Omega: Optimistic EMA Gradients

Juan Ramirez [1 2] **Rohan Sukumaran** [1 2] **Gauthier Gidel** [1 2 3] **Quentin Bertrand** [1 2]

Abstract

Stochastic min-max optimization has gained interest in the machine learning community with the advancements in GANs and adversarial training. Although game optimization is fairly well understood in the deterministic setting, some issues persist in the stochastic regime. Recent work has shown that stochastic gradient descent-ascent methods such as the optimistic gradient are highly sensitive to noise or can fail to converge. Although alternative strategies exist, they can be prohibitively expensive. We introduce Omega, a method with optimistic-like updates that mitigates the impact of noise by incorporating an EMA of historic gradients in its update rule. We also explore a variation of this algorithm that incorporates momentum. Although we do not provide convergence guarantees, our experiments on stochastic games show that Omega outperforms the optimistic gradient method when applied to linear players. Our code is available here[1].

1. Introduction

The recent progress in machine learning can be attributed to being able to reliably minimize (or maximize) objective functions, such as losses or rewards, using gradient-based optimization. Min-max optimization has emerged as particularly relevant for machine learning due to the success of GANs (Goodfellow et al., 2014), adversarial training (Madry et al., 2018), and actor-critic systems (Pfau & Vinyals, 2016). Even though the dynamics of gradient-based methods are well understood for minimization, some issues emerge in the context of saddle point optimization (Gidel et al., 2019). Moreover, the noise introduced by using stochastic estimates of the gradients can exacerbate the issue (Chavdarova et al., 2019).

[1]Mila, Québec AI Institute [2]DIRO, Université de Montréal, Canada [3]Canada CIFAR AI Chair. Correspondence to: Juan Ramirez <juan.ramirez@mila.quebec>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1]Code: https://github.com/juan43ramirez/Omega

Extragradient (Korpelevich, 1976) and extrapolation from the past (Popov, 1980) are popular methods for solving deterministic min-max optimization problems as they enjoy better convergence guarantees to gradient descent-ascent (Gidel et al., 2019). However, their natural extensions to the stochastic setting, independent samples stochastic extragradient (ISEG) (Gorbunov et al., 2022a; Beznosikov et al., 2022) and independent samples optimistic gradient (ISOG) can fail to converge for a simple stochastic bilinear game and are sensitive to noise (Chavdarova et al., 2019). This is concerning for machine learning applications, where performing full batch optimization is prohibitively expensive.

An alternative for ISEG has been considered where the same sample is used during the extrapolation and update steps (Gidel et al., 2019; Mishchenko et al., 2020), and it has been shown to converge under weaker assumptions. For the optimistic gradient method, however, same-sample style updates would require the computation of two gradients per parameter update. Therefore, the same-sample stochastic optimistic gradient (SSOG) is equivalent to same-sample SEG in terms of its computational cost. Other methods such as variance-reduced extragradient mitigate the issues associated with stochasticity but significantly increase the cost of the algorithm (Chavdarova et al., 2019; Alacaoglu & Malitsky, 2022).

We focus on methods that employ optimistic-like updates. We propose *Omega*, a variation of SOG, where an exponential moving average of historic gradients is considered in the update rule. This helps us mitigate the shortcoming of the independent samples SOG approach. More importantly, Omega requires only one gradient computation per parameter update, thereby guaranteeing better computational complexity when compared to SEG. Although we do not provide convergence guarantees for our approach, we demonstrate that Omega outperforms SOG in stochastic bilinear games (see Figure 1) and showcases similar performance to other methods for quadratic games.

Outline: the optimistic gradient method is presented in Section 2. Our proposed method, Omega, and its extension to incorporate momentum are introduced in Section 3. General stochastic quadratic games are described in Section 4. The performance of Omega for bilinear, quadratic and quadratic-linear games is presented in Section 5. Closing remarks and

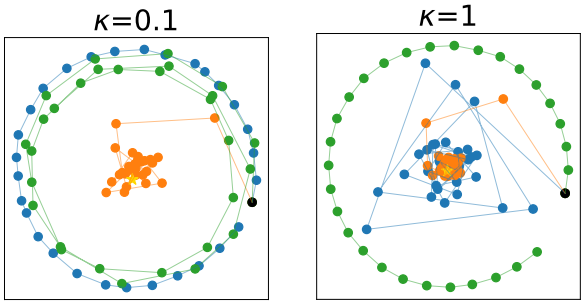

Figure 1. Iterates of stochastic gradient descent-ascent (SGD), the independent samples stochastic optimistic gradient method (ISOG) and Omega for a 2D stochastic bilinear game. κ indicates the conditioning of the optimization problem. SGD diverges, while ISOG and Omega converge. Omega converges faster than ISOG.

future directions are included in Section 6.

