Supplementary material for Discretization Drift in Two-Player Games

Contents

A. Proof of the main theorems

Notation: We use $\boldsymbol{\phi} \in \mathbb{R}^m$ to denote the parameters of the first player and $\boldsymbol{\theta} \in \mathbb{R}^n$ for the second player. We assume that the players parameters are updated simultaneously with learning rates αh and λh respectively. We consider the vector fields $f(\boldsymbol{\phi}, \boldsymbol{\theta}) : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$ and $g(\boldsymbol{\phi}, \boldsymbol{\theta}) : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$. Unless otherwise specified, we assume that vectors are row vectors. We denote $\nabla_{\mathbf{x}} f$ the transpose of the Jacobian of f with respect to $\mathbf{x} \in \{\boldsymbol{\phi}, \boldsymbol{\theta}\}$, and similarly for g. Thus $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\phi}, \boldsymbol{\theta}) \in \mathbb{R}^{n,m}$ denotes the matrix with entries $\left[\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\phi}, \boldsymbol{\theta})\right]_{i,j} = \frac{df_j}{d\boldsymbol{\theta}_i}$ with $i \in \{1, ..., n\}, j \in \{1, ..., m\}$. We use bold notation to denote vectors — $\mathbf{x}$ is a vector while x is a scalar.

We now prove Theorem 3.1 and Theorem 3.2 in the main paper, corresponding to the *simultaneous* Euler updates and to the *alternating* Euler updates, respectively. In both cases, our goal is to find corrections f_1 and g_1 to the original system

$$\dot{\boldsymbol{\phi}} = f(\boldsymbol{\phi}, \boldsymbol{\theta}), \tag{A.1}$$

$$\dot{\boldsymbol{\theta}} = g(\boldsymbol{\phi}, \boldsymbol{\theta}), \tag{A.2}$$

such that the modified continuous system

$$\dot{\boldsymbol{\phi}} = f(\boldsymbol{\phi}, \boldsymbol{\theta}) + h f_1(\boldsymbol{\phi}, \boldsymbol{\theta}), \tag{A.3}$$

$$\dot{\boldsymbol{\theta}} = g(\boldsymbol{\phi}, \boldsymbol{\theta}) + h g_1(\boldsymbol{\phi}, \boldsymbol{\theta}), \tag{A.4}$$

follows the discrete steps of the method with a local error of order $\mathcal{O}(h^3)$. More precisely, if $(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t)$ denotes the discrete step of the method at time t, and $(\tilde{\boldsymbol{\phi}}(s), \tilde{\boldsymbol{\theta}}(s))$ corresponds to the continuous solution of the modified system above starting at $(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1})$, we want that the local errors for both players, i.e.,

$$\|\boldsymbol{\phi}_t - \tilde{\boldsymbol{\phi}}(\alpha h)\| \qquad \text{and} \qquad \|\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}(\lambda h)\|$$

to be of order $\mathcal{O}(h^3)$. In the expression above, αh and λh are the effective learning rates (or step-sizes) for the first and the second player respectively for both the simultaneous and alternating Euler updates.

Our proofs from backward error analysis follow the same steps:

1. Expand the discrete updates to find a relationship between $\boldsymbol{\phi}_t$ and $\boldsymbol{\phi}_{t-1}$ and $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_{t-1}$ up to order $\mathcal{O}(h^2)$.
2. Expand the changes in continuous time of the modified ODE given by backward error analysis.
3. Find the first order Discretization Drift (DD) by matching the discrete and continuous updates up to second order in learning rates.

Notation: To avoid cluttered notations, we use $f_{(t)}$ to denote the $f(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t)$ and $g_{(t)}$ to denote $g(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t)$ for all t. If no index is specified, we use f to denote $f(\boldsymbol{\phi}, \boldsymbol{\theta})$, where $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are variables in a continuous system.

A.1. Simultaneous updates (Theorem 3.1)

Here we prove Theorem 3.1 in the main paper, which we reproduce here:

The simultaneous Euler updates with learning rates αh and λh respectively are given by:

$$\boldsymbol{\phi}_t = \boldsymbol{\phi}_{t-1} + \alpha h f(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) \tag{A.5}$$

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \lambda h g(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) \tag{A.6}$$

Theorem 3.1 *The discrete* simultaneous *Euler updates in* (A.5) *and* (A.6) *follow the continuous system*

$$\dot{\boldsymbol{\phi}} = f - \frac{\alpha h}{2}\left(f \nabla_{\boldsymbol{\phi}} f + g \nabla_{\boldsymbol{\theta}} f\right)$$

$$\dot{\boldsymbol{\theta}} = g - \frac{\lambda h}{2}\left(f \nabla_{\boldsymbol{\phi}} g + g \nabla_{\boldsymbol{\theta}} g\right)$$

with an error of size $\mathcal{O}(h^3)$ after one update step.

Step 1: Expand the updates per player via a Taylor expansion.
We expand the numerical scheme to find a relationship between $\boldsymbol{\phi}_t$ and $\boldsymbol{\phi}_{t-1}$ and $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_{t-1}$ up to order $\mathcal{O}(h^2)$. Here in the case of the simultaneous Euler updates, this does not require any change to Equations (A.5) and (A.6). For the first player, the discrete Euler updates are:

$$\boldsymbol{\phi}_t = \boldsymbol{\phi}_{t-1} + \alpha h f(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) \tag{A.7}$$

For the second player, the discrete Euler update has the same form:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \lambda h g(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) \tag{A.8}$$

Step 2: Expand the continuous time changes for the modified ODE given by backward error analysis
We expand the changes in time of the modified ODE of the form:

$$\begin{aligned}\dot{\boldsymbol{\phi}} &= \tilde{f}(\boldsymbol{\phi}, \boldsymbol{\theta})\\ \dot{\boldsymbol{\theta}} &= \tilde{g}(\boldsymbol{\phi}, \boldsymbol{\theta})\end{aligned}$$

where

$$\begin{aligned}\tilde{f}(\boldsymbol{\phi}, \boldsymbol{\theta}) &= f(\boldsymbol{\phi}, \boldsymbol{\theta}) + \sum_{i=1} {\tau_{\boldsymbol{\phi}}}^i f_i(\boldsymbol{\phi}, \boldsymbol{\theta})\\ \tilde{g}(\boldsymbol{\phi}, \boldsymbol{\theta}) &= g(\boldsymbol{\phi}, \boldsymbol{\theta}) + \sum_{i=1} {\tau_{\theta}}^i g_i(\boldsymbol{\phi}, \boldsymbol{\theta})\end{aligned}$$

our aim is to find f_i and g_i which match the discrete updates we have found above.

Lemma A.1 *If:*

$$\begin{aligned}\dot{\boldsymbol{\phi}} &= \tilde{f}(\boldsymbol{\phi}, \boldsymbol{\theta})\\ \dot{\boldsymbol{\theta}} &= \tilde{g}(\boldsymbol{\phi}, \boldsymbol{\theta})\end{aligned}$$

where

$$\begin{aligned}\tilde{f}(\boldsymbol{\phi}, \boldsymbol{\theta}) &= f(\boldsymbol{\phi}, \boldsymbol{\theta}) + \sum_{i=1} {\tau_{\phi}}^i f_i(\boldsymbol{\phi}, \boldsymbol{\theta})\\ \tilde{g}(\boldsymbol{\phi}, \boldsymbol{\theta}) &= g(\boldsymbol{\phi}, \boldsymbol{\theta}) + \sum_{i=1} {\tau_{\theta}}^i g_i(\boldsymbol{\phi}, \boldsymbol{\theta})\end{aligned}$$

and τ_θ and τ_ϕ are scalars. Then — for ease of notation, we drop the argument τ on the evaluations of $\boldsymbol{\phi}$ and θ on the RHS:

$$\boldsymbol{\phi}(\tau + \tau_\phi) = \boldsymbol{\phi}(\tau) + \tau_\phi f + \tau_\phi^2 f_1 + \tau_\phi^2 \frac{1}{2} f \nabla_{\boldsymbol{\phi}} f + \tau_\phi^2 \frac{1}{2} g \nabla_{\boldsymbol{\theta}} f + \mathcal{O}(\tau_\phi^3)$$

Proof: We expand to see what happens after 1 time step of τ_ϕ by doing a Taylor expansion:

$$
\begin{aligned}
\boldsymbol{\phi}(\tau + \tau_\phi) &= \boldsymbol{\phi} + \tau_\phi \dot{\boldsymbol{\phi}} + \tau_\phi^2 \frac{1}{2} \ddot{\boldsymbol{\phi}} + \mathcal{O}(\tau_\phi^3) \\
&= \boldsymbol{\phi} + \tau_\phi \tilde{f} + \tau_\phi^2 \frac{1}{2} \dot{\tilde{f}} + \mathcal{O}(\tau_\phi^3) \\
&= \boldsymbol{\phi} + \tau_\phi \tilde{f} + \tau_\phi^2 \frac{1}{2} \left(\tilde{f} \nabla_{\boldsymbol{\phi}} \tilde{f} + \tilde{g} \nabla_{\boldsymbol{\theta}} \tilde{f} \right) + \mathcal{O}(\tau_\phi^3) \\
&= \boldsymbol{\phi} + \tau_\phi (f + \tau_\phi f_1 + \mathcal{O}(\tau_\phi^2)) + \tau_\phi^2 \frac{1}{2} \left(\tilde{f} \nabla_{\boldsymbol{\phi}} \tilde{f} + \tilde{g} \nabla_{\boldsymbol{\theta}} \tilde{f} \right) + \mathcal{O}(\tau_\phi^3) \\
&= \boldsymbol{\phi} + \tau_\phi f + \tau_\phi^2 f_1 + \tau_\phi^2 \frac{1}{2} \left(\tilde{f} \nabla_{\boldsymbol{\phi}} \tilde{f} + \tilde{g} \nabla_{\boldsymbol{\theta}} \tilde{f} \right) + \mathcal{O}(\tau_\phi^3) \\
&= \boldsymbol{\phi} + \tau_\phi f + \tau_\phi^2 f_1 + \tau_\phi^2 \frac{1}{2} \left(\left(f + \tau_\phi f_1 + \mathcal{O}(\tau_\phi^2) \right) \nabla_{\boldsymbol{\phi}} \tilde{f} + \left(g + \tau_\theta g_1 + \mathcal{O}(\tau_\theta^2) \right) \nabla_{\boldsymbol{\theta}} \tilde{f} \right) + \mathcal{O}(\tau_\phi^3) \\
&= \boldsymbol{\phi} + \tau_\phi f + \tau_\phi^2 f_1 + \tau_\phi^2 \frac{1}{2} f \nabla_{\boldsymbol{\phi}} \tilde{f} + \tau_\phi^2 \frac{1}{2} g \nabla_{\boldsymbol{\theta}} \tilde{f} + \mathcal{O}(\tau_\phi^3) \\
&= \boldsymbol{\phi} + \tau_\phi f + \tau_\phi^2 f_1 + \tau_\phi^2 \frac{1}{2} f \nabla_{\boldsymbol{\phi}} f + \tau_\phi^2 \frac{1}{2} g \nabla_{\boldsymbol{\theta}} f + \mathcal{O}(\tau_\phi^3)
\end{aligned}
$$

where we assumed that τ_ϕ and τ_θ are in the same order of magnitude. □

Step 3: Matching the discrete and modified continuous updates.
From Lemma A.1, we model how the continuous updates of the two players change in time for the modified ODEs given by backward error analysis. To do so, we substitute the *current values* as those given by the discrete updates, namely $\boldsymbol{\phi}_{t-1}$ and $\boldsymbol{\theta}_{t-1}$, in order to calculate the displacement according to the continuous updates:

$$\boldsymbol{\phi}(\tau + \tau_\phi) = \boldsymbol{\phi}_{t-1} + \tau_\phi f_{(t-1)} + \tau_\phi^2 f_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) + \frac{1}{2} \tau_\phi^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \frac{1}{2} \tau_\phi^2 g_{(t-1)} \nabla_\theta f_{(t-1)} + \mathcal{O}(\tau_\phi^3) \tag{A.9}$$

$$\boldsymbol{\theta}(\tau + \tau_\theta) = \boldsymbol{\theta}_{t-1} + \tau_\theta g_{(t-1)} + \tau_\theta^2 g_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) + \frac{1}{2} \tau_\theta^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \frac{1}{2} \tau_\theta^2 g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} + \mathcal{O}(\tau_\theta^3) \tag{A.10}$$

In order to find f_1 and g_1 such that the continuous dynamics of the modified updates $f + \alpha h f_1$ and $g + \lambda h g_1$ match the discrete updates in Equations (A.7) and (A.8), we look for the corresponding continuous increments of the discrete updates in the modified continuous system, such that $\|\boldsymbol{\phi}(\tau + \tau_\phi) - \boldsymbol{\phi}_t\|$ and $\|\boldsymbol{\theta}(\tau + \tau_\theta) - \boldsymbol{\theta}_t\|$ are $\mathcal{O}(h^3)$.

The first order terms in Equations (A.7) and (A.8) and those in Equations (A.9) and (A.10) suggest that:

$$
\begin{aligned}
\alpha h &= \tau_\phi \\
\lambda h &= \tau_\theta
\end{aligned}
$$

We can now proceed to find f_1 and g_1 from the second order terms:

$$
\begin{aligned}
0 &= \alpha^2 h^2 f_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) + \frac{1}{2} \alpha^2 h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \frac{1}{2} \alpha^2 h^2 g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)} \\
f_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) &= -\frac{1}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} - \frac{1}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)}
\end{aligned}
$$

Similarly, for g_1 we obtain:

$$g_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) = -\frac{1}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} - \frac{1}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)}$$

Leading to the first order corrections:

$$f_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) = -\frac{1}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} - \frac{1}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)} \tag{A.11}$$

$$g_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) = -\frac{1}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} - \frac{1}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} \tag{A.12}$$

We have found the functions f_1 and g_1 such that after one discrete optimization step the ODEs $\dot{\boldsymbol{\phi}} = f + \alpha h f_1$ and $\dot{\boldsymbol{\theta}} = g + \lambda h g_1$ follow the discrete updates up to order $\mathcal{O}(h^3)$, finishing the proof.

A.2. Alternating updates (Theorem 3.2)

Here we prove Theorem 3.2 in the main paper, which we reproduce here:

For the *alternating Euler updates*, the players take turns to update their parameters, and can perform multiple updates each. We denote the number of alternating updates of the first player (resp. second player) by m (resp. k). We scale the learning rates by the number of updates, leading to the following updates $\boldsymbol{\phi}_t := \boldsymbol{\phi}_{m,t}$ and $\boldsymbol{\theta}_t := \boldsymbol{\theta}_{k,t}$ where

$$\boldsymbol{\phi}_{i,t} = \boldsymbol{\phi}_{i-1,t} + \frac{\alpha h}{m} f(\boldsymbol{\phi}_{i-1,t}, \boldsymbol{\theta}_{t-1}), \quad i = 1 \dots m, \tag{A.13}$$

$$\boldsymbol{\theta}_{j,t} = \boldsymbol{\theta}_{j-1,t} + \frac{\lambda h}{k} g(\boldsymbol{\phi}_{m,t}, \boldsymbol{\theta}_{j-1,t}), \quad j = 1 \dots k. \tag{A.14}$$

Theorem 3.2 *The discrete* alternating *Euler updates in* (A.13) *and* (A.14) *follow the continuous system*

$$\dot{\boldsymbol{\phi}} = f - \frac{\alpha h}{2} \left(\frac{1}{m} f \nabla_{\boldsymbol{\phi}} f + g \nabla_{\boldsymbol{\theta}} f \right)$$
$$\dot{\boldsymbol{\theta}} = g - \frac{\lambda h}{2} \left((1 - \frac{2\alpha}{\lambda}) f \nabla_{\boldsymbol{\phi}} g + \frac{1}{k} g \nabla_{\boldsymbol{\theta}} g \right)$$

with an error of size $\mathcal{O}(h^3)$ after one update step.

