the validation set and stopping if no improvement was observed for 10 consecutive epochs. Each epoch of training involved evaluating the strategic response to the current model parameter settings, computing the loss with respect to those strategic values, and then updating the model parameters with respect to that loss. The loss used for our linear results was the hinge loss, while the cross entropy loss was used for training the MLP models. The learning rate used for the training loop was specified per dataset. For more details on this parameter setting see Appendix F.3.

F.2. Response Computation

Three different response methods were evaluated as part of the experiments in this work; the closed form response to the linear model, the Gradient response (Δ^{GD}) and the Lagrangian Dual response (Δ^{LD}). The linear response is parameter free, and was computed according to the definition provided in Equation 16.

Δ^{GD} was computed using standard Gradient ascent-based app on the objective specified in Equation 17. Δ^{LD} was learned using an Augmented Lagrangian-based approach to optimise the Lagrangian as proposed in Equation 8. Both of these responses were optimised with the Adam optimiser with a weight decay specified per dataset. Each response was computed by running the iterative optimiser for 1000 iterations per batch of the training dataset with a learning rate specified per dataset. This number of iterations was deemed sufficient to ensure that both methods had reliably converged and produced plausible gaming behaviour on several datasets. For more details on the setting of the learning rate and the weight decay setting see Appendix F.3.

F.3. Hyperparameter Optimisation

The model based optimisation had a single hyperparameter, the learning rate, that was set on a per-dataset basis. The response optimsation had two hyperparameters to be specified on a per-dataset basis; the learning rate and the weight decay. For each hyperparameter we specified a range of plausible values each could attain. Experiments were run for each parameter setting combination and each combination was scored according to the Learner's objective (strategic accuracy) and the Agents' objective (response objective). We then derived the Stackelberg equilibrium treating the

Learner as the leader with actions (learning rate, weight decay) and the Agents as the follower with action (response learning rate). The resulting parameters were used in all experimental evaluations.

F.4. Implicit Gradient

In practice, the indirect derivative can be approximated efficiently using a truncated Neumann series, as discussed in Lorraine et al. [20]. To compute this derivative, we require a solution to the Lagrangian Dual problem given in Equation 8. In order for training to remain tractable, we aggressively cap the number of iterations used to solve this lower level optimisation problem.

Appendix G. Dataset Details

The results in this work made use of datasets from various sources. This section provides the details required to locate or generate the datasets used in this work. Note that, in the process of training our models for our experiments, the dataset values are standardised using the 'StandardScaler' as part of the python 'sklearn.preprocessing' module. All of the figures presented in this paper, and the results presented, are evaluated on these standardised values.

G.1. Toy Datasets

Figure 1 is produced from synthetic data generated for the example. The dataset is comprised of a set of positive and negative points sampled i.i.d from a Gaussian distribution with fixed variance. This dataset was sourced from [19]. For our results we used a variance of 1.5 for each ball, and and means 10 ± 2.5.

Figure 2 is the standard twin-moons dataset as provided by the 'sklearn' python package.

Figure 3 is realised on a novel dataset; positive points are sampled from a uniform ball with radius 1. Negative points are them sampled uniformly from a positive half-disk of radius 1 with inner radius 2.

G.2. Experiment Datasets

The datasets used to produce the experimental results in this paper are derived from three sources; the majority of the datasets were sampled from the TALENT dataset collection [32], while Give Me Some Credit (GMSC, [8]) and the Houses dataset ([31]) are popular datasets that have previously been used to evaluate results in Strategic or Performative settings.

The GMSC dataset is comprised of historical data relating to borrowers seeking to get a loan from a bank. The motivating objective of the dataset is to predict the probability that a given borrower will default on their loan. [22, 24] have previously used this dataset to derive their results. However, we note, whereas [24] only considered the case where a subset of the dataset features were vulnerable to gaming, the experiments in this paper consider all features equally vulnerable to gaming behaviour.

The Houses dataset is a binarized form of the Housing dataset use in [4]. Specific details pertaining to this dataset can be found at [31].

The TALENT dataset is a collection of datasets used in the evaluation of Deep Learning models on Tabular data [32]. From the set of classification tasks within this collection, we selected a subset where one could plausibly observe gaming behaviour. The datasets used are:

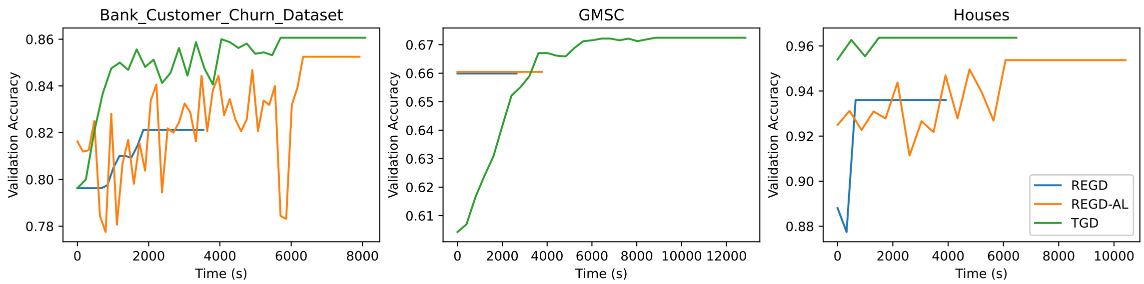

Figure 4: Validation accuracy v.s. runtime (s) for MLP models on subset of TALENT datasets. Models evaluated are;

- Bank Customer Churn Dataset
- Default of Credit Card Clients
- Employee
- HR Analytics Job Change of Data Scientists
- mobile c36 oversampling
- Statlog
- Water Quality and Potability

G.3. Choice of Dataset Radius/Cost Function

In order to elicit gaming behaviour, the ϵ parameter in the cost (Equation 5) is learned for each dataset. This parameter value is chosen as the value that allowed approximately 50% of the points in the dataset to game a linear classifier trained on the clean data.

Appendix H. Proof of Proposition 4

Proof From Section 4.1, if $f_\theta(\Delta(\boldsymbol{x}, \theta)) \neq f_\theta(\boldsymbol{x})$ after the post-response checks, then $f_\theta(\Delta(\boldsymbol{x}, \theta)) > 0$ and $c(\boldsymbol{x}, \Delta(\boldsymbol{x}, \theta)) < 2$. Therefore, $\boldsymbol{x}$ is a gameable point in response to θ, so Δ^* also games it; $f_\theta(\Delta^*(\boldsymbol{x}, \theta)) \neq f_\theta(\boldsymbol{x})$. Hence $f_\theta(\Delta(\boldsymbol{x}, \theta)) \neq f_\theta(\boldsymbol{x}) \implies f_\theta(\Delta^*(\boldsymbol{x}, \theta)) \neq f_\theta(\boldsymbol{x})$ and the checked Δ is a reputable response. ■

H.1. Wall Clock Time for REGD vs. REGD-LD vs. TGD

Figure 4 presents the wall clock times associated with training MLP models on a subset of the TALENT datasets. Times are computed based on experiments run on a single GeForce RTX 2080 Ti. REGD corresponds to the MLP model trained with the strategic REGD with the Gradient response. REGD-AL corresponds to an MLP trained with the strategic REGD using the Lagrangian Dual

Response. TGD corresponds to training using Gradient Descent optimising the total gradient instead of the direct gradient.