2. The Stochastic Optimistic Gradient Method

Consider the unconstrained stochastic saddle point optimization problem stated in Equation (1), where ξ terms represent the stochasticity, x and y represent the value of each player and ℓ refers to the payoff function.

$$\min_{x \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}^{d_y}} \mathcal{L}(x, y) = \mathbb{E}_{\xi}\left[\ell(x, y, \xi)\right] \tag{1}$$

Let $w = [x, y]^\top$, $F_\xi(w) = [\nabla_x \ell(x, y, \xi), -\nabla_y \ell(x, y, \xi)]^\top$. The update scheme of the stochastic optimistic gradient method with *independent samples* (ISOG) is:

$$w_{t+1} = w_t - \eta\Big[(1+\alpha)F_{\xi_t}(w_t) - \alpha F_{\xi_{t-1}}(w_{t-1})\Big] \tag{2}$$

Where η is a step size, α is the optimism hyper-parameter, and ξ_t is a stochastic sample drawn at step t. On each step, the gradient field $F_{\xi_t}(w_t)$ is computed and stored for use on the next iteration. Equation (2) employs independent samples ξ_t and ξ_{t-1} for computing the gradients at w_t and w_{t-1}, respectively.

The deterministic optimistic gradient method converges at a rate of the same order as extragradient (Korpelevich, 1976; Gidel et al., 2019), while only requiring one gradient calculation per parameter update as opposed to two (Gorbunov et al., 2022b). In the stochastic setting, however, independent samples stochastic extragradient (ISEG) and ISOG require stronger assumptions for convergence, like bounded variance of the gradient field (Beznosikov et al., 2022). For extragradient, this assumption is unnecessary if the same stochastic sample is used for the extrapolation and update steps. The same-sample extragradient (SSEG) method does:

$$w_{t+1/2} = w_t - \eta F_{\xi_t}(w_t) \tag{3}$$

$$w_{t+1} = w_t - \eta F_{\xi_t}(w_{t+1/2}) \tag{4}$$

This approach computes the gradient for updating the current iterate w_t at an extrapolated point $w_{t+1/2}$. The look-ahead nature of the updates in Equation (3) is associated with stabilizing the dynamics of the min-max problem (Gidel et al., 2019). Considering the same stochastic sample ξ_t to compute the look-ahead and update steps allows SSEG to converge under weaker assumptions to ISOG (Beznosikov et al., 2023). However, performing the look-ahead step requires the computation of two gradient fields per parameter update as opposed to one.

For optimistic methods, however, same-sample updates are less practical as they represent a computational overhead similar to that of extragradient. A *same-sample* variant of ISOG is presented in Equation (5). Unfortunately, $F_{\xi_t}(w_{t-1})$ must be computed at iteration t as it does not carry over from previous iterations, thus requiring two gradient calculations per parameter update.

$$w_{t+1} = w_t - \eta\Big[(1+\alpha)F_{\xi_t}(w_t) - \alpha F_{\xi_t}(w_{t-1})\Big] \tag{5}$$

Moreover, it has been noted that stochastic min-max optimization is especially sensitive to noise. Chavdarova et al. (2019) propose variance reduced extragradient to mitigate this issue, but their approach requires five gradient computations per parameter update on average. We are interested in an algorithm that: (i) has similar convergence properties to same-sample extragradient, (ii) is robust to noisy estimates of the gradients, and (iii) retains the computational cost of an optimistic method by requiring one gradient field calculation per parameter update.