Step 1: Discrete updates
In the case of alternating Euler discrete updates, we have:

$$\begin{aligned}
\boldsymbol{\phi}_{1,t} &= \boldsymbol{\phi}_{t-1} + \frac{\alpha}{m} h f(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) \\
\boldsymbol{\phi}_{2,t} &= \boldsymbol{\phi}_{1,t} + \frac{\alpha}{m} h f(\boldsymbol{\phi}_{1,t}, \boldsymbol{\theta}_{t-1}) \\
&\dots \\
\boldsymbol{\phi}_{m,t} &= \boldsymbol{\phi}_{m-1,t} + \frac{\alpha}{m} h f(\boldsymbol{\phi}_{m-1,t}, \boldsymbol{\theta}_{t-1}) \\
\boldsymbol{\theta}_{1,t} &= \boldsymbol{\theta}_{t-1} + \frac{\lambda}{k} h g(\boldsymbol{\phi}_{m,t}, \boldsymbol{\theta}_{t-1}) \\
\boldsymbol{\theta}_{2,t} &= \boldsymbol{\theta}_{1,t-1} + \frac{\lambda}{k} h g(\boldsymbol{\phi}_{m,t}, \boldsymbol{\theta}_{1,t}) \\
&\dots \\
\boldsymbol{\theta}_{k,t} &= \boldsymbol{\theta}_{k-1,t-1} + \frac{\lambda}{k} h g(\boldsymbol{\phi}_{m,t}, \boldsymbol{\theta}_{k-1,t}) \\
\boldsymbol{\phi}_t &= \boldsymbol{\phi}_{m,t} \\
\boldsymbol{\theta}_t &= \boldsymbol{\theta}_{k,t}
\end{aligned}$$

Lemma A.2 *For update with $\boldsymbol{\phi}_{m,t} = \boldsymbol{\phi}_{m-1,t} + h f(\boldsymbol{\phi}_{m-1,t}, \boldsymbol{\theta}_{t-1})$ with step size h, the m-step update has the form:*

$$\boldsymbol{\phi}_{m,t} = \boldsymbol{\phi}_{t-1} + m h f_{(t-1)} + \frac{m(m-1)}{2} h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^3)$$

Proof: *Proof by induction. Base step.*

$$\begin{aligned}
\boldsymbol{\phi}_{2,t} &= \boldsymbol{\phi}_{1,t} + hf(\boldsymbol{\phi}_{1,t}, \boldsymbol{\theta}_{t-1}) \\
&= \boldsymbol{\phi}_{t-1} + hf(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) + hf(\boldsymbol{\phi}_{1,t}, \boldsymbol{\theta}_{t-1}) \\
&= \boldsymbol{\phi}_{t-1} + hf_{(t-1)} + hf\big(\boldsymbol{\phi}_{t-1} + hf_{(t-1)}, \boldsymbol{\theta}_{t-1}\big) \\
&= \boldsymbol{\phi}_{t-1} + hf_{(t-1)} + h\left(f_{(t-1)} + hf_{(t-1)}\nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^2)\right) \\
&= \boldsymbol{\phi}_{t-1} + 2hf_{(t-1)} + h^2 f_{(t-1)}\nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^3)
\end{aligned}$$

Inductive step:

$$\begin{aligned}
\boldsymbol{\phi}_{m+1,t} &= \boldsymbol{\phi}_{m,t} + hf(\boldsymbol{\phi}_{m,t}, \boldsymbol{\theta}_{t-1}) \\
&= \boldsymbol{\phi}_{t-1} + mhf_{(t-1)} + \frac{m(m-1)}{2}h^2 f_{(t-1)}\nabla_{\boldsymbol{\phi}} f_{(t-1)} \\
&\quad + hf\Big(\boldsymbol{\phi}_{t-1} + mhf_{(t-1)} + \frac{m(m-1)}{2}h^2 f_{(t-1)}\nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^3), \boldsymbol{\theta}_{t-1}\Big) + \mathcal{O}(h^3) \\
&= \boldsymbol{\phi}_{t-1} + mhf_{(t-1)} + \frac{m(m-1)}{2}h^2 f_{(t-1)}\nabla_{\boldsymbol{\phi}} f_{(t-1)} \\
&\quad + h\left(f_{(t-1)} + (mhf_{(t-1)} + \frac{m(m-1)}{2}h^2 f_{(t-1)}\nabla_{\boldsymbol{\phi}} f_{(t-1)})\nabla_{\boldsymbol{\phi}} f_{(t-1)}\right) + \mathcal{O}(h^3) \\
&= \boldsymbol{\phi}_{t-1} + (m+1)hf_{(t-1)} + \frac{m(m-1)}{2}h^2 f_{(t-1)}\nabla_{\boldsymbol{\phi}} f_{(t-1)} \\
&\quad + h\left(mhf_{(t-1)} + \frac{m(m-1)}{2}h^2 f_{(t-1)}\nabla_{\boldsymbol{\phi}} f_{(t-1)})\right)\nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^3) \\
&= \boldsymbol{\phi}_{t-1} + (m+1)hf_{(t-1)} + \frac{m(m+1)}{2}h^2 f_{(t-1)}\nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^3)
\end{aligned}$$

□

From Lemma A.2 with $h = \alpha h/m$ we have that:

$$\boldsymbol{\phi}_{m,t} = \phi_{t-1} + \alpha hf_{(t-1)} + \frac{m-1}{2m}\alpha^2 h^2 f_{(t-1)}\nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^3)$$

We now turn our attention to the second player. We define $g'_t = g(\boldsymbol{\phi}_{m,t}, \boldsymbol{\theta}_{t-1})$. This is where we get the difference between simultaneous and alternating updates comes in. From Lemma A.2 with $h = \lambda h/k$ we have that:

$$\boldsymbol{\theta}_{k,t} = \boldsymbol{\theta}_{t-1} + \lambda hg'_t + \frac{(k-1)}{2k}\lambda^2 h^2 g'_t \nabla_{\boldsymbol{\theta}} g'_t + \mathcal{O}(h^3)$$

We now expand g'_t by Taylor expansion:

$$\begin{aligned}
g'_t &= g(\boldsymbol{\phi}_{m,t}, \boldsymbol{\theta}_{t-1}) \\
&= g(\boldsymbol{\phi}_{t-1} + \alpha hf_{(t-1)} + \alpha^2\frac{m-1}{2m}h^2 f_{(t-1)}\nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^3), \boldsymbol{\theta}_{t-1}) \\
&= g_{(t-1)} + \left(\alpha hf_{(t-1)} + \frac{(m-1)}{2m}\alpha^2 h^2 f_{(t-1)}\nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^3)\right)\nabla_{\boldsymbol{\phi}} g_{(t-1)}
\end{aligned}$$

Thus, if we expand the RHS:

$$\begin{aligned}
\boldsymbol{\theta}_{k,t} &= \boldsymbol{\theta}_{t-1} + \lambda h g'_{t-1} + \frac{k-1}{2k}\lambda^2 h^2 g'_{t-1} \nabla_{\boldsymbol{\theta}} g'_{t-1} + \mathcal{O}(h^3) \\
&= \boldsymbol{\theta}_{t-1} + \lambda h \left(g_{(t-1)} + \left(\alpha h f_t + \frac{m-1}{2m}\alpha^2 h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^3) \right) \nabla_{\boldsymbol{\phi}} g_{(t-1)} \right) + \frac{k-1}{2k}\lambda^2 h^2 g'_{t-1} \nabla_{\boldsymbol{\theta}} g'_{t-1} + \mathcal{O}(h^3) \\
&= \boldsymbol{\theta}_{t-1} + \lambda h g_{(t-1)} + \lambda h \left(\alpha h f_{(t-1)} + \frac{m-1}{2m}\alpha^2 h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} \right) \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \frac{k-1}{2k}\lambda^2 h^2 g'_{t-1} \nabla_{\boldsymbol{\theta}} g'_{t-1} + \mathcal{O}(h^3) \\
&= \boldsymbol{\theta}_{t-1} + \lambda h g_{(t-1)} + \lambda \alpha h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \frac{k-1}{2k}\lambda^2 h^2 g'_{t-1} \nabla_{\boldsymbol{\theta}} g'_{t-1} + \mathcal{O}(h^3) \\
&= \boldsymbol{\theta}_{t-1} + \lambda h g_{(t-1)} + \lambda \alpha h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} \\
&\quad + \frac{k-1}{2k} h^2 \left(g_t + \left(\alpha h f_{(t-1)} + \frac{(m-1)}{2m}\alpha^2 h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^3) \right) \nabla_{\boldsymbol{\phi}} g_{(t-1)} \right) \nabla_{\boldsymbol{\theta}} g'_{t-1} + \mathcal{O}(h^3) \\
&= \boldsymbol{\theta}_{t-1} + \lambda h g_{(t-1)} + \lambda \alpha h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \frac{k-1}{2k}\lambda^2 h^2 g_{(t-1)} \nabla_{\boldsymbol{\theta}} g'_{t-1} + \mathcal{O}(h^3) \\
&= \boldsymbol{\theta}_{t-1} + \lambda h g_{(t-1)} + \lambda \alpha h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} \\
&\quad + \frac{k-1}{2k}\lambda^2 h^2 g_{(t-1)} \nabla_{\boldsymbol{\theta}} \left(g_{(t-1)} + \left(\alpha h f_{(t-1)} + \frac{(m-1)}{2m}\alpha^2 h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^3) \right) \nabla_{\boldsymbol{\phi}} g_{(t-1)} \right) + \mathcal{O}(h^3) \\
&= \boldsymbol{\theta}_{t-1} + \lambda h g_{(t-1)} + \lambda \alpha h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \frac{k-1}{2k}\lambda^2 h^2 g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} + \mathcal{O}(h^3)
\end{aligned}$$

We then have:

$$\boldsymbol{\phi}_{m,t} = \boldsymbol{\phi}_{t-1} + \alpha h f_{(t-1)} + \frac{m-1}{2m}\alpha^2 h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^3) \tag{A.15}$$

$$\boldsymbol{\theta}_{k,t} = \boldsymbol{\theta}_{t-1} + \lambda h g_{(t-1)} + \lambda \alpha h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \frac{k-1}{2k}\lambda^2 h^2 g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} + \mathcal{O}(h^3) \tag{A.16}$$

Step 2: Expand the continuous time changes for the modified ODE given by backward error analysis
(Identical to the simultaneous update case.)

Step 3: Matching the discrete and modified continuous updates.
The linear terms are identical to those in the simultaneous updates, which we reproduce here:

$$\begin{aligned}
\tau_\phi &= \alpha h \\
\tau_\theta &= \lambda h
\end{aligned}$$

We can then obtain f_1 from matching the quadratic terms in Equations (A.9) and Equations (A.15) — below we denote $f_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1})$ by f_1 and $g_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1})$ by g_1, for brevity:

$$\begin{aligned}
\alpha^2 h^2 f_1 + \frac{1}{2}\alpha^2 h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \frac{1}{2}\alpha^2 h^2 g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)} &= \frac{m-1}{2m}\alpha^2 h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} \\
f_1 + \frac{1}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \frac{1}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)} &= \frac{m-1}{2m} f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} \\
f_1 &= \left(\frac{m-1}{2m} - \frac{1}{2} \right) f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} - \frac{1}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)} \\
f_1 &= -\frac{1}{2m} f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} - \frac{1}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)}
\end{aligned}$$

For g_1, from Equations (A.10) and (A.16):

$$\begin{aligned}
\lambda^2 h^2 g_1 + \lambda^2 h^2 \frac{1}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \lambda^2 h^2 \frac{1}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} &= \lambda \alpha h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \frac{(k-1)}{2k} \lambda^2 h^2 g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} \\
g_1 + \frac{1}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \frac{1}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} &= \frac{\alpha}{\lambda} f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \frac{(k-1)}{2k} g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} \\
g_1 + \frac{1}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \frac{1}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} &= \frac{\alpha}{\lambda} f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \frac{(k-1)}{2k} g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} \\
g_1 &= \left(\frac{\alpha}{\lambda} - \frac{1}{2} \right) f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} - \frac{1}{2k} g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)}
\end{aligned}$$

We thus have that:

$$f_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) = -\frac{1}{2m} f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} - \frac{1}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)} \tag{A.17}$$

$$g_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) = \left(\frac{\alpha}{\lambda} - \frac{1}{2} \right) f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} - \frac{1}{2k} g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} \tag{A.18}$$

We have found the functions f_1 and g_1 such that after one discrete optimization step the ODEs $\dot{\boldsymbol{\phi}} = f + \alpha h f_1$ and $\dot{\boldsymbol{\theta}} = g + \lambda h g_1$ follow the discrete updates up to order $\mathcal{O}(h^3)$, finishing the proof.

B. Proof of the main corollaries

In this section, we will write the modified equations in the case of using gradient descent common-payoff games and zero-sum games. This amounts to specialize the following first order corrections we have derived in the previous sections.

To do so, we will replace f and g for the values given by gradient descent, eg. in the common pay-off case $f = -\nabla_{\boldsymbol{\phi}} E$ and $g = -\nabla_{\boldsymbol{\theta}} E$ and $f = \nabla_{\boldsymbol{\phi}} E$ and $g = -\nabla_{\boldsymbol{\theta}} E$ where $E(\boldsymbol{\phi}, \boldsymbol{\theta})$ is a function of the player parameters. We will use the following identities:

$$\begin{aligned}
\nabla_{\boldsymbol{\phi}} E \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\phi}} E &= \nabla_{\boldsymbol{\phi}} \Big(\tfrac{\left\| \nabla_{\boldsymbol{\phi}} E \right\|^2}{2} \Big), \qquad & \nabla_{\boldsymbol{\theta}} E \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\phi}} E &= \nabla_{\boldsymbol{\phi}} \Big(\frac{\left\| \nabla_{\boldsymbol{\theta}} E \right\|^2}{2} \Big) \\
\nabla_{\boldsymbol{\phi}} E \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}} E &= \nabla_{\boldsymbol{\theta}} \Big(\tfrac{\left\| \nabla_{\boldsymbol{\phi}} E \right\|^2}{2} \Big), \qquad & \nabla_{\boldsymbol{\theta}} E \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} E &= \nabla_{\boldsymbol{\theta}} \Big(\frac{\left\| \nabla_{\boldsymbol{\theta}} E \right\|^2}{2} \Big)
\end{aligned}$$

B.1. Common-payoff alternating two player-games (Corollary 5.1)

Corollary 5.1 *In a two-player common-payoff game with common loss E, alternating gradient descent – as described in Equations* (A.13) *and* (A.14) *- with one update per player follows a gradient flow given by the modified losses*

$$\tilde{L}_1 = E + \frac{\alpha h}{4} \left(\left\| \nabla_{\boldsymbol{\phi}} E \right\|^2 + \left\| \nabla_{\boldsymbol{\theta}} E \right\|^2 \right) \tag{A.19}$$

$$\tilde{L}_2 = E + \frac{\lambda h}{4} \left((1 - \frac{2\alpha}{\lambda}) \left\| \nabla_{\boldsymbol{\phi}} E \right\|^2 + \left\| \nabla_{\boldsymbol{\theta}} E \right\|^2 \right) \tag{A.20}$$

with an error of size $\mathcal{O}(h^3)$ after one update step.