3. Optimistic EMA gradients

We propose an algorithm with optimistic-like updates for stochastic min-max optimization. We consider an exponential moving average of historic gradients for the correction term in the optimistic update. This is shown in Equation (6):

$$w_{t+1} = w_t - \eta\Big[(1+\alpha)F_{\xi_t}(w_t) - \alpha \tilde{F}_{t-1}\Big] \tag{6}$$

Where $\tilde{F}_t$ is an EMA of previously observed gradients.

$$\tilde{F}_t = (1-\beta)F_{\xi_t}(w_t) + \beta \tilde{F}_{t-1} \tag{7}$$

A new hyper-parameter $\beta \in [0, 1]$ is introduced for decaying the previous EMA value. We initialize $\tilde{F}_0 = F_{\xi_0}(w_0)$ so the first step of Omega corresponds to a gradient descent-ascent step. $\tilde{F}_0$ is not initialized to 0 to prevent later EMA iterates from being biased towards 0.

Note that the use of an EMA reduces the variance of the correction term $\tilde{F}_t$ when compared to $F_{\xi_t}(w_t)$. Thus, Omega is less sensitive to noisy estimates of the gradients than the optimistic gradient method. Moreover, Omega requires one gradient field calculation per parameter update, with

a memory footprint associated with storing $\tilde{F}_t$. Therefore, the cost of Omega matches that of the independent samples optimistic gradient method in Equation (2).

3.1. Optimism and Momentum

Alternatively, the EMA can be employed both for the update direction and the correction term. We refer to this method as optimism with momentum (or OmegaM) given that $\tilde{F}_t$ can be viewed as a momentum term with a dampening of $(1-\beta)$. The updates of this approach are:

$$w_{t+1} = w_t - \eta\Big[(1+\alpha)\tilde{F}_t - \alpha\tilde{F}_{t-1}\Big] \tag{8}$$

Where $\tilde{F}_0 = F_{\xi_0}(w_0)$. It can be seen that the update rule in Equation (8) is equivalent to Equation (6) under a specific choice of α (see Appendix A.1). Section 5 contains experiments with both Omega and Omega with momentum. Appendix A.2 discusses an approach to combine Adam (Kingma & Ba, 2014) and optimism, as previously considered by Gidel et al. (2019).

4. Stochastic Quadratic Games

We consider the following stochastic quadratic game:

$$\mathbb{E}_\xi\left[\frac{1}{2}x^T A_\xi x + a_\xi x + x^T B_\xi y - c_\xi y - \frac{1}{2}y^T C_\xi y\right] \tag{9}$$

Where $x \in \mathbb{R}^{d_x}$ minimizes and $y \in \mathbb{R}^{d_y}$ maximizes. By ensuring that all A_ξ and C_ξ are symmetric positive-definite matrices, Equation (9) is strongly convex in x and strongly concave in y. The bilinear coupling terms B_ξ represent the interaction between both players.

In practice, we generate a fixed set of independent tuples $\{(A_i, B_i, C_i, a_i, c_i)\}_{i=1}^n$, and sample one on each iteration of the algorithm. Given $L_A \geq \mu_A > 0$ and $L_C \geq \mu_C > 0$, the A_i and C_i are sampled such that $\mu_A \preceq A_i \preceq L_A$ and $\mu_C \preceq C_i \preceq L_C$. The B_i are generated so that their smallest singular value is $\mu_B > 0$ and their largest singular value is L_B. The condition numbers of each component of the game are: $\kappa_A = L_A/\mu_A$, $\kappa_B = L_B/\mu_B$, and $\kappa_C = L_C/\mu_C$. Our implementation of stochastic quadratic games is inspired by that of Loizou et al. (2021)

5. Experiments

We solve a series of bilinear, quadratic, and quadratic-linear stochastic games. All of these are derived from Equation (9). The distance to the optimal solution is measured throughout the optimization. The derivation of the optimal solution for a game is presented in Appendix B. More details about the experimental setup and how the games are generated can be found in Appendix C.

5.1. Stochastic Bilinear Game

For bilinear games, we set the terms A and C in Equation (9) to 0. Figure 2 shows the distance to the optimum when using SGD, ISOG, and Omega to solve a 100-dimensional bilinear game. The learning rates for ISOG and Omega are tuned separately to be 0.05 and 0.02, respectively. Although we tried OmegaM on this task, it would diverge for a wide range of learning rates.