In the common-payoff case, both players minimize the same loss function E. Substituting $f = -\nabla_{\boldsymbol{\phi}} E$ and $g = -\nabla_{\boldsymbol{\theta}} E$

into the corrections f_1 and g_1 for the alternating Euler updates in Theorem 3.2 and using the gradient identities above yields

$$
\begin{aligned}
f_1 &= -\frac{1}{2}\left(f\nabla_{\boldsymbol{\phi}} f + g\nabla_{\boldsymbol{\theta}} f\right) \\
&= -\frac{1}{2}\left(\frac{1}{m}\nabla_{\boldsymbol{\phi}} E\nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\phi}} E + \nabla_{\boldsymbol{\theta}} E\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\phi}} E\right), \\
&= -\frac{1}{2}\left(\frac{1}{m}\nabla_{\boldsymbol{\phi}}\frac{\|\nabla_{\boldsymbol{\phi}} E\|^2}{2} + \nabla_{\boldsymbol{\phi}}\frac{\|\nabla_{\boldsymbol{\theta}} E\|^2}{2}\right), \\
&= -\nabla_{\boldsymbol{\phi}}\left(\frac{1}{4m}\|\nabla_{\boldsymbol{\phi}} E\|^2 + \frac{1}{4}\|\nabla_{\boldsymbol{\theta}} E\|^2\right)
\end{aligned}
$$

$$
\begin{aligned}
g_1 &= -\frac{1}{2}\left((1-\frac{2\alpha}{\lambda})f\nabla_{\boldsymbol{\phi}} g + \frac{1}{k}g\nabla_{\boldsymbol{\theta}} g\right) \\
&= -\frac{1}{2}\left((1-\frac{2\alpha}{\lambda})\nabla_{\boldsymbol{\phi}} E\nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\theta}} E + \frac{1}{k}\nabla_{\boldsymbol{\theta}} E\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}} E\right), \\
&= -\frac{1}{2}\left((1-\frac{2\alpha}{\lambda})\nabla_{\boldsymbol{\theta}}\frac{\|\nabla_{\boldsymbol{\phi}} E\|^2}{2} + \frac{1}{k}\nabla_{\boldsymbol{\theta}}\frac{\|\nabla_{\boldsymbol{\theta}} E\|^2}{2}\right), \\
&= -\nabla_{\boldsymbol{\theta}}\left(\frac{1}{4}(1-\frac{2\alpha}{\lambda})\|\nabla_{\boldsymbol{\phi}} E\|^2 + \frac{1}{k}\|\nabla_{\boldsymbol{\theta}} E\|^2\right)
\end{aligned}
$$

Now, replacing the gradient expressions for f_1 and g_1 calculated above into the modified equations $\dot{\boldsymbol{\phi}} = -\nabla_{\boldsymbol{\phi}} E + \alpha h f_1$ and $\dot{\boldsymbol{\theta}} = -\nabla_{\boldsymbol{\theta}} E + \lambda h g_1$ and factoring out the gradients, we obtain the modified equations in the form of the ODEs:

$$\dot{\boldsymbol{\phi}} = -\nabla_{\boldsymbol{\phi}}\widetilde{L}_1, \tag{A.21}$$

$$\dot{\boldsymbol{\theta}} = -\nabla_{\boldsymbol{\theta}}\widetilde{L}_2, \tag{A.22}$$

with the following modified losses for each players:

$$\widetilde{L}_1 = E + \frac{\alpha h}{4}\left(\frac{1}{m}\|\nabla_{\boldsymbol{\phi}} E\|^2 + \|\nabla_{\boldsymbol{\theta}} E\|^2\right), \tag{A.23}$$

$$\widetilde{L}_2 = E + \frac{\lambda h}{4}\left((1-\frac{2\alpha}{\lambda})\|\nabla_{\boldsymbol{\phi}} E\|^2 + \frac{1}{k}\|\nabla_{\boldsymbol{\theta}} E\|^2\right). \tag{A.24}$$

We obtain Corollary 5.1 by setting the number of player updates to one: $m = k = 1$.

B.2. Zero-sum simultaneous two player-games (Corollary 6.1)

Corollary 6.1 *In a zero-sum two-player differentiable game, simultaneous gradient descent updates - as described in Equations* (A.5) *and* (A.6) *- follows a gradient flow given by the modified losses*

$$\tilde{L}_1 = -E + \frac{\alpha h}{4}\|\nabla_{\boldsymbol{\phi}} E\|^2 - \frac{\alpha h}{4}\|\nabla_{\boldsymbol{\theta}} E\|^2, \tag{A.25}$$

$$\tilde{L}_2 = E - \frac{\lambda h}{4}\|\nabla_{\boldsymbol{\phi}} E\|^2 + \frac{\lambda h}{4}\|\nabla_{\boldsymbol{\theta}} E\|^2, \tag{A.26}$$

with an error of size $\mathcal{O}(h^3)$ after one update step.

In this case, substituting $f = \nabla_{\boldsymbol{\phi}} E$ and $g = -\nabla_{\boldsymbol{\theta}} E$ into the corrections f_1 and g_1 for the simultaneous Euler updates and using the gradient identities above yields

$$
\begin{aligned}
f_1 &= -\frac{1}{2}\left(f\nabla_{\boldsymbol{\phi}} f + g\nabla_{\boldsymbol{\theta}} f\right) \\
&= -\frac{1}{2}\left(\nabla_{\boldsymbol{\phi}} E\nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\phi}} E - \nabla_{\boldsymbol{\theta}} E\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\phi}} E\right), \\
&= -\frac{1}{2}\left(\nabla_{\phi}\frac{\|\nabla_{\boldsymbol{\phi}} E\|^2}{2} - \nabla_{\phi}\frac{\|\nabla_{\boldsymbol{\theta}} E\|^2}{2}\right), \\
&= -\nabla_{\phi}\left(\frac{1}{4}\|\nabla_{\boldsymbol{\phi}} E\|^2 - \frac{1}{4}\|\nabla_{\boldsymbol{\theta}} E\|^2\right)
\end{aligned}
$$

$$
\begin{aligned}
g_1 &= -\frac{1}{2}\left(f\nabla_{\boldsymbol{\phi}} g + g\nabla_{\boldsymbol{\theta}} g\right) \\
&= -\frac{1}{2}\left(-\nabla_{\boldsymbol{\phi}} E\nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\theta}} E + \nabla_{\boldsymbol{\theta}} E\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}} E\right) \\
&= -\frac{1}{2}\left(-\nabla_{\boldsymbol{\theta}}\frac{\|\nabla_{\boldsymbol{\phi}} E\|^2}{2} + \nabla_{\boldsymbol{\theta}}\frac{\|\nabla_{\boldsymbol{\theta}} E\|^2}{2}\right), \\
&= -\nabla_{\boldsymbol{\theta}}\left(-\frac{1}{4}\|\nabla_{\boldsymbol{\phi}} E\|^2 + \frac{1}{4}\|\nabla_{\boldsymbol{\theta}} E\|^2\right)
\end{aligned}
$$

Now, replacing the gradient expressions for f_1 and g_1 calculated above into the modified equations $\dot{\boldsymbol{\phi}} = -\nabla_{\boldsymbol{\phi}}(-E) + \alpha h f_1$ and $\dot{\boldsymbol{\theta}} = -\nabla_{\boldsymbol{\theta}} E + \lambda h g_1$ and factoring out the gradients, we obtain the modified equations in the form of the ODEs:

$$
\begin{aligned}
\dot{\boldsymbol{\phi}} &= -\nabla_{\boldsymbol{\phi}}\widetilde{L}_1, &\text{(A.27)}\\
\dot{\boldsymbol{\theta}} &= -\nabla_{\boldsymbol{\theta}}\widetilde{L}_2, &\text{(A.28)}
\end{aligned}
$$

with the following modified losses for each players:

$$
\begin{aligned}
\widetilde{L}_1 &= -E + \frac{\alpha h}{4}\|\nabla_{\boldsymbol{\phi}} E\|^2 - \frac{\alpha h}{4}\|\nabla_{\boldsymbol{\theta}} E\|^2, &\text{(A.29)}\\
\widetilde{L}_2 &= E - \frac{\lambda h}{4}\|\nabla_{\boldsymbol{\phi}} E\|^2 + \frac{\lambda h}{4}\|\nabla_{\boldsymbol{\theta}} E\|^2. &\text{(A.30)}
\end{aligned}
$$

B.3. Zero-sum alternating two-player games (Corollary 6.2)

Corollary 6.2 *In a zero-sum two-player differentiable game, alternating gradient descent - as described in Equations* (A.13) *and* (A.14) *- follows a gradient flow given by the modified losses*

$$
\begin{aligned}
\tilde{L}_1 &= -E + \frac{\alpha h}{4m}\left\|\nabla_{\boldsymbol{\phi}} E\right\|^2 - \frac{\alpha h}{4}\left\|\nabla_{\boldsymbol{\theta}} E\right\|^2 &\text{(A.31)}\\
\tilde{L}_2 &= E - \frac{\lambda h}{4}\left(1 - \frac{2\alpha}{\lambda}\right)\left\|\nabla_{\boldsymbol{\phi}} E\right\|^2 + \frac{\lambda h}{4k}\left\|\nabla_{\boldsymbol{\theta}} E\right\|^2 &\text{(A.32)}
\end{aligned}
$$

with an error of size $\mathcal{O}(h^3)$ after one update step.

In this last case, substituting $f = \nabla_{\boldsymbol{\phi}} E$ and $g = -\nabla_{\boldsymbol{\theta}} E$ into the corrections f_1 and g_1 for the alternating Euler updates and using the gradient identities above yields the modified system as well as the modified losses exactly in the same way as for the two previous cases above. (This amounts to a single sign change in the proof of Corollary 5.1)

B.4. Self and interaction terms in zero-sum games

Remark B.1 *Throughout the Supplementary Material, we will refer to self terms and interaction terms, as originally defined in our paper (Definition 3.1), and we will also use this terminology to refer to terms in our derivations that originate from the self terms and interaction terms. While a slight abuse of language, we find it useful to emphasize the provenance of these terms in our discussion.*

For the case of zero-sum games trained with simultaneous gradient descent, the self terms encourage the minimization of the player's own gradient norm, while the interaction terms encourage the maximization of the other player's gradient norm:

$$\widetilde{L}_1 = -E + \underbrace{\frac{\alpha h}{4}\|\nabla_{\boldsymbol{\phi}} E\|^2}_{self} \underbrace{- \frac{\alpha h}{4}\|\nabla_{\boldsymbol{\theta}} E\|^2}_{interaction},$$

$$\widetilde{L}_2 = E \underbrace{- \frac{\lambda h}{4}\|\nabla_{\boldsymbol{\phi}} E\|^2}_{interaction} \underbrace{+ \frac{\lambda h}{4}\|\nabla_{\boldsymbol{\theta}} E\|^2}_{self}.$$

Similar terms are obtained for alternating gradient descent, with the only difference that the sign of the *interaction* term for the second player can change and become positive.

C. General differentiable two-player games

Consider now the case where we have two loss functions for the two players respectively $L_1(\boldsymbol{\phi}, \boldsymbol{\theta}) : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ and $L_2(\boldsymbol{\phi}, \boldsymbol{\theta}) : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$. This leads to the update functions $f = -\nabla_{\boldsymbol{\phi}} L_1$ and $g = -\nabla_{\boldsymbol{\theta}} L_2$.

We show below that in the most general case *we cannot write the modified updates as gradient*. That is, in the general case we cannot write $f + hf_1$ as $\nabla_{\boldsymbol{\phi}} \widetilde{L}_1$, since f_1 will not be the gradient of a function, and similarly for g_1. Consequently, if we want to study the effect of DD on general games, we have to work at the level of the *modified vector fields* $f + hf_1$ and $g + hg_1$ defining the modified ODEs directly — as we have done for the stability analysis results — rather than working with losses.

By using $f = -\nabla_{\boldsymbol{\phi}} L_1$ and $g = -\nabla_{\boldsymbol{\theta}} L_2$, we can rewrite the drift of the simultaneous Euler updates (corresponding to simultaneous gradient descent in this setting) as:

$$\begin{aligned} f_1 &= -\frac{1}{2} f \nabla_{\boldsymbol{\phi}} f - \frac{1}{2} g \nabla_{\boldsymbol{\theta}} f && \text{(A.33)} \\ &= -\frac{1}{2} \nabla_{\boldsymbol{\phi}} L_1 \nabla_{\boldsymbol{\phi}} (\nabla_{\boldsymbol{\phi}} L_1) - \frac{1}{2} (\nabla_{\boldsymbol{\theta}} L_2) \nabla_{\boldsymbol{\theta}} (\nabla_{\boldsymbol{\phi}} L_1) && \text{(A.34)} \\ &= -\frac{1}{4} \nabla_{\boldsymbol{\phi}} \left\| \nabla_{\boldsymbol{\phi}} L_1 \right\|^2 - \frac{1}{2} (\nabla_{\boldsymbol{\theta}} L_2) \nabla_{\boldsymbol{\theta}} (\nabla_{\boldsymbol{\phi}} L_1) && \text{(A.35)} \end{aligned}$$

and similarly

$$g_1 = -\frac{1}{4} \nabla_{\boldsymbol{\theta}} \left\| \nabla_{\boldsymbol{\theta}} L_2 \right\|^2 - \frac{1}{2} (\nabla_{\boldsymbol{\phi}} L_1) \nabla_{\boldsymbol{\phi}} (\nabla_{\boldsymbol{\theta}} L_2) \qquad \text{(A.36)}$$

As we can see here, it is not possible in general to write f_1 and g_1 as gradient functions, as was possible for a zero-sum game or a common-payoff game.

D. Discretization drift in Runge-Kutta 4 (RK4)

Runge-Kutta 4 for one player
RK4 is a Runge-Kutta a method of order 4. This means that the discrete steps of RK4 coincide with the exact flow of the original ODE up to $\mathcal{O}(h^5)$ (i.e., the local error after one step is of order $\mathcal{O}(h^5)$). The modified equation for a method of order n starts with corrections at order h^{n+1} (i.e., all the lower corrections vanish; see (Hairer & Lubich, 1997) and (Hairer et al., 2006) for further details). This means that RK4 has no DD up to order $\mathcal{O}(h^5)$, and why for small learning rates RK4 can be used as a proxy for the exact flow.

Runge-Kutta 4 for two players
When we use equal learning rates and simultaneous updates, the two-player game is always equivalent to the one player case, so Runge-Kutta 4 will have a local error of $\mathcal{O}(h^5)$. However, in the case of two-players games, we have the additional freedom of having different learning rates for each of the players. We now show that when the learning rates of the two

players are different, *RK4 also also has a drift effect of order* 2 *and the DD term comes exclusively from the interaction terms*. To do so, we apply the same steps as we have done for the Euler updates.

Step 1: Expand the updates per player via a Taylor expansion.
The simultaneous Runge-Kutta 4 updates for two players are:

$$\begin{aligned}
k_{1,\boldsymbol{\phi}} &= f(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) \\
k_{1,\boldsymbol{\theta}} &= g(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) \\
k_{2,\boldsymbol{\phi}} &= f(\boldsymbol{\phi}_{t-1} + \frac{\alpha h}{2} k_{1,\boldsymbol{\phi}}, \boldsymbol{\theta}_{t-1} + \frac{\lambda h}{2} k_{1,\boldsymbol{\theta}}) \\
k_{2,\boldsymbol{\theta}} &= g(\boldsymbol{\phi}_{t-1} + \frac{\alpha h}{2} k_{1,\boldsymbol{\phi}}, \boldsymbol{\theta}_{t-1} + \frac{\lambda h}{2} k_{1,\boldsymbol{\theta}}) \\
k_{3,\boldsymbol{\phi}} &= f(\boldsymbol{\phi}_{t-1} + \frac{\alpha h}{2} k_{2,\boldsymbol{\phi}}, \boldsymbol{\theta}_{t-1} + \frac{\lambda h}{2} k_{2,\boldsymbol{\theta}}) \\
k_{3,\boldsymbol{\theta}} &= g(\boldsymbol{\phi}_{t-1} + \frac{\alpha h}{2} k_{2,\boldsymbol{\phi}}, \boldsymbol{\theta}_{t-1} + \frac{\lambda h}{2} k_{2,\boldsymbol{\theta}}) \\
k_{4,\boldsymbol{\phi}} &= f(\boldsymbol{\phi}_{t-1} + \frac{\alpha h}{2} k_{3,\boldsymbol{\phi}}, \boldsymbol{\theta}_{t-1} + \frac{\lambda h}{2} k_{3,\boldsymbol{\theta}}) \\
k_{4,\boldsymbol{\theta}} &= g(\boldsymbol{\phi}_{t-1} + \frac{\alpha h}{2} k_{3,\boldsymbol{\phi}}, \boldsymbol{\theta}_{t-1} + \frac{\lambda h}{2} k_{3,\boldsymbol{\theta}})
\end{aligned}$$

$$\begin{aligned}
k_{\boldsymbol{\phi}} &= \frac{1}{6}\left(k_{1,\boldsymbol{\phi}} + 2k_{2,\boldsymbol{\phi}} + 2k_{3,\boldsymbol{\phi}} + k_{4,\boldsymbol{\phi}}\right) \\
k_{\boldsymbol{\theta}} &= \frac{1}{6}\left(k_{1,\boldsymbol{\theta}} + 2k_{2,\boldsymbol{\theta}} + 2k_{3,\boldsymbol{\theta}} + k_{4,\boldsymbol{\theta}}\right)
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\phi}_t &= \boldsymbol{\phi}_{t-1} + \alpha h k_{\boldsymbol{\phi}} \\
\boldsymbol{\theta}_t &= \boldsymbol{\theta}_{t-1} + \lambda h k_{\boldsymbol{\theta}}
\end{aligned}$$

We expand each intermediate step:

$$\begin{aligned}
k_{1,\boldsymbol{\phi}} &= f_{(t-1)} \\
k_{1,\boldsymbol{\theta}} &= g_{(t-1)} \\
k_{2,\boldsymbol{\phi}} &= f(\boldsymbol{\phi}_{t-1} + \frac{\alpha h}{2} k_{1,\boldsymbol{\phi}}, \boldsymbol{\theta}_{t-1} + \frac{\lambda h}{2} k_{1,\boldsymbol{\theta}}) = f(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1} + \frac{\lambda h}{2} k_{1,\boldsymbol{\theta}}) + \frac{\alpha h}{2} k_{1,\boldsymbol{\phi}} \nabla_{\boldsymbol{\phi}} f(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1} + \frac{\lambda h}{2} k_{1,\boldsymbol{\theta}}) + \mathcal{O}(h^2) \\
&= f_{(t-1)} + \frac{\lambda h}{2} k_{1,\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} f_{(t-1)} + \frac{\alpha h}{2} k_{1,\boldsymbol{\phi}} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^2) \\
&= f_{(t-1)} + \frac{\lambda h}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)} + \frac{\alpha h}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^2) \\
k_{2,\boldsymbol{\theta}} &= g(\boldsymbol{\phi}_{t-1} + \frac{\alpha h}{2} k_{1,\boldsymbol{\phi}}, \boldsymbol{\theta} + \frac{\lambda h}{2} k_{1,\boldsymbol{\theta}}) \\
&= g_{(t-1)} + \frac{\lambda h}{2} k_{1,\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} g_{(t-1)} + \frac{\alpha h}{2} k_{1,\boldsymbol{\phi}} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \mathcal{O}(h^2) \\
&= g_{(t-1)} + \frac{\lambda h}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} + \frac{\alpha h}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \mathcal{O}(h^2)
\end{aligned}$$

$$
\begin{aligned}
k_{3,\boldsymbol{\phi}} &= f_{(t-1)} + \frac{\lambda h}{2} k_{2,\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} f_{(t-1)} + \frac{\alpha h}{2} k_{2,\boldsymbol{\phi}} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^2) \\
&= f_{(t-1)} + \frac{\lambda h}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)} + \frac{\alpha h}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^2) \\
k_{3,\boldsymbol{\theta}} &= g_{(t-1)} + \frac{\lambda h}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} + \frac{\alpha h}{2} f(\boldsymbol{\phi}, \boldsymbol{\theta}) \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \mathcal{O}(h^2) \\
k_{4,\boldsymbol{\phi}} &= f_{(t-1)} + \frac{\lambda h}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)} + \frac{\alpha h}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^2) \\
k_{4,\boldsymbol{\theta}} &= g_{(t-1)} + \frac{\lambda h}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} + \frac{\alpha h}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \mathcal{O}(h^2)
\end{aligned}
$$

with the update direction:

$$
\begin{aligned}
k_{\boldsymbol{\phi}} &= \frac{1}{6}\left(k_{1,\boldsymbol{\phi}} + 2k_{2,\boldsymbol{\phi}} + 2k_{3,\boldsymbol{\phi}} + k_{4,\boldsymbol{\phi}}\right) = f_{(t-1)} + \frac{\lambda h}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)} + \frac{\alpha h}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^2) \\
k_{\boldsymbol{\theta}} &= \frac{1}{6}\left(k_{1,\boldsymbol{\theta}} + 2k_{2,\boldsymbol{\theta}} + 2k_{3,\boldsymbol{\theta}} + k_{4,\boldsymbol{\theta}}\right) = g_{(t-1)} + \frac{\lambda h}{2} g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} + \frac{\alpha h}{2} f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \mathcal{O}(h^2)
\end{aligned}
$$

and thus the discrete update of the Runge-Kutta 4 for two players are:

$$
\boldsymbol{\phi}_t = \boldsymbol{\phi}_{t-1} + \alpha h f_{(t-1)} + \frac{1}{2}\alpha\lambda h^2 g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)} + \frac{1}{2}\alpha^2 h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \mathcal{O}(h^3) \tag{A.37}
$$

$$
\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \lambda h g_{(t-1)} + \frac{1}{2}\lambda^2 h^2 g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} + \frac{1}{2}\alpha\lambda h^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \mathcal{O}(h^3) \tag{A.38}
$$

Step 2: Expand the continuous time changes for the modified ODE
(Identical to the simultaneous Euler updates.)

Step 3: Matching the discrete and modified continuous updates.
As in the always in Step 3, we substitute $\boldsymbol{\phi}_{t-1}$ and $\boldsymbol{\theta}_{t-1}$ in Lemma A.1:

$$
\boldsymbol{\phi}(\tau + \tau_\phi) = \boldsymbol{\phi}_{t-1} + \tau_\phi f_{(t-1)} + \tau_\phi^2 f_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) + \frac{1}{2}\tau_\phi^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} f_{(t-1)} + \frac{1}{2}\tau_\phi^2 g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{(t-1)} + \mathcal{O}(\tau_\phi^3) \tag{A.39}
$$

$$
\boldsymbol{\theta}(\tau + \tau_\theta) = \boldsymbol{\theta}_{t-1} + \tau_\theta g_{(t-1)} + \tau_\theta^2 g_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) + \frac{1}{2}\tau_\theta^2 f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{(t-1)} + \frac{1}{2}\tau_\theta^2 g_{(t-1)} \nabla_{\boldsymbol{\theta}} g_{(t-1)} + \mathcal{O}(\tau_\theta^3) \tag{A.40}
$$

For the first order terms we obtain $\tau_\phi = \alpha h$ and $\tau_\theta = \lambda h$. We match the $\mathcal{O}(h^2)$ terms in the equations above with the discrete Runge-Kutta 4 updates shown in Equation (A.37) and (A.38) and notice that:

$$
f_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) = \frac{1}{2}(\frac{\lambda}{\alpha} - 1) g_{(t-1)} \nabla_{\boldsymbol{\theta}} f_{t-1} \tag{A.41}
$$

$$
g_1(\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) = \frac{1}{2}(\frac{\alpha}{\lambda} - 1) f_{(t-1)} \nabla_{\boldsymbol{\phi}} g_{t-1} \tag{A.42}
$$

Thus, if $\alpha \neq \lambda$ RK4 has second order drift. This is why, in all our experiments comparing with RK4, we use the same learning rates for the two players $\alpha = \lambda$, to ensure that we use a method which has no DD up to order $\mathcal{O}(h^5)$.

E. Stability analysis

In this section, we give all the details of the stability analysis results, to showcase how the modified ODEs we have derived can be used as a tool for stability analysis. We provide the full computation for the Jacobian of the modified vector fields for simultaneous and alternating Euler updates, as well as the calculation of their trace, and show how this can be used to determine the stability of the modified vector fields. While analyzing the modified vector fields is not equivalent to analyzing the discrete dynamics due to the higher order errors of the drift which we ignore, it provides a better approximation than what is often used in practice, namely the original ODEs, which completely ignore the drift.

E.1. Simultaneous Euler updates

Consider a two-player game with dynamics given by $\boldsymbol{\phi}_t = \boldsymbol{\phi}_{t-1} + \alpha h f_{(t-1)}$ and $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \lambda h g_{(t-1)}$ (Equations (A.5) and (A.6)). The modified dynamics for this game are given by $\dot{\boldsymbol{\phi}} = \widetilde{f}$, $\dot{\boldsymbol{\theta}} = \widetilde{g}$, where $\widetilde{f} = f - \frac{\alpha h}{2}(f \nabla_{\boldsymbol{\phi}} f + g \nabla_{\boldsymbol{\theta}} f)$ and $\widetilde{g} = g - \frac{\lambda h}{2}(f \nabla_{\boldsymbol{\phi}} g + g \nabla_{\boldsymbol{\theta}} g)$ (Theorem 3.1).

The stability of this system can be characterized by the modified Jacobian matrix evaluated at the equilibria of the two-player game. The equilibria that we are interested in for our stability analysis are the steady-state solutions of Equations (A.5) and (A.6), given by $f = \mathbf{0}, g = \mathbf{0}$. These are also equilibrium solutions for the steady-state modified equations[1] given by $\tilde{f} = \mathbf{0}, \tilde{g} = \mathbf{0}$.

The modified Jacobian can be written, using block matrix notation as:

$$\widetilde{J} = \begin{bmatrix} \nabla_{\boldsymbol{\phi}} \widetilde{f} & \nabla_{\boldsymbol{\theta}} \widetilde{f} \\ \nabla_{\boldsymbol{\phi}} \widetilde{g} & \nabla_{\boldsymbol{\theta}} \widetilde{g} \end{bmatrix} \tag{A.43}$$

Next, we calculate each term in this block matrix. (In the following analysis, each term is evaluated at an equilibrium solution given by $f = \mathbf{0}, g = \mathbf{0}$). We find:

$$\begin{aligned}
\nabla_{\boldsymbol{\phi}} \tilde{f} &= \nabla_{\boldsymbol{\phi}} f - \frac{\alpha h}{2}\left((\nabla_{\boldsymbol{\phi}} f)^2 + f \nabla_{\boldsymbol{\phi},\boldsymbol{\phi}} f + \nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} f + g \nabla_{\boldsymbol{\phi},\boldsymbol{\theta}} f\right) \\
&= \nabla_{\boldsymbol{\phi}} f - \frac{\alpha h}{2}\left((\nabla_{\boldsymbol{\phi}} f)^2 + \nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} f\right) \\
\nabla_{\boldsymbol{\theta}} \tilde{f} &= \nabla_{\boldsymbol{\theta}} f - \frac{\alpha h}{2}\left(\nabla_{\boldsymbol{\theta}} f \nabla_{\boldsymbol{\phi}} f + f \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} f + \nabla_{\boldsymbol{\theta}} g \nabla_{\boldsymbol{\theta}} f + g \nabla_{\boldsymbol{\theta},\boldsymbol{\theta}} f\right) \\
&= \nabla_{\boldsymbol{\theta}} f - \frac{\alpha h}{2}\left(\nabla_{\boldsymbol{\theta}} f \nabla_{\boldsymbol{\phi}} f + \nabla_{\boldsymbol{\theta}} g \nabla_{\boldsymbol{\theta}} f\right) \\
\nabla_{\boldsymbol{\phi}} \tilde{g} &= \nabla_{\boldsymbol{\phi}} g - \frac{\lambda h}{2}(\nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} g + g \nabla_{\boldsymbol{\phi},\boldsymbol{\theta}} g + \nabla_{\boldsymbol{\phi}} f \nabla_{\boldsymbol{\phi}} g + f \nabla_{\boldsymbol{\phi},\boldsymbol{\phi}} g) \\
&= \nabla_{\boldsymbol{\phi}} g - \frac{\lambda h}{2}(\nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} g + \nabla_{\boldsymbol{\phi}} f \nabla_{\boldsymbol{\phi}} g) \\
\nabla_{\boldsymbol{\theta}} \tilde{g} &= \nabla_{\boldsymbol{\theta}} g - \frac{\lambda h}{2}\left((\nabla_{\boldsymbol{\theta}} g)^2 + g \nabla_{\boldsymbol{\theta},\boldsymbol{\theta}} g + \nabla_{\boldsymbol{\theta}} f \nabla_{\boldsymbol{\phi}} g + f \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} g\right) \\
&= \nabla_{\boldsymbol{\theta}} g - \frac{\lambda h}{2}\left((\nabla_{\boldsymbol{\theta}} g)^2 + \nabla_{\boldsymbol{\theta}} f \nabla_{\boldsymbol{\phi}} g\right)
\end{aligned}$$

Given these calculations, we can now write:

$$\widetilde{J} = \begin{bmatrix} \nabla_{\boldsymbol{\phi}} \widetilde{f} & \nabla_{\boldsymbol{\theta}} \widetilde{f} \\ \nabla_{\boldsymbol{\phi}} \widetilde{g} & \nabla_{\boldsymbol{\theta}} \widetilde{g} \end{bmatrix} = J - \frac{h}{2} K_{\text{sim}} \tag{A.44}$$

where J is the Jacobian of the unmodified ODE:

$$J = \begin{bmatrix} \nabla_{\boldsymbol{\phi}} f & \nabla_{\boldsymbol{\theta}} f \\ \nabla_{\boldsymbol{\phi}} g & \nabla_{\boldsymbol{\theta}} g \end{bmatrix} \tag{A.45}$$

and

$$K_{\text{sim}} = \begin{bmatrix} \alpha(\nabla_{\boldsymbol{\phi}} f)^2 + \alpha \nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} f & \alpha \nabla_{\boldsymbol{\theta}} f \nabla_{\boldsymbol{\phi}} f + \alpha \nabla_{\boldsymbol{\theta}} g \nabla_{\boldsymbol{\theta}} f \\ \lambda \nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} g + \lambda \nabla_{\boldsymbol{\phi}} f \nabla_{\boldsymbol{\phi}} g & \lambda (\nabla_{\boldsymbol{\theta}} g)^2 + \lambda \nabla_{\boldsymbol{\theta}} f \nabla_{\boldsymbol{\phi}} g \end{bmatrix} \tag{A.46}$$

The modified system of equations are asymptotically stable if all the real parts of all the eigenvalues of the modified Jacobian are negative. If some of the eigenvalues are zero, the equilibrium may or may not be stable, depending on the non-linearities of the system. If the real parts of any eigenvalues are positive, then the dynamical system is unstable.

[1] There are additional steady-state solutions for the modified equations. However, we can ignore these since they are spurious solutions that do not correspond to steady states of the two-player game, arising instead as an artifact of the $\mathcal{O}(h^3)$ error in backward error analysis.

A necessary condition for stability is that the trace of the modified Jacobian is less than or equal to zero (i.e. $\mathrm{Tr}(\widetilde{J}) \leq 0$), since the trace is the sum of eigenvalues. Using the property of trace additivity and the trace cyclic property we see that:

$$\mathrm{Tr}(\widetilde{J}) = \mathrm{Tr}(J) - \frac{h}{2}\left(\alpha\,\mathrm{Tr}((\nabla_{\boldsymbol{\phi}} f)^2) + \lambda\,\mathrm{Tr}((\nabla_{\boldsymbol{\theta}} g)^2)\right) - \frac{h}{2}(\alpha+\lambda)\,\mathrm{Tr}(\nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} f) \tag{A.47}$$

E.1.1. Instability caused by discretization drift

We now use the above analysis to show that the equilibria of a two-player game following the modified ODE obtained simultaneous Euler updates as defined by Equations (A.5) and (A.6) can become asymptotically unstable for some games.