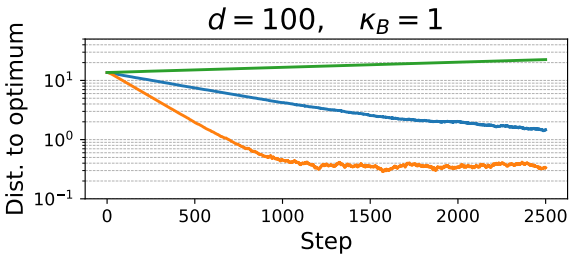

Figure 2. Distance to optimum when solving a 100-dimensional stochastic bilinear game with SGD, ISOG, and Omega ($\beta = 0.9$).Omega ($\beta = 0.9$) reaches the optimum faster than ISOG while SGD diverges.

We observe that SGD diverges while ISOG and Omega get close to the optimum and then oscillate in a neighborhood around it. Given that the optimization problem is stochastic and we use a constant step size, these oscillations are expected. In this experiment, Omega approaches the optimum faster than ISOG.

We carry out a sensitivity analysis over the EMA decay hyper-parameter β. The results are presented in Figure 3. Note that the experiment with $\beta = 0$ corresponds to ISOG. Generally, large EMA values outperform smaller values. However, using a very large β is detrimental as it introduces oscillations and iterates get stuck far from the optimum. Overall, we found $\beta = 0.9$ to be a reasonable choice and used it throughout the rest of our work.

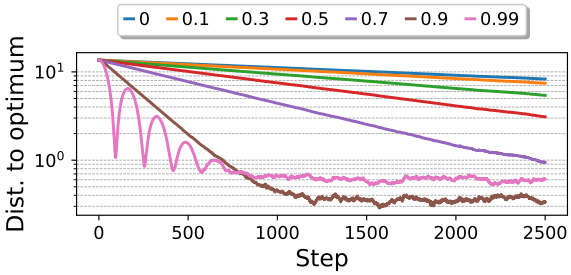

Figure 3. Training dynamics of Omega when solving a 100-dimensional bilinear game for different choices of the EMA decay hyper-parameter β. We use a learning rate of 0.02. We notice that an EMA of 0.99 produces larger oscillations compared to the others. An EMA of 0.9 seemed to be the best choice from the experiment, for the given setup

Table 1 presents the distance to the optimum after 1000 steps for different choices of batch size. The learning rates for ISOG and Omega are tuned independently. Overall, both methods improve when the effect of stochasticity is mitigated by considering larger batch sizes. Notably, Omega performs well in the deterministic bilinear game.

Batch Size	1	5	10	20	100
ISOG	4.224	3.941	3.876	3.916	3.900
Omega	0.458	0.323	0.291	0.281	0.280

Table 1. Distance to the optimum after 1000 steps for different batch size choices. A batch size of 100 corresponds to a full batch.

5.2. Stochastic Quadratic Game

Figure 4 shows the distance to the optimum for different optimization methods when applied to a stochastic quadratic game with good conditioning. Table 2 reports the distance to the optimum after 500 steps for problems with different conditioning. We identify that a learning rate of 0.01 yields good performance for all optimization approaches.

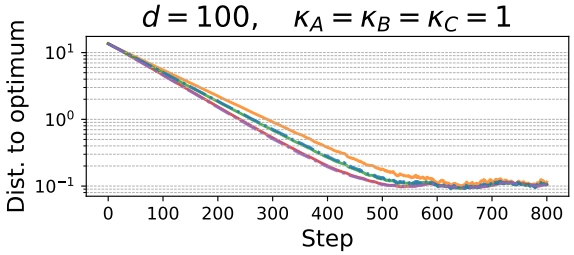

Figure 4. Distance to optimum for SGD, ISOG, Omega, SGDM, and OmegaM when solving a 100-dimensional stochastic quadratic game. For the latter 3 methods, we choose $\beta = 0.9$. We can see that, given the nature of quadratic games, all algorithms converge with similar behavior.

All methods converge to a neighborhood of the optimum at a linear rate. In particular, SGDM and OmegaM yield slightly faster convergence. Although momentum can be detrimental to min-max optimization, our experiments show quadratic games as benefiting from acceleration. This could be related to the quadratic component dominating over the bilinear term and making the optimization problem easier. For quadratic games, Omega slows down progress when compared to ISOG and SGD. Similar trends are seen when considering worse conditioning for A and C.

κ_A, κ_C	κ_B	SGD	ISOG	Omega	SGDM	OmegaM
1	1	0.119	0.125	**0.112**	**0.112**	**0.112**
1	10	0.128	0.129	0.135	0.180	**0.106**
10	1	0.963	0.967	**1.108**	**0.105**	0.116
10	10	0.915	0.908	1.035	**0.777**	0.820

Table 2. Distance to the optimum at 500 steps for stochastic quadratic games with different conditioning.

5.3. Quadratic-Linear Games

This section considers a game that is quadratic on one player, but linear on the other player. Min-max optimization with one linear player is relevant for the Lagrangian-based constrained optimization literature (Elenter et al., 2022), as the Lagrangian is always a linear function of the dual variables. Given that Omega performs well in bilinear games but under-performs in quadratic games, we fix SGD with a step size of 0.02 for the quadratic player and experiment with the choice of optimizer for the linear player.

Figure 5 shows the distance to the optimum when solving a quadratic-linear game and using SGD, ISOG, and Omega for the linear player, all with a step size of 0.01. While SGD and ISOG perform very similarly, the Omega experiment approaches the optimum at a slightly faster rate.

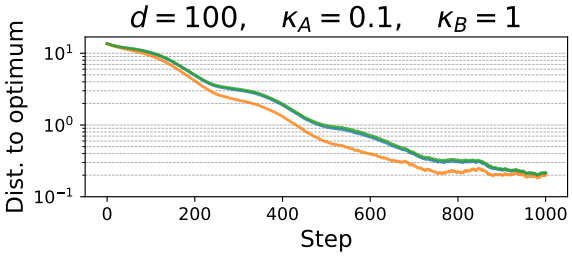

Figure 5. Distance to optimum when solving a 100-dimensional stochastic quadratic-linear game. SGD is used for the quadratic player, and SGD, ISOG, and Omega for the linear player. We can notice that Omega makes progress toward the optimum faster than the other methods.

Table 3 reports the distance to the optimum after 500 steps for problems with different conditioning. Omega generally approaches the optimum faster than SGD and ISOG. This is most notorious when the conditioning of the quadratic term is relatively poor ($\kappa_A = 10$, $\kappa_B = 1$).

κ_A	κ_B	SGD	ISOG	Omega
1	1	0.225	0.221	0.176
1	10	2.691	2.689	2.664
10	1	0.990	0.941	0.587
10	10	1.247	1.242	1.198

Table 3. Distance to the optimum at 500 steps for stochastic quadratic-linear games with different conditioning.

6. Conclusion

We consider algorithms for stochastic min-max optimization. Given that extragradient methods are computationally expensive, we focus on updates based on optimism. However, as independent samples SOG can fail to converge in some settings, and same-sample SOG is as expensive as stochastic extragradient, we propose Omega, a simple modification of the optimistic gradient method. By considering an EMA

of past gradients for the update, we obtain an algorithm that has the computational requirements of an optimistic method and that is competitive on a set of toy experiments. In particular, Omega outperforms the optimistic gradient method when applied to linear players in stochastic games.

An important step for future work is to analyze the convergence properties of Omega. Moreover, it would interesting to evaluate Omega on a machine learning task such as training of GANs (Goodfellow et al., 2014).

Acknowledgements

We would like to thank Golnoosh Farnadi for her suggestions for the writing of the paper. This research was enabled in part by compute resources, software, and technical help provided by Mila (mila.quebec). We would also like to thank the anonymous reviewers of the LXAI workshop for their constructive comments.

References

Alacaoglu, A. and Malitsky, Y. Stochastic Variance Reduction for Variational Inequality Methods. *CLT*, 2022.

Beznosikov, A., Polyak, B., Gorbunov, E., Kovalev, D., and Gasnikov, A. Smooth Monotone Stochastic Variational Inequalities and Saddle Point Problems–Survey. *arXiv preprint arXiv:2208.13592*, 2022.