There are choices of f and g that have stable equilibrium without DD, but are unstable under DD. For example, consider the zero-sum two-player game with $f = \nabla_{\boldsymbol{\phi}} E$ and $g = -\nabla_{\boldsymbol{\theta}} E$. Now, we see that

$$\mathrm{Tr}(\widetilde{J}) = \mathrm{Tr}(J) - \frac{h}{2}\left(\alpha \|\nabla_{\boldsymbol{\phi},\boldsymbol{\phi}} E\|_F^2 + \lambda \|\nabla_{\boldsymbol{\theta},\boldsymbol{\theta}} E\|_F^2\right) + \frac{h}{2}(\alpha+\lambda)\|\nabla_{\boldsymbol{\phi},\boldsymbol{\theta}} E\|_F^2 \tag{A.48}$$

where $\|.\|_F$ denotes the Frobenius norm. The Dirac-GAN is an example of a zero-sum two-player game that is stable without DD, but becomes unstable under DD with $\mathrm{Tr}(\widetilde{J}) = h(\alpha+\lambda)\|\nabla_{\boldsymbol{\phi},\boldsymbol{\theta}} E\|_F^2/2 > 0$ (see Section G.2).

E.2. Alternating Euler updates

Consider a two-player game with dynamics given by Equations (A.13) and (A.14). The modified dynamics for this game are given by $\dot{\boldsymbol{\phi}} = f - \frac{\alpha h}{2}\left(\frac{1}{m} f \nabla_{\boldsymbol{\phi}} f + g \nabla_{\boldsymbol{\theta}} f\right)$, $\dot{\boldsymbol{\theta}} = g - \frac{\lambda h}{2}\left((1 - \frac{2\alpha}{\lambda}) f \nabla_{\boldsymbol{\phi}} g + \frac{1}{k} g \nabla_{\boldsymbol{\theta}} g\right)$ (Theorem 3.2).

The stability of this system can be characterized by the modified Jacobian matrix evaluated at the equilibria of the two-player game. The equilibria that we are interested in for our stability analysis are the steady-state solutions of Equations (A.5) and (A.6), given by $f = \mathbf{0}, g = \mathbf{0}$. These are also equilibrium solutions for the steady-state modified equations[2] given by $\tilde{f} = \mathbf{0}, \tilde{g} = \mathbf{0}$.

The modified Jacobian can be written, using block matrix notation as:

$$\widetilde{J} = \begin{bmatrix} \nabla_{\boldsymbol{\phi}} \widetilde{f} & \nabla_{\boldsymbol{\theta}} \widetilde{f} \\ \nabla_{\boldsymbol{\phi}} \widetilde{g} & \nabla_{\boldsymbol{\theta}} \widetilde{g} \end{bmatrix} \tag{A.49}$$

Next, we calculate each term in this block matrix. (In the following analysis, each term is evaluated at an equilibrium solution given by $f = \mathbf{0}, g = \mathbf{0}$). We find:

$$\begin{aligned}
\nabla_{\boldsymbol{\phi}} \tilde{f} &= \nabla_{\boldsymbol{\phi}} f - \frac{\alpha h}{2}\left(\frac{1}{m}(\nabla_{\boldsymbol{\phi}} f)^2 + \frac{1}{m} f \nabla_{\boldsymbol{\phi},\boldsymbol{\phi}} f + \nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} f + g \nabla_{\boldsymbol{\phi},\boldsymbol{\theta}} f\right) \\
&= \nabla_{\boldsymbol{\phi}} f - \frac{\alpha h}{2}\left(\frac{1}{m}(\nabla_{\boldsymbol{\phi}} f)^2 + \nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} f\right) \\
\nabla_{\boldsymbol{\theta}} \tilde{f} &= \nabla_{\boldsymbol{\theta}} f - \frac{\alpha h}{2}\left(\frac{1}{m}\nabla_{\boldsymbol{\theta}} f \nabla_{\boldsymbol{\phi}} f + \frac{1}{m} f \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} f + \nabla_{\boldsymbol{\theta}} g \nabla_{\boldsymbol{\theta}} f + g \nabla_{\boldsymbol{\theta},\boldsymbol{\theta}} f\right) \\
&= \nabla_{\boldsymbol{\theta}} f - \frac{\alpha h}{2}\left(\frac{1}{m}\nabla_{\boldsymbol{\theta}} f \nabla_{\boldsymbol{\phi}} f + \nabla_{\boldsymbol{\theta}} g \nabla_{\boldsymbol{\theta}} f\right) \\
\nabla_{\boldsymbol{\phi}} \tilde{g} &= \nabla_{\boldsymbol{\phi}} g - \frac{\lambda h}{2}(\frac{1}{k}\nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} g + \frac{1}{k} g \nabla_{\boldsymbol{\phi},\boldsymbol{\theta}} g + (1 - \frac{2\alpha}{\lambda})\nabla_{\boldsymbol{\phi}} f \nabla_{\boldsymbol{\phi}} g + (1 - \frac{2\alpha}{\lambda}) f \nabla_{\boldsymbol{\phi},\boldsymbol{\phi}} g) \\
&= \nabla_{\boldsymbol{\phi}} g - \frac{\lambda h}{2}(\frac{1}{k}\nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} g + (1 - \frac{2\alpha}{\lambda})\nabla_{\boldsymbol{\phi}} f \nabla_{\boldsymbol{\phi}} g) \\
\nabla_{\boldsymbol{\theta}} \tilde{g} &= \nabla_{\boldsymbol{\theta}} g - \frac{\lambda h}{2}\left(\frac{1}{k}(\nabla_{\boldsymbol{\theta}} g)^2 + \frac{1}{k} g \nabla_{\boldsymbol{\theta},\boldsymbol{\theta}} g + (1 - \frac{2\alpha}{\lambda})\nabla_{\boldsymbol{\theta}} f \nabla_{\boldsymbol{\phi}} g + (1 - \frac{2\alpha}{\lambda}) f \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} g\right) \\
&= \nabla_{\boldsymbol{\theta}} g - \frac{\lambda h}{2}\left(\frac{1}{k}(\nabla_{\boldsymbol{\theta}} g)^2 + (1 - \frac{2\alpha}{\lambda})\nabla_{\boldsymbol{\theta}} f \nabla_{\boldsymbol{\phi}} g\right)
\end{aligned}$$

[2] There are additional steady-state solutions for the modified equations. However, we can ignore these since they are spurious solutions that do not correspond to steady states of the two-player game, arising instead as an artifact of the $\mathcal{O}(h^3)$ error in backward error analysis.

Given these calculations, we can now write:

$$\widetilde{J} = \begin{bmatrix} \nabla_{\boldsymbol{\phi}} \widetilde{f} & \nabla_{\boldsymbol{\theta}} \widetilde{f} \\ \nabla_{\boldsymbol{\phi}} \widetilde{g} & \nabla_{\boldsymbol{\theta}} \widetilde{g} \end{bmatrix} = J - \frac{h}{2} K_{\text{alt}} \tag{A.50}$$

where J is the Jacobian of the unmodified ODE:

$$J = \begin{bmatrix} \nabla_{\boldsymbol{\phi}} f & \nabla_{\boldsymbol{\theta}} f \\ \nabla_{\boldsymbol{\phi}} g & \nabla_{\boldsymbol{\theta}} g \end{bmatrix} \tag{A.51}$$

and

$$K_{\text{alt}} = \begin{bmatrix} \frac{\alpha}{m}(\nabla_{\boldsymbol{\phi}} f)^2 + \alpha \nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} f & \frac{\alpha}{m} \nabla_{\boldsymbol{\theta}} f \nabla_{\boldsymbol{\phi}} f + \alpha \nabla_{\boldsymbol{\theta}} g \nabla_{\boldsymbol{\theta}} f \\ \frac{\lambda}{k} \nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} g + \lambda(1 - \frac{2\alpha}{\lambda}) \nabla_{\boldsymbol{\phi}} f \nabla_{\boldsymbol{\phi}} g & \frac{\lambda}{k} (\nabla_{\boldsymbol{\theta}} g)^2 + \lambda(1 - \frac{2\alpha}{\lambda}) \nabla_{\boldsymbol{\theta}} f \nabla_{\boldsymbol{\phi}} g \end{bmatrix} \tag{A.52}$$

The modified system of equations are asymptotically stable if all the real parts of all the eigenvalues of the modified Jacobian are negative. If some of the eigenvalues are zero, the equilibrium may or may not be stable, depending on the non-linearities of the system. If the real parts of any eigenvalues are positive, then the dynamical system is unstable.

A necessary condition for stability is that the trace of the modified Jacobian is less than or equal to zero (i.e. $\operatorname{Tr}(\widetilde{J}) \leq 0$), since the trace is the sum of eigenvalues. Using the property of trace additivity and the trace cyclic property we see that:

$$\operatorname{Tr}(\widetilde{J}) = \operatorname{Tr}(J) - \frac{h}{2}\left(\frac{\alpha}{m} \operatorname{Tr}((\nabla_{\boldsymbol{\phi}} f)^2) + \frac{\lambda}{k} \operatorname{Tr}((\nabla_{\boldsymbol{\theta}} g)^2)\right) - \frac{h}{2}(\lambda - \alpha) \operatorname{Tr}(\nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} f) \tag{A.53}$$

We note that unlike for simultaneous updates, even if $\operatorname{Tr}(\nabla_{\boldsymbol{\phi}} g \nabla_{\boldsymbol{\theta}} f)$ is negative, if $\lambda < \alpha$, the trace of the modified system will stay negative, so the necessary condition for the system to remain stable is still satisfied. However, since this is not a sufficient condition, the modified system could still be unstable.

E.3. A new tool for stability analysis: different ODEs for simultaneous and alternating Euler updates

We will now highlight how having access to different systems of ODEs to closely describe the dynamics of simultaneous and alternating Euler updates can be a useful tool for stability analysis. Stability analysis has been used to understand the local convergence properties of games such as GANs (Nagarajan & Kolter, 2017). Thus far, this type of analysis has relied on the original ODEs describing the game and has ignored discretization drift. Moreover, since no ODEs were available to capture the difference in dynamics between simultaneous and alternating Euler updates, alternating updates have not been studied using stability analysis, despite being predominantly used in practice by practitioners. We will use an illustrative example to show how the modified ODEs we have derived using backward error analysis can be used to analyze the local behaviour of Euler updates and uncover the different behavior of simultaneous and alternating updates. As before, we use the ODEs

$$\dot{\phi} = f(\phi, \theta) = -\epsilon_1 \phi + \theta; \qquad \dot{\theta} = g(\phi, \theta) = \epsilon_2 \theta - \phi$$

We set $\epsilon_1 = 0.09, \epsilon_2 = 0.09$ and a learning rate 0.2 and show the behavior of the original flow as well as simultaneous and alternating Euler updates and their corresponding modified flows in Figure A.1. By replacing the values of f and g into the results for the Jacobian of the modified ODEs obtain above, we obtain the corresponding Jacobians. For simultaneous updates:

$$\tilde{J}_{\text{sim}} = \begin{bmatrix} -\epsilon_1 - h/2\epsilon_1^2 + h/2 & 1 + h/2\epsilon_1 - h/2\epsilon_2 \\ -1 - h/2\epsilon_1 + h/2\epsilon_2 & \epsilon_2 + h/2 - h/2\epsilon_2^2 \end{bmatrix} \tag{A.54}$$

For alternating updates:

$$\tilde{J}_{\text{alt}} = \begin{bmatrix} -\epsilon_1 - h/2\epsilon_1^2 + h/2 & 1 + h/2\epsilon_1 - h/2\epsilon_2 \\ -1 + h/2\epsilon_1 + h/2\epsilon_2 & \epsilon_2 - h/2 - h/2\epsilon_2^2 \end{bmatrix} \tag{A.55}$$

When we replace the values of $\epsilon_1 = \epsilon_2 = 0.09$ and $h = 0.2$, we obtain that $\operatorname{Tr}(\tilde{J}_{\text{sim}}) = 0.19 > 0$, thus the system of the modified ODEs corresponding to simultaneous Euler updates diverge. For alternating updates, we obtain $\operatorname{Tr}(\tilde{J}_{\text{alt}}) =$

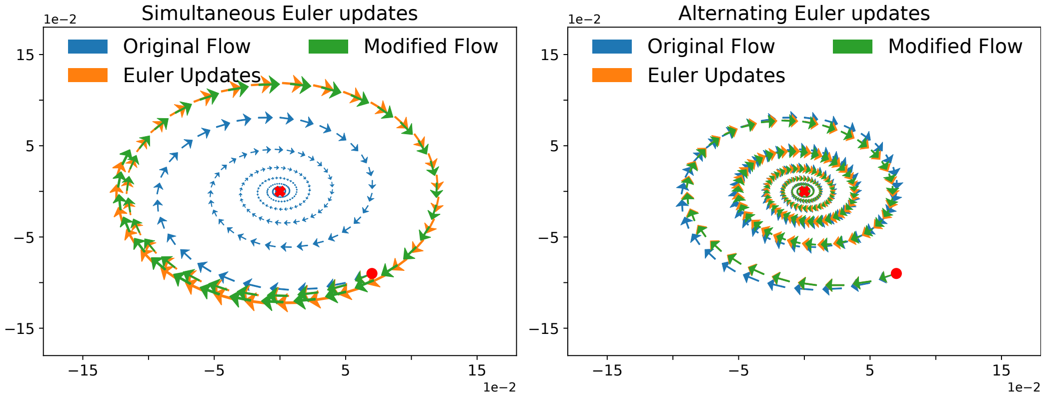


Figure A.1. Discretization drift can change the stability of a game. $\epsilon_1 = \epsilon_2 = 0.09$ and a learning rate 0.2.

$-0.0016 < 0$ and $|\tilde{J}_{\text{alt}}| = 0.981 > 0$, leading to a stable system. Our analysis is thus consistent with what we observe empirically in Figure A.1. We observe the same results for other choices of ϵ_1, ϵ_2 and h.

Benefits and caveats of using the modified ODEs for stability analysis: Using the modified ODEs we have derived using backward error analysis allows us to study systems which are closer to the discrete Euler updates, as they do not ignore discretization drift. The modified ODEs also provide a tool to discriminate between alternating and simultaneous Euler updates in two-player games when performing stability analysis to understand the discrete behavior of the system. Accounting for the drift in the analysis of two-player games is crucial, since as we have seen, the drift can change a stable equilibrium into an unstable one – which is not the case for supervised learning (Barrett & Dherin, 2021). These strong benefits of using the modified ODEs we propose for stability analysis also come with caveats: the modified ODEs nonetheless ignore higher order drift terms, and our approximations might not hold for very large learning rates.