Beznosikov, A., Gorbunov, E., Berard, H., and Loizou, N. Stochastic Gradient Descent-Ascent: Unified Theory and New Efficient Methods. *AISTATS*, 2023.

Chavdarova, T., Gidel, G., Fleuret, F., and Lacoste-Julien, S. Reducing Noise in GAN Training with Variance Reduced Extragradient. *NeurIPS*, 2019.

Elenter, J., NaderiAlizadeh, N., and Ribeiro, A. A Lagrangian Duality Approach to Active Learning. *NeurIPS*, 2022.

Gidel, G., Berard, H., Vignoud, G., Vincent, P., and Lacoste-Julien, S. A Variational Inequality Perspective on Generative Adversarial Networks. *ICLR*, 2019.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Networks. *NeurIPS*, 2014.

Gorbunov, E., Berard, H., Gidel, G., and Loizou, N. Stochastic Extragradient: General Analysis and Improved Rates. In *AISTATS*, 2022a.

Gorbunov, E., Taylor, A., and Gidel, G. Last-Iterate Convergence of Optimistic Gradient Method for Monotone Variational Inequalities. *NeurIPS*, 2022b.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. *ICLR*, 2014.

Korpelevich, G. M. The extragradient method for finding saddle points and other problems. *Matecon*, 1976.

Loizou, N., Berard, H., Gidel, G., Mitliagkas, I., and Lacoste-Julien, S. Stochastic Gradient Descent-Ascent and Consensus Optimization for Smooth Games: Convergence Analysis under Expected Co-coercivity. *NeurIPS*, 2021.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. *ICLR*, 2018.

Mishchenko, K., Kovalev, D., Shulgin, E., Richtárik, P., and Malitsky, Y. Revisiting Stochastic Extragradient. In *AISTATS*, 2020.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library, 2019.

Pfau, D. and Vinyals, O. Connecting Generative Adversarial Networks and Actor-Critic Methods. *NeurIPS Workshop on Adversarial Training*, 2016.

Popov, L. D. A modification of the arrow-hurwicz method for search of saddle points. *Mathematical notes of the Academy of Sciences of the USSR*, 1980.

Schmidt, M. Convergence rate of stochastic gradient with constant step size. 2014.

A. Optimistic-like Updates.

A.1. Omega with Momentum is Equivalent to Omega

It can be seen that the update rule in Equation (8) is equivalent to Equation (6) under a specific choice of the optimism hyper-parameter α. By replacing $\tilde{F}_t$ with its definition, given in Equation (7), it follows that:

$$
\begin{aligned}
w_{t+1} &= w_t - \eta\Big[\tilde{F}_t + \alpha\big(\tilde{F}_t - \tilde{F}_{t-1}\big)\Big] && (10)\\
&= w_t - \eta\Big[(1-\beta)F_{\xi_t}(w_t) + \beta\tilde{F}_{t-1} + \alpha\big[(1-\beta)F_{\xi_t}(w_t) + \beta\tilde{F}_{t-1} - \tilde{F}_{t-1}\big]\Big] && (11)\\
&= w_t - \eta\Big[(1-\beta)F_{\xi_t}(w_t) + \beta\tilde{F}_{t-1} + \alpha(1-\beta)\big[F_{\xi_t}(w_t) - \tilde{F}_{t-1}\big]\Big] && (12)\\
&= w_t - \eta\Big[F_{\xi_t}(w_t) - \beta\big[F_{\xi_t}(w_t) - \tilde{F}_{t-1}\big] + \alpha(1-\beta)\big[F_{\xi_t}(w_t) - \tilde{F}_{t-1}\big]\Big] && (13)\\
&= w_t - \eta\Big[F_{\xi_t}(w_t) + (\alpha(1-\beta) - \beta)\big[F_{\xi_t}(w_t) - \tilde{F}_{t-1}\big]\Big] && (14)
\end{aligned}
$$

Which is equivalent to Omega with a choice of optimism of $\alpha(1-\beta) - \beta$ and with the same EMA coefficient.

A.2. Optimism and Adam

Adam (Kingma & Ba, 2014) is a popular method for stochastic minimization that has been shown to work well for min-max problems such as training Generative Adversarial Networks (Goodfellow et al., 2014). Adam updates can be written as:

$$
\begin{aligned}
\tilde{F}_t &= (1-\beta)F_{\xi_t}(w_t) + \beta\tilde{F}_{t-1} && (15)\\
v_t &= (1-\gamma)F_{\xi_t}(w_t)^2 + \gamma v_{t-1} && (16)\\
w_t &= w_{t-1} - \eta\,\frac{\tilde{F}_t}{\sqrt{v_t} + \epsilon} && (17)
\end{aligned}
$$

Where v_t is an EMA on the square of the gradients with coefficient $\gamma \in [0, 1]$, and $\epsilon > 0$ is a small number. The usual Adam updates apply a correction to the first and second order estimates to account for their bias towards 0.

$$
\hat{F}_t = \tilde{F}_t/(1-\beta^t) \quad \hat{v}_t = v_t/(1-\gamma^t) \tag{18}
$$

We disregard this correction in our presentation of the updates in Equation (17) as we choose to initialize $\tilde{F}_0 = F_{\xi_0}(w_0)$ and $v_0 = F_{\xi_0}(w_0)^2$, as opposed to starting both at 0.

Combining optimistic-like updates with Adam yields the following update rule:

$$
w_t = w_{t-1} - \eta\left[\frac{\tilde{F}_t}{\sqrt{v_t} + \epsilon} + \alpha\left(\frac{\tilde{F}_t}{\sqrt{v_t} + \epsilon} - \frac{\tilde{F}_{t-1}}{\sqrt{v_{t-1}} + \epsilon}\right)\right] \tag{19}
$$

This formulation already considers EMAs on the gradients. As such, optimistic Adam does not need to be extended to Omega-style updates. Note that the pairing of optimistic updates with Adam presented in Equation (19) has been considered in the past by (Gidel et al., 2019).

B. Stochastic Games

B.1. Solving the Stochastic Quadratic Game

The stochastic quadratic game is given by:

$$
\min_{x\in\mathbb{R}^{d_x}} \max_{y\in\mathbb{R}^{d_y}} \mathbb{E}_\xi\left[\frac{1}{2}x^T A_\xi x + a_\xi x + x^T B_\xi y - c_\xi y - \frac{1}{2}y^T C_\xi y\right] \tag{20}
$$

Equation (20) represents a strongly convex-strongly concave game if $\mathbb{E}_\xi[A_\xi]$ and $\mathbb{E}_\xi[C_\xi]$ are positive definite matrices with their smallest eigenvalues being $\mu_A > 0$ and $\mu_C > 0$, respectively. In such a setting, the min-max optimization problem has

a unique Nash equilibrium. Given a dataset $\{(A_i, B_i, C_i, a_i, c_i)\}_{i=1}^n$, the Nash equilibrium can be derived analytically by considering:

$$\bar{A} = \frac{1}{n}\sum_{i=1}^{n} A_i \qquad \bar{B} = \frac{1}{n}\sum_{i=1}^{n} B_i \qquad \bar{C} = \frac{1}{n}\sum_{i=1}^{n} C_i \tag{21}$$

$$\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i \qquad \bar{c} = \frac{1}{n}\sum_{i=1}^{n} c_i \tag{22}$$

And finding a stationary point of Equation (20) by solving the following linear system:

$$\begin{bmatrix} \bar{A} & \bar{B} \\ -\bar{B}^\top & \bar{C} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \bar{a} \\ \bar{c} \end{bmatrix} = 0 \tag{23}$$

B.2. Solving the Stochastic Bilinear Game

The stochastic bilinear game is given by:

$$\min_{x \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}^{d_y}} \mathbb{E}_\xi \left[a_\xi x + x^T B_\xi y - c_\xi y \right] \tag{24}$$

Consider a dataset $\{(B_i, a_i, c_i)\}_{i=1}^n$. Moreover, let:

$$\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i \qquad \bar{B} = \frac{1}{n}\sum_{i=1}^{n} B_i \qquad \bar{c} = \frac{1}{n}\sum_{i=1}^{n} c_i \tag{25}$$

The first order necessary conditions for the problem in Equation (24) are $\bar{B}y + \bar{a} = 0$ and $x^\top \bar{B} + \bar{c} = 0$. If, in addition, $\bar{B}$ has full rank, each system of equations has a unique solution.