F. SGA in two-player games

For clarity, this section reproduces Symplectic Gradient Adjustment (SGA) from Balduzzi et al. (2018) for two-player games, using notations consistent with this paper. If we have two players $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ minimizing loss functions L_1 and L_2, SGA defines the vector:

$$\boldsymbol{\xi} = \begin{bmatrix} \nabla_{\boldsymbol{\phi}} L_1 \\ \nabla_{\boldsymbol{\theta}} L_2 \end{bmatrix} \tag{A.56}$$

As defined in SGA, $\boldsymbol{\xi}$ is the *negative of the vector field* that the system dynamics follow. This entails that according to our notation:

$$\boldsymbol{\xi} = \begin{bmatrix} -f \\ -g \end{bmatrix} \tag{A.57}$$

defining the Jacobian:

$$J_{\boldsymbol{\xi}} = \begin{bmatrix} -\nabla_{\boldsymbol{\phi}} f & -\nabla_{\boldsymbol{\theta}} f \\ -\nabla_{\boldsymbol{\phi}} g & -\nabla_{\boldsymbol{\theta}} g \end{bmatrix} \tag{A.58}$$

Since in SGA $f = \nabla_{\boldsymbol{\phi}} L_1$ and $g = -\nabla_{\boldsymbol{\theta}} L_2$ the Jacobian has as an *anti-symmetric* component

$$A = \frac{1}{2}(J_{\boldsymbol{\xi}} - J_{\boldsymbol{\xi}}^T) = \frac{1}{2}\begin{bmatrix} 0 & -\nabla_{\boldsymbol{\theta}} f + \nabla_{\boldsymbol{\phi}} g \\ -\nabla_{\boldsymbol{\phi}} g + \nabla_{\boldsymbol{\theta}} f & 0 \end{bmatrix} \tag{A.59}$$

Ignoring the sign change from alignment (Balduzzi et al., 2018) for simplicity, the vector field $\boldsymbol{\xi}$ is modified according to SGA as

$$\hat{\boldsymbol{\xi}} = \boldsymbol{\xi} + A^T \boldsymbol{\xi} = \begin{bmatrix} -f \\ -g \end{bmatrix} + \frac{1}{2}\begin{bmatrix} (\nabla_{\boldsymbol{\phi}} g - \nabla_{\boldsymbol{\theta}} f)^T \, g \\ (\nabla_{\boldsymbol{\theta}} f - \nabla_{\boldsymbol{\phi}} g)^T \, f \end{bmatrix} \tag{A.60}$$

For zero-sum games, we have that $f = -\nabla_{\boldsymbol{\phi}} E$ and $g = \nabla_{\boldsymbol{\theta}} E$ and thus:

$$\nabla_{\boldsymbol{\phi}} g^T g = -\nabla_{\boldsymbol{\theta}} f^T g = \frac{1}{2} \nabla_{\boldsymbol{\phi}} \left\| \nabla_{\boldsymbol{\theta}} E \right\|^2$$
$$\nabla_{\boldsymbol{\theta}} f^T f = -\nabla_{\boldsymbol{\phi}} g^T f = \frac{1}{2} \nabla_{\boldsymbol{\theta}} \left\| \nabla_{\boldsymbol{\phi}} E \right\|^2$$

Thus the modified gradient field can be simplified to

$$\hat{\boldsymbol{\xi}} = \begin{bmatrix} -f \\ -g \end{bmatrix} + \begin{bmatrix} \nabla_{\boldsymbol{\phi}} g^T g \\ \nabla_{\boldsymbol{\theta}} f^T f \end{bmatrix} = \begin{bmatrix} \nabla_{\boldsymbol{\phi}} E \\ -\nabla_{\boldsymbol{\theta}} E \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \nabla_{\boldsymbol{\phi}} \left\| \nabla_{\boldsymbol{\theta}} E \right\|^2 \\ \nabla_{\boldsymbol{\theta}} \left\| \nabla_{\boldsymbol{\phi}} E \right\|^2 \end{bmatrix} = \begin{bmatrix} \nabla_{\boldsymbol{\phi}} (E + \frac{1}{2} \left\| \nabla_{\boldsymbol{\theta}} E \right\|^2) \\ \nabla_{\boldsymbol{\theta}} (-E + \frac{1}{2} \left\| \nabla_{\boldsymbol{\phi}} E \right\|^2) \end{bmatrix} \tag{A.61}$$

Therefore, since $\hat{\boldsymbol{\xi}}$ defines the negative of the vector field followed by the system, the modified losses for the two players can be written respectively as:

$$\widetilde{L}_1 = E + \frac{1}{2} \left\| \nabla_{\boldsymbol{\theta}} E \right\|^2 \tag{A.62}$$
$$\widetilde{L}_2 = -E + \frac{1}{2} \left\| \nabla_{\boldsymbol{\phi}} E \right\|^2 \tag{A.63}$$

The functional form of the modified losses given by SGA is the same used to cancel the interaction terms of DD in the case of simultaneous gradient descent updates in zero sum games. We do however highlight a few differences in our approach compared to SGA: our approach extends to alternating updates and provides the optimal regularization coefficients; canceling the interaction terms of the drift is different compared to SGA for general games (see Equation A.60 for SGA and Theorem 3.1 for the interaction terms of DD).

G. DiracGAN - an illustrative example

In their work assessing the convergence of GAN training Mescheder et al. (2018) introduce the example of the DiracGAN, where the GAN is trying to learn a delta distribution with mass at zero. More specifically, the generator $G_\theta(z) = \theta$ with parameter θ parametrizes constant functions whose images $\{\theta\}$ correspond to the support of the delta distribution δ_θ. The discriminator is a linear model $D_\phi(x) = \phi \cdot x$ with parameter ϕ.

The loss function is given by:

$$E(\theta, \phi) = l(\theta\phi) + l(0) \tag{A.64}$$

where f depends on the GAN used - for the standard GAN it is $l = -\log(1 + e^{-t})$. As in Mescheder et al. (2018), we assume l is continuously differentiable with $l'(x) \neq 0$ for all $x \in \mathbb{R}$. The partial derivatives of the loss function

$$\frac{\partial E}{\partial \phi} = l'(\theta\phi)\theta, \qquad \frac{\partial E}{\partial \theta} = l'(\theta\phi)\phi,$$

lead to the underlying continuous dynamics:

$$\dot{\phi} = f(\theta, \phi) = l'(\theta\phi)\theta, \qquad \dot{\theta} = g(\theta, \phi) = -l'(\theta\phi)\phi. \tag{A.65}$$

Thus the only equilibrium of the game is $\theta = 0$ and $\phi = 0$.

G.1. Reconciling discrete and continuous updates in Dirac-GAN

Mescheder et al. (2018) observed a discrepancy between the continuous and discrete dynamics. They show that, for the problem in Equation (A.64), the continuous dynamics preserve $\theta^2 + \phi^2$, and thus cannot converge (Lemma 2.3 in Mescheder et al. (2018)), since:

$$\frac{d\left(\theta^2 + \phi^2\right)}{dt} = 2\theta \frac{d\theta}{dt} + 2\phi \frac{d\phi}{dt} = -2\theta l'(\theta\phi)\phi + 2\phi l'(\theta\phi)\theta = 0.$$

They also observe that with the discrete dynamics of simultaneous gradient descent that $\theta^2 + \phi^2$ increases in time (Lemma 2.4 in Mescheder et al. (2018)). We resolve this discrepancy here, by showing that the modified continuous dynamics given by discretization drift result in behavior consistent with that of the discrete updates.

Proposition G.1 *The continuous vector field given by discretization drift for simultaneous Euler updates in DiracGAN increases $\theta^2 + \phi^2$ in time.*

Proof: We assume both the generator and the discriminator use learning rates h, as in (Mescheder et al., 2018). We first compute terms used by the zero-sum Collloraries in the main paper.

$$\|\nabla_\theta E\|^2 = l'(\theta\phi)^2\phi^2$$
$$\|\nabla_\phi E\|^2 = l'(\theta\phi)^2\theta^2$$

and

$$\nabla_\theta \|\nabla_\theta E\|^2 = 2\phi^3 l'(\theta\phi)l''(\theta\phi)$$
$$\nabla_\theta \|\nabla_\phi E\|^2 = 2\theta l'(\theta\phi)^2 + 2\theta^2\phi l'(\theta\phi)l''(\theta\phi)$$
$$\nabla_\phi \|\nabla_\theta E\|^2 = 2\phi l'(\theta\phi)^2 + 2\phi^2\theta l'(\theta\phi)l''(\theta\phi)$$
$$\nabla_\phi \|\nabla_\phi E\|^2 = 2\theta^3 l'(\theta\phi)l''(\theta\phi)$$

Thus, the modified ODEs are given by:

$$\dot{\phi} = l'(\theta\phi)\theta + \frac{h}{2}\left[-\theta^3 l'(\theta\phi)l''(\theta\phi) + \phi l'(\theta\phi)^2 + \phi^2\theta l'(\theta\phi)l''(\theta\phi)\right]$$
$$\dot{\theta} = -l'(\theta\phi)\phi - \frac{h}{2}\left[-\theta l'(\theta\phi)^2 - \theta^2\phi l'(\theta\phi)l''(\theta\phi) + \phi^3 l'(\theta\phi)l''(\theta\phi)\right]$$

By denoting $l'(\theta\phi)$ by l' and $l''(\theta\phi)$ by l'', then we have:

$$\begin{aligned}
\frac{d\left(\theta^2+\phi^2\right)}{dt} &= 2\theta\frac{d\theta}{dt} + 2\phi\frac{d\phi}{dt} \\
&= 2\theta\left(-l'\phi - \frac{h}{2}\left[-\theta l'^2 - \theta^2\phi l' l'' + \phi^3 l' l''\right]\right) + 2\phi\left(l'\theta + \frac{h}{2}\left[-\theta^3 l' l'' + \phi l'^2 + \phi^2\theta l' l''\right]\right) \\
&= 2\theta\left(-\frac{h}{2}\left[-\theta l'^2 - \theta^2\phi l' l'' + \phi^3 l' l''\right]\right) + 2\phi\left(\frac{h}{2}\left[-\theta^3 l' l'' + \phi l'^2 + \phi^2\theta l' l''\right]\right) \\
&= h\theta^2 l'^2 + h\theta^3\phi l' l'' - h\phi^3\theta l' l'' - h\phi\theta^3 l' l'' + h\phi^2 l'^2 + h\phi^3\theta l' l'' \\
&= h\theta^2 l'^2 + h\phi^2 l'^2 > 0
\end{aligned}$$

for all $\phi, \theta \neq 0$, which shows that $\theta^2 + \phi^2$ is not preserved and it will strictly increase for all values away from the equilibrium (we have used the assumption that $l'(x) \neq 0, \forall x \in \mathbb{R}$). □

We have thus identified a continuous system which exhibits the same behavior as described by Lemma 2.4 in Mescheder et al. (2018), where a discrete system is analysed. Figure A.2 illustrates that the divergent behavior of simultaneous gradient descent in this case can be predicted from the dynamics of the modified continuous system given by backward error analysis.

G.2. DD changes the convergence behavior of Dirac-GAN

The Jacobian of the unmodified Dirac-GAN evaluated at the equilibrium solution given by $\phi = 0$, $\theta = 0$ is given by

$$J = \begin{bmatrix} \nabla_{\phi,\phi}E & \nabla_{\theta,\phi}E \\ -\nabla_{\phi,\theta}E & -\nabla_{\theta,\theta}E \end{bmatrix} = \begin{bmatrix} 0 & l'(0) \\ -l'(0) & 0 \end{bmatrix} \tag{A.66}$$

We see that $\mathrm{Tr}(J) = 0$ and the determinant $|J| = l'(0)^2$. Therefore, the eigenvalues of this Jacobian are $\lambda_\pm = \mathrm{Tr}(J)/2 \pm \sqrt{\mathrm{Tr}(J)^2 - 4|J|}/2 = \pm i l'(0)$ (Reproduced from Mescheder et al. (2018)). This is an example of a stable *center equilibrium*.

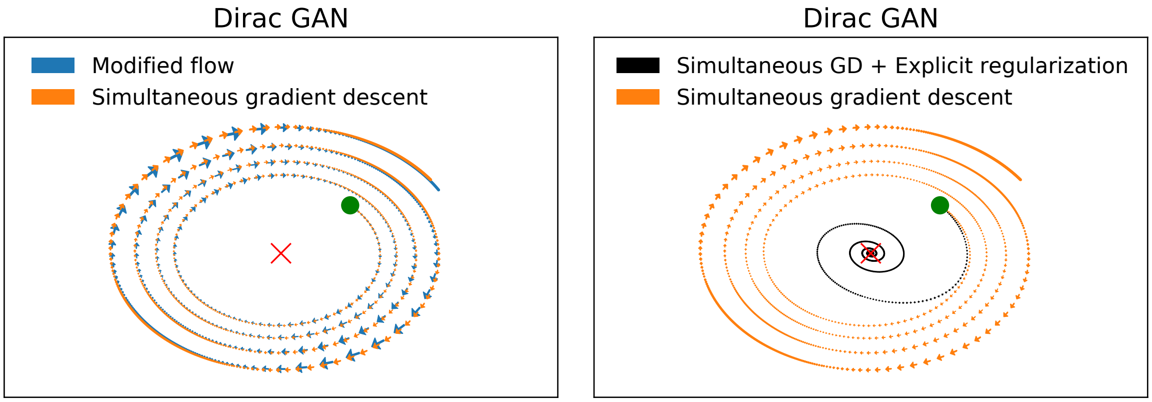


Figure A.2. DiracGAN. Left: The dynamics of simultaneous gradient descent updates match the continuous dynamics given by backward error analysis for DiracGAN. Right: Explicit regularization canceling the interaction terms of DD stabilizes the DiracGAN trained with simultaneous gradient descent.

Next we calculate the Jacobian of the modified ODEs given by DiracGAN, evaluated at an equilibrium solution and find $\widetilde{J} = J - h\Delta/2$, where

$$\begin{aligned}\Delta &= \begin{bmatrix} \alpha(\nabla_{\phi,\phi}E)^2 - \alpha\nabla_{\phi,\theta}E\nabla_{\theta,\phi}E & \alpha\nabla_{\theta,\phi}E\nabla_{\phi,\phi}E - \alpha\nabla_{\theta,\theta}E\nabla_{\theta,\phi}E \\ \lambda\nabla_{\phi,\theta}E\nabla_{\theta,\theta}E - \lambda\nabla_{\phi,\phi}E\nabla_{\phi,\theta}E & \lambda(\nabla_{\theta,\theta}E)^2 - \lambda\nabla_{\theta,\phi}E\nabla_{\phi,\theta}E \end{bmatrix} \\ &= \begin{bmatrix} -\alpha\nabla_{\phi,\theta}E\nabla_{\theta,\phi}E & 0 \\ 0 & -\lambda\nabla_{\theta,\phi}E\nabla_{\phi,\theta}E \end{bmatrix} \\ &= \begin{bmatrix} -\alpha l'(0)^2 & 0 \\ 0 & -\lambda l'(0)^2 \end{bmatrix}\end{aligned}$$

so

$$\widetilde{J} = \begin{bmatrix} h\alpha l'(0)^2/2 & l'(0) \\ -l'(0) & h\lambda l'(0)^2/2 \end{bmatrix} \tag{A.67}$$

Now, we see that the trace of the modified Jacobian for the Dirac-GAN is $\mathrm{Tr}(\widetilde{J}) = (h/2)(\alpha+\lambda)l'(0)^2 > 0$, so the modified ODEs induced by gradient descent in DiracGAN are unstable.

G.3. Explicit regularization stabilizes Dirac-GAN

Here, we show that we can use our stability analysis to identify forms of explicit regularization that can counteract the destabilizing impact of DD. Consider the Dirac-GAN with explicit regularization of the following form: $L_1 = -E + u\|\nabla_\theta E\|^2$ and $L_2 = E + \nu\|\nabla_\phi E\|^2$ where $\phi_t = \phi_{t-1} - \alpha h\nabla_\phi L_1$ and $\theta_t = \theta_{t-1} - \lambda h \nabla_\theta L_2$ and with $u, \nu \sim \mathcal{O}(h)$. The modified Jacobian for this system is given by

$$\begin{aligned}\widetilde{J} &= \begin{bmatrix} h\alpha/2\nabla_{\phi,\theta}E\nabla_{\theta,\phi}E - u\nabla_{\phi,\phi}\left\|\nabla_\theta E\right\|^2 & \nabla_{\theta,\phi}E \\ -\nabla_{\phi,\theta}E & h\lambda/2\nabla_{\theta,\phi}E\nabla_{\phi,\theta}E - \nu\nabla_{\theta,\theta}\left\|\nabla_\phi E\right\|^2 \end{bmatrix} \\ &= \begin{bmatrix} (h\alpha/2 - 2u)l'(0)^2 & l'(0) \\ -l'(0) & (h\lambda/2 - 2\nu)l'(0)^2 \end{bmatrix}\end{aligned}$$

The determinant of the modified Jacobian is $|\widetilde{J}| = (h\alpha/2 - 2u)(h\lambda/2 - 2\nu)l'(0)^4 + l'(0)^2$ and the trace is $\mathrm{Tr}(\widetilde{J}) = (h\alpha/2 - 2u)l'(0)^2 + (h\lambda/2 - 2\nu)l'(0)^2$. A necessary and sufficient condition for asymptotic stability is $|\widetilde{J}| > 0$ and $\mathrm{Tr}(\widetilde{J}) < 0$ (since this guarantees that the eigenvalues of the modified Jacobian have negative real part). Therefore, if $u > h\alpha/4$ and $\nu > h\lambda/4$, the system is asymptotically stable. We note however that in practice, when using discrete updates, the exact threshold for stability will have a $\mathcal{O}(h^3)$ correction, arising from the $\mathcal{O}(h^3)$ error in our backward error analysis. Also, we see that when $u = h\alpha/4$ and $\nu = h\lambda/4$, the contribution of the cross-terms is cancelled out (up to an

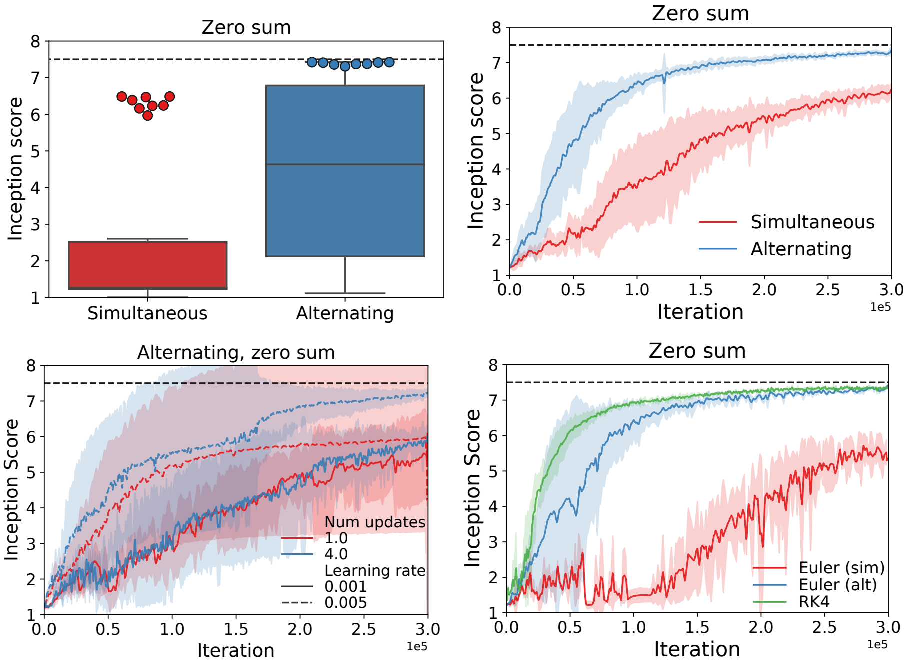


Figure A.3. The effect of discretization drift on zero-sum games.

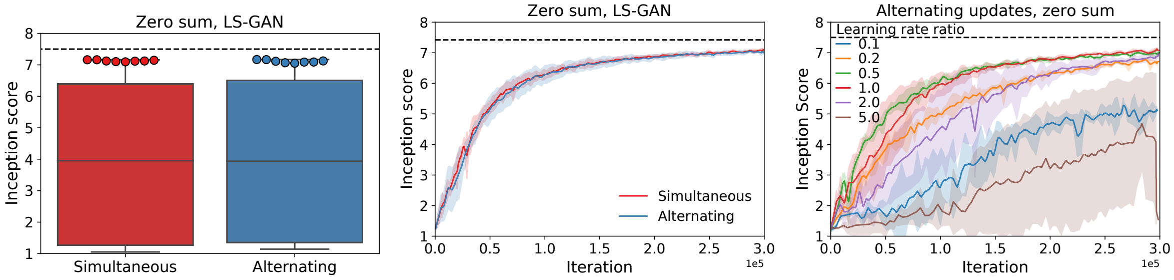


Figure A.4. Least squares GAN: The effect of discretization drift on zero-sum games.

$\mathcal{O}(h^3)$ correction). In Figure A.2, we see an example where this explicit regularization stabilizes the DiracGAN, so that it converges toward the equilibrium solution.

H. Additional experimental results

H.1. Additional results using zero-sum GANs

We show additional results showcasing the effect of DD on zero-sum games in Figure A.3. We see that not only do simultaneous updates perform worse than alternating updates when using the best hyperparameters, but that simultaneous updating is much more sensitive to hyperparameter choice. We also see that multiple updates can improve the stability of a GAN trained using zero-sum losses, but this strongly depends on the choice of learning rate.

H.1.1. Least squares GANs

In order to assess the robustness of our results independent of the GAN loss used, we perform additional experimental results using GANs trained with a least square loss (LS-GAN (Mao et al., 2017)). We show results in Figure A.4, where we see that for the least square loss too, the learning rate ratios for which the generator drift does not maximize the discriminator norm (learning rate ratios above or equal to 0.5) perform best and exhibit less variance.

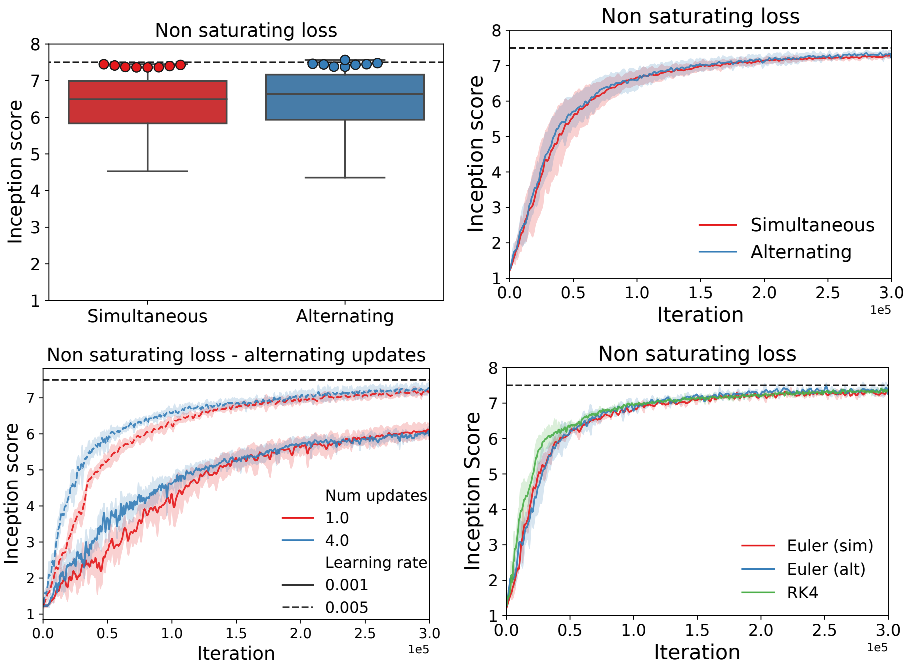


Figure A.5. The effect of discretization drift depends on the game: with the non saturating loss, the relative performance of different numerical estimators is different compared to the saturating loss, and the effect of explicit regularization is also vastly different.

H.2. GANs using the non-saturating loss

Next, we explore how the strength of the DD depends on the game dynamics. We do this by comparing the *relative effect* that numerical integration schemes have across different games. To this end, we consider the non-saturating loss introduced in the original GAN paper $(-\log D_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\mathbf{z})))$. This loss has been extensively used since it helps to avoid problematic gradients early in training. When using this loss, we see that there is little difference between simultaneous and alternating updates (Figure A.5), unlike the saturating loss case. These results demonstrate that since DD depends on the underlying dynamics, it is difficult to make general game-independent predictions about which numerical integrator will perform best.

H.3. Explicit regularization in zero-sum games trained using simultaneous gradient descent

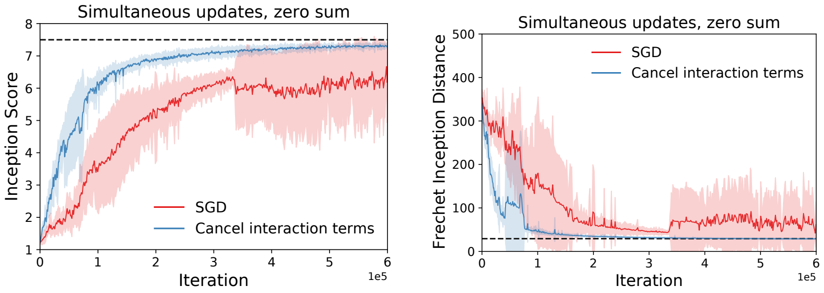
We now show additional experimental results and visualizations obtained using explicit regularization obtained using the original zero sum GAN objective, as presented in Goodfellow et al. (2014).

We show the improvement that can be obtained compared to gradient descent with simultaneous updates by canceling the interaction terms in Figure A.6. We additionally show results obtained from strengthening the self terms in Figure A.8.

In Figure A.7, we show that by cancelling interaction terms, SGD becomes a competitive optimization algorithm when training the original GAN. We also note that in the case of Adam, while convergence is substantially faster than with SGD, we notice a degrade in performance later in training. This is something that has been observed in other works as well (e.g. (Qin et al., 2020)).

H.3.1. More percentiles

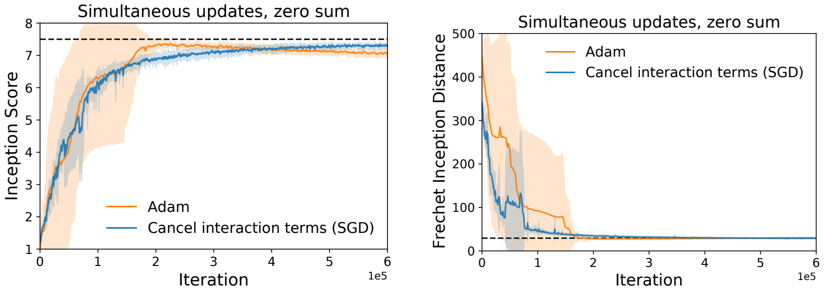
Throughout the main paper, we displayed the best 10% performing models for each optimization algorithm used. We now expand that to show performance results across the best 20% and 30% of models in Figure A.9. We observe a consistent increase in performance obtained by canceling the interaction terms.



(a) Inception Score (↑).

(b) Frechet Inception Distance (↓).

Figure A.6. Using explicit regularization to cancel the effect of the interaction components of drift eads to a substantial improvement compared to SGD without explicit regularization.



(a) Inception Score (↑).

(b) Frechet Inception Distance (↓).

Figure A.7. Using explicit regularization to cancel the effect of the interaction components of drift allows us to obtain the same peak performance as Adam using SGD without momentum.

H.3.2. Batch size comparison

We now show that the results which show the efficacy of canceling the interaction terms are resiliant to changes in batch size in Figure A.10.

H.3.3. Comparison with Symplectic Gradient Adjustment

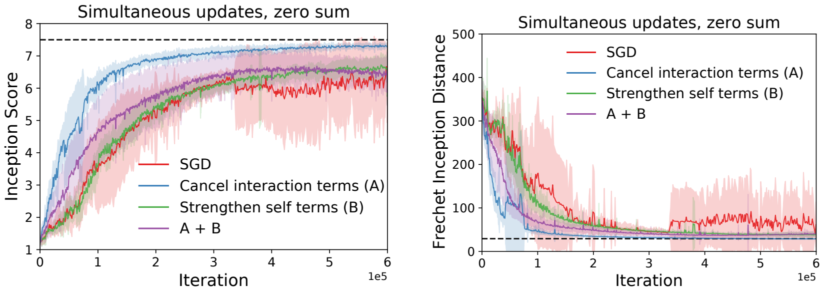
We show results comparing with Symplectic Gradient Adjustment (SGA) (Balduzzi et al., 2018) in Figure A.11 (best performing models) and Figure A.12 (quantiles showing performance across all hyperparameters and seeds). We observe that despite having the same functional form as SGA, canceling the interaction terms of DD performs better; this is due to the choice of regularization coefficients, which in the case of canceling the interaction terms is provided by Corollary 6.1.

H.3.4. Comparison with Consensus Optimization

We show results comparing with Consensus Optimization (CO) (Mescheder et al., 2017) in Figure A.13 (best performing models) and Figure A.14 (quantiles showing performance across all hyperparameters and seeds). We observe that canceling the interaction terms performs best, and that additionally strengthening the self terms does not provide a performance benefit.

H.3.5. Variance across seeds

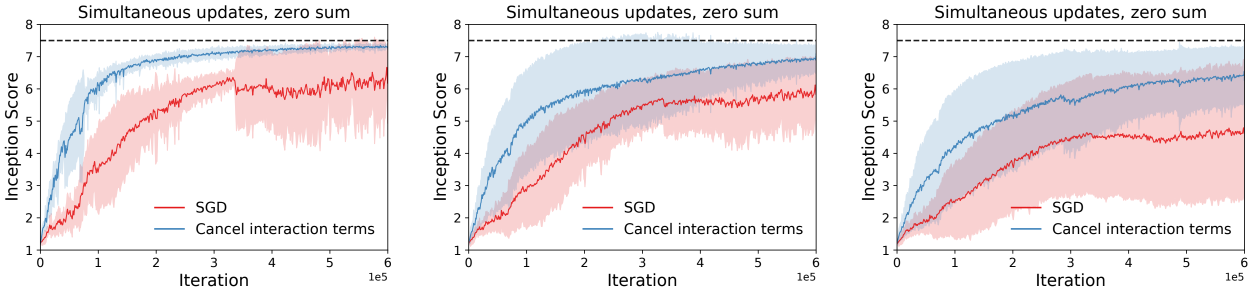
We have mentioned in the main text the challenge with variance across seeds observed when training GANs with SGD, especially in the case of simultaneous updates in zero sum games. We first notice that performance of simultaneous updates depends strongly on learning rates, with most models not learning. We also notice variance across seeds, both in vanilla SGD and when using explicit regularization to cancel interaction terms. In order to investigate this effect, we ran a sweep of 50 seeds for the best learning rates we obtain when canceling interaction terms in simultaneous updates, namely a discriminator learning rate of 0.01 and a generator learning rate of 0.005, we obtain Inception Score results with mean 5.61, but a very



(a) Inception Score (↑).

(b) Frechet Inception Distance (↓).

Figure A.8. Using explicit regularization to cancel the effect of the drift eads to a substantial improvement compared to SGD without explicit regularization. Strengthening the self terms – the terms which minimize the player's own norm – does not lead to a substantial improvement; this is somewhat expected since while the modified ODEs give us the exact coefficients required to *cancel* the drift, they do not tell us how to strengthen it, and our choice of exact coefficients from the drift might not be optimal.



(a) Top 10% models.

(b) Top 20% models.

(c) Top 30% models.

Figure A.9. Performance across the top performing models for vanilla SGD and with canceling the interaction terms. Across all percentages, canceling the interaction terms improves performance.

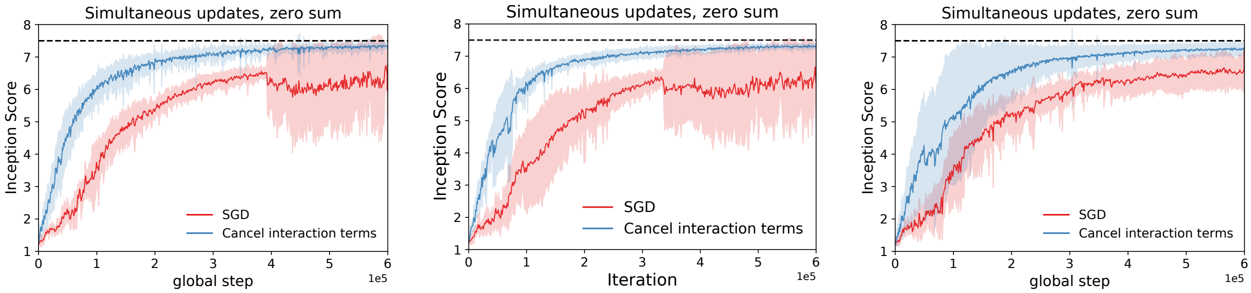
large standard deviation of 2.34. Indeed, as shown in Figure A.15, more than 50% of the seeds converge to an IS grater than 7. To investigate the reason for the variability, we repeat the same experiment, but clip the gradient value for each parameter to be in $[-0.1, 0.1]$, and show results in Figure A.16. We notice that another 10% of jobs converge to an IS score grater than 7, and 10% drop in the number of jobs that do not manage to learn. This makes us postulate that the reason for the variability is due to large gradients, perhaps early in training. We contrast this variability across seeds with the consistent performance we obtain by looking at the best performing *models* across learning rates, where as we have shown in the main paper and throughout the Supplementary Material, we obtain consistent performance which consistently leads to a substantial improvement compared to SGD without explicit regularization, and obtains performance comparable with Adam.