Contribution of LatinX Individuals

The first author of this work is a LatinX individual.

C. Experimental Details

We use Pytorch 1.13 (Paszke et al., 2019). Our implementation of the optimistic gradient method is a slight modification of that provided by Gidel et al. (2019)[2]. Our code for Omega is inspired by their implementation of optimism.

C.1. Games

We perform experiments on bilinear, quadratic and quadratic-linear games. The default hyper-parameters for these problems are presented in Table 4. We focus on games where both player vectors have 100 dimensions. We consider 100 stochastic samples for the matrices and vectors of the game.

Game	d_x	d_y	n	μ_A	L_A	μ_B	L_B	μ_C	L_C
Bilinear	100	100	100	-	-	1	1	-	-
Quadratic	100	100	100	1	1	1	1	1	1
Quadratic-Linear	100	100	100	1	1	1	1	-	-

Table 4. Default game hyper-parameters.

[2]Optimistic gradient method code available here: https://github.com/GauthierGidel/Variational-Inequality-GAN

C.2. Generating the Stochastic Games

We generate a fixed set of independent tuples $\{(A_i, B_i, C_i, a_i, c_i)\}_{i=1}^n$ to sample from during each iteration of the algorithm. Given $L_A \geq \mu_A > 0$ and $L_C \geq \mu_C > 0$, we generate A_i and C_i such that:

$$\mu_A \cdot \mathbb{I}_{d_x} \preceq A_i \preceq L_A \cdot \mathbb{I}_{d_x} \tag{26}$$

$$\mu_C \cdot \mathbb{I}_{d_y} \preceq C_i \preceq L_C \cdot \mathbb{I}_{d_y} \tag{27}$$

The B_i are generated so that their smallest singular value is $\mu_B > 0$ and their largest singular value is L_B. We consider the following condition numbers throughout our work:

$$\kappa_A = L_A/\mu_A, \quad \kappa_B = L_B/\mu_B, \quad \kappa_C = L_C/\mu_C \tag{28}$$

For bilinear games, we set $A_i = C_i = 0$. Each entry in a_i is drawn independently from a Gaussian distribution with mean 0 and variance $1/d_x$. The c_i terms are drawn analogously.

The initial guess for the algorithm (x_0, y_0) is initialized from an isotropic Gaussian distribution with unit variance.

C.3. Optimization Hyper-Parameters

Different optimization algorithms are considered for solving stochastic games: gradient descent-ascent (SGD), gradient descent-ascent with momentum (SGDA, see Appendix C.4), the independent samples optimistic gradient method (ISOG) from Equation (2), Omega, and Omega with momentum from Equation (8). We refer to Omega with momentum as *OmegaM*.

For ISOG, Omega, and OmegaM, we set the optimism hyper-parameter to 1. For the approaches with EMAs, namely, SGDM, Omega, and OmegaM, we employ a decay hyper-parameter β of 0.9. For all optimization methods, we employ simultaneous updates. Experiments are run for 5000 steps with a batch size of 1. We only consider one random seed per experiment. Although a constant step size for stochastic optimization allows for convergence up to a neighborhood of the optimal point (Schmidt, 2014), our experiments do not employ decreasing step sizes.

C.4. Our Implementation of SGD with Momentum

We consider an implementation of SGD with momentum which incorporates dampening:

$$d_t = \beta d_{t-1} + (1-\tau) F_{\xi_{t-1}}(w_{t-1}) \tag{29}$$

$$w_{t+1} = w_t - \eta d_t \tag{30}$$

Where β is the momentum coefficient and τ is the dampening. By letting $\beta \in [0, 1]$ and $\tau = \beta$, we recover that updates are performed in the direction of an EMA of historic gradients:

$$\begin{aligned} \tilde{F}_t &= \beta \tilde{F}_{t-1} + (1-\beta) F_{\xi_t}(w_t) \\ w_{t+1} &= w_t - \eta \tilde{F}_t \end{aligned} \tag{31}$$

For ease of comparison with Omega and Omega with momentum, our experiments with SGD with momentum consider the formulation presented in Equation (31).