H.4. Explicit regularization in zero-sum games trained using alternating gradient descent

We perform the same experiments as done for simultaneous updates also with alternating updates. To do so, we cancel the effect of DD using the same explicit regularization functional form, but updating the coefficients to be those of alternating updates. We show results in Figure A.17, where we see very little difference in the results compared to vanilla SGD, perhaps apart from less instability early on in training. We postulate that this could be due the effect of DD in alternating updates can can be beneficial, especially for learning rate ratios for which the second player also minimizes the gradient norm of the first player. We additionally show results obtained from strengthening the self terms in Figure A.18.

H.4.1. More percentiles

Throughout the main paper, we displayed the best 10% performing models for each optimization algorithm used. We now expand that to show performance results across the 20% and 30% jobs in Figure A.19. We observe a consistent increase in performance obtained by canceling the interaction terms.

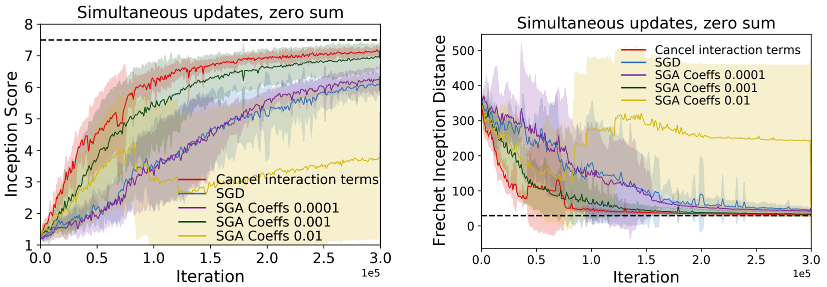


(a) Batch size 64.

(b) Batch size 128.

(c) Batch size 256.

Figure A.10. Performance when changing the batch size. We consistently see that canceling te interaction terms improves performance.



(a) Inception Score (↑).

(b) Frechet Inception Distance (↓).

Figure A.11. Comparison with Symplectic Gradient Adjustment (SGA). Canceling the interaction terms results in similar performance, but with less variance. The performance of SGA heavily depends on the strength of regularization, adding another hyperparmeter to the hyperparameter sweep, while canceling the interaction terms of the drift requires no other hyperparameters, since the explicit regularization coefficients strictly depend on learning rates.

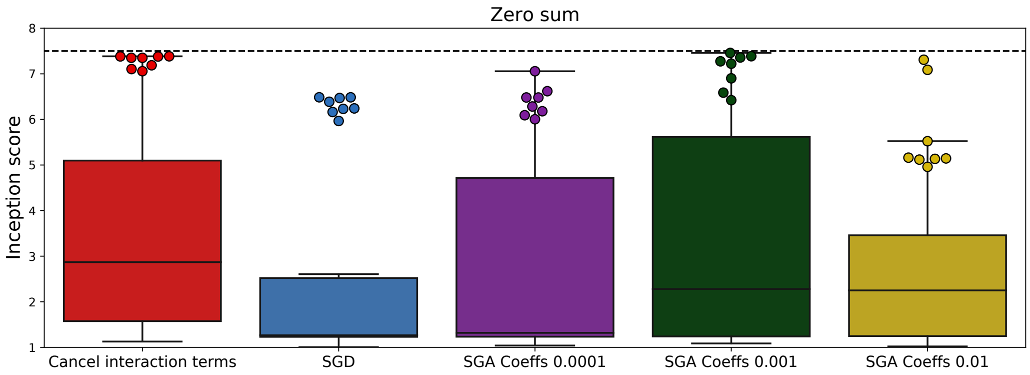
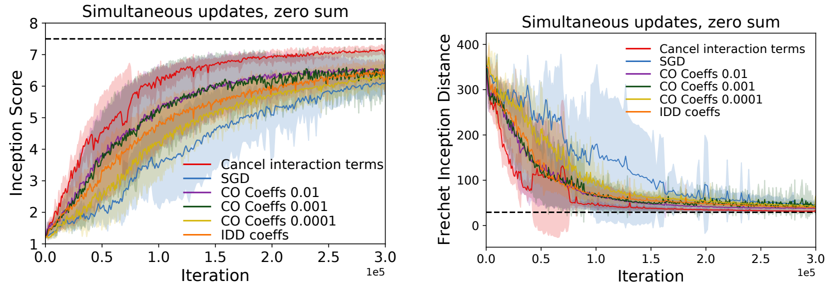


Figure A.12. Comparison with Symplectic Gradient Adjustment (SGA), obtained from *all models in the sweep*. Without requiring an additional sweep over the regularization coefficient, canceling the interaction terms results in better performance across the learning rate sweep and less sensitivity to hyperparameters.



(a) Inception Score (↑).

(b) Frechet Inception Distance (↓).

Figure A.13. Comparison with Consensus Optimization (CO). Despite not requiring additional hyperparameters compared to the standard SGD learning rate sweep, canceling the interaction terms of the drift performs better than consensus optimization. Using consensus optimization with a fixed coefficient can perform better than using the drift coefficients when we use them to the strengthen the norms – this is somewhat expected since while the modified ODEs give us the exact coefficients required to *cancel* the drift, they do not tell us how to strengthen it, and our choice of exact coefficients from the drift might not be optimal.

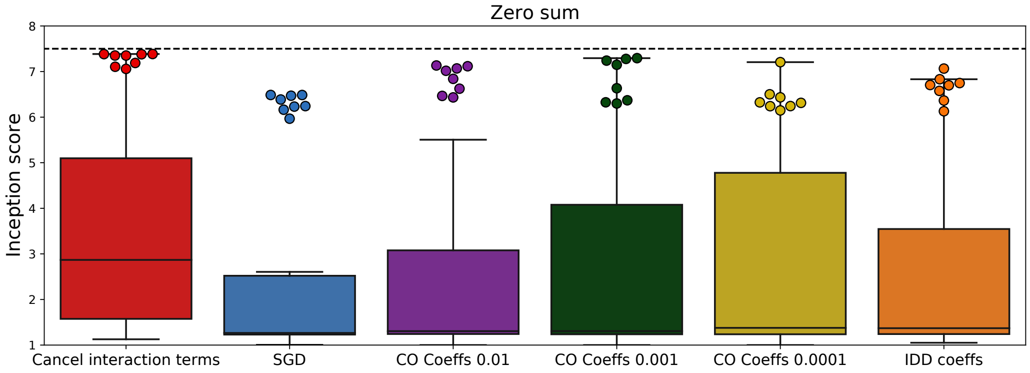
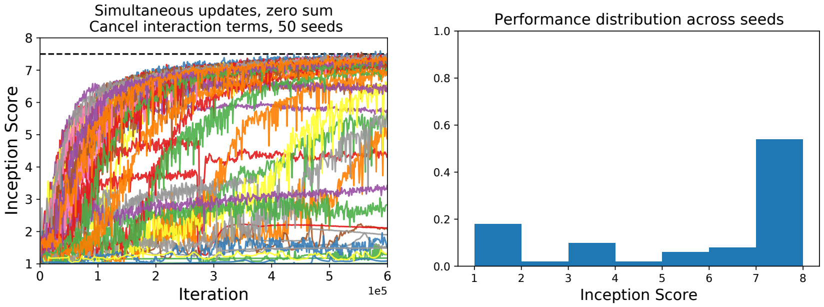


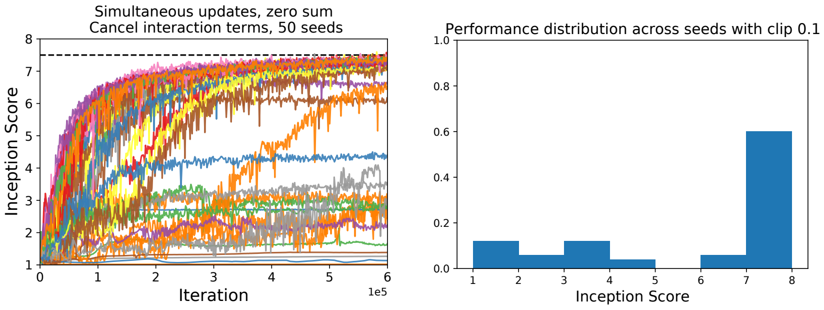
Figure A.14. Comparison with Consensus Optimization (CO), obtained from *all models in the sweep*. Without requiring an additional sweep over the regularization coefficient, canceling the interaction terms results in better performance across the learning rate sweep and less sensitivity to hyperparameters.



(a) Learning curves.

(b) Performance Histogram.

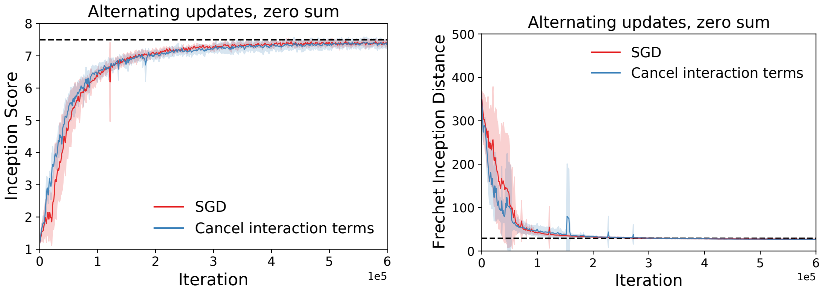
Figure A.15. Variability across seeds for the best performing hyperparameters, when canceling interaction terms in simultaneous updates for the original GAN, with a zero sum loss.



(a) Learning curves.

(b) Performance Histogram.

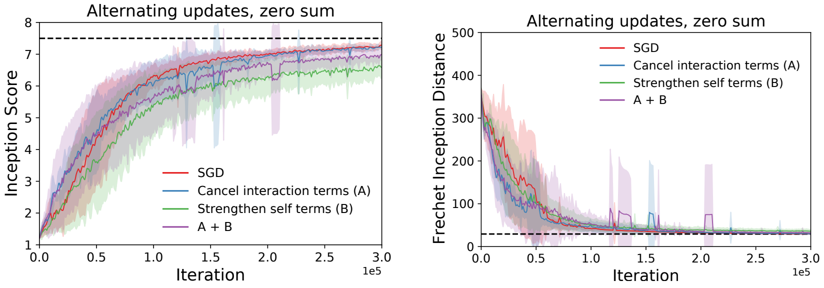
Figure A.16. With gradient clipping. Gradient clipping can reduce variability. This suggests that the instabilities observed in gradient descent are caused by large gradient updates.



(a) Inception Score (↑).

(b) Frechet Inception Distance (↓).

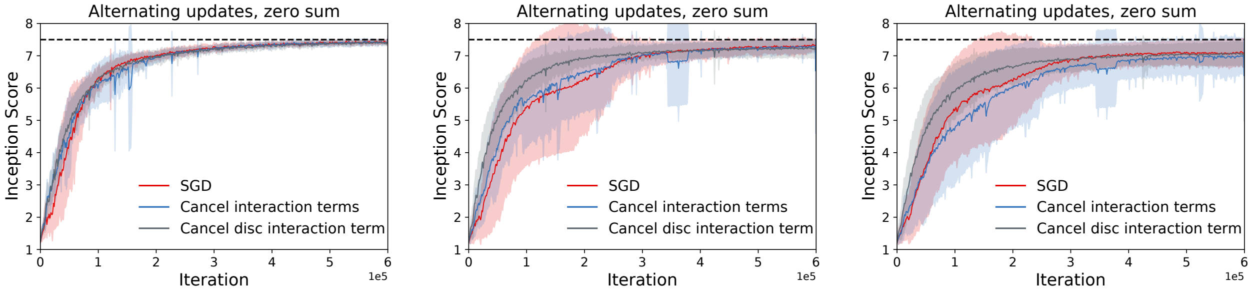
Figure A.17. In alternating updates, using explicit regularization to cancel the effect of the interaction components of drift does not substantially improve performance compared to SGD, but can reduce variance. This is expected, given that the interaction terms for the second player in the case of alternating updates can have a beneficial regularization effect.



(a) Inception Score (↑).

(b) Frechet Inception Distance (↓).

Figure A.18. In alternating updates, using explicit regularization to cancel the effect of the interaction components of drift does not substantially improve performance compared to SGD, but can reduce variance. This is expected, given that the interaction terms for the second player in the case of alternating updates can have a beneficial regularization effect. Strengthening the self terms – the terms which minimize the player's own norm – does not lead to a substantial improvement; this is somewhat expected since while the modified ODEs give us the exact coefficients required to *cancel* the drift, they do not tell us how to strengthen it, and our choice of exact coefficients from the drift might not be optimal.



(a) Top 10% models.

(b) Top 20% models.

(c) Top 30% models.

Figure A.19. Performance across the top performing models for vanilla SGD and with canceling the interaction terms and only cancelling the discriminator interaction terms. We notice that canceling only the discriminator interaction terms can result in higher performance across more models, and that the interaction term of the generator can play a positive role, likely due to the smaller strength of the generator interaction term compared to simultaneous updates.

I. Experimental details

I.1. Classification experiments

For the MNIST classification results with alternating updates, we use an MLP with layers of size $[100, 100, 100, 10]$ and a learning rate of 0.08. The batch size used is 50 and models are trained for 1000 iterations. Error bars are obtained from 5 different seeds.

I.2. GAN experiments

SGD results: All results are obtained using sweep over $\{0.01, 0.005, 0.001, 0.0005\}$ for the discriminator and for the generator learning rates. We restrict the ratio between the two learning rates to be in the interval $[0.1, 10]$ to ensure the validity of our approximations. All experiments use a batch size of 128.

Learning rate ratios: In the learning rate ratios experiments, we control for the number of experiments which have the same learning rate ratio. To do so, we obtain 5 learning rates uniformly sampled from the interval $[0.001, 0.01]$ which we use for the discriminator, we fix the learning rate ratios to be in $\{0.1, 0.2, 0.5, 1., 2., 5.\}$ and we obtain the generator learning rate from the discriminator learning rate and the learning rate ratio.

Adam results: For Adam, we use the learning rate sweep $\{10^{-4}, 2 \times 10^{-4}, 3 \times 10^{-4}, 4 \times 10^{-4}\}$ for the discriminator and the same for the generator, with $\beta_1 = 0.5$ and $\beta_2 = 0.99$.

Explicit regularization coefficients: For all the experiments where we cancel the interaction terms of the drift, we use the coefficients given by DD. For SGA and consensus optimization, we do a sweep over coefficients in $\{0.01, 0.001, 0.0001\}$.

Model Architectures: All GAN experiments use the CIFAR-10 convolutional SN-GAN architectures – see Table 3 in Section B.4 in Miyato et al. (2018).

Libraries: We use JAX (Bradbury et al., 2018) to implement our models, with Haiku (Hennigan et al., 2020) as the neural network library, and Optax (Hessel et al., 2020) for optimization.

Computer Architectures: All models are trained on NVIDIA V100 GPUs. Each model is trained on 4 devices.

I.3. Implementing explicit regularization

The loss functions we are interested are of the form

$$L = \mathbb{E}_{p(\mathbf{x})} f_{\boldsymbol{\theta}}(\mathbf{x})$$

We then have

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}_i} L &= \mathbb{E}_{p(\mathbf{x})} \nabla_{\boldsymbol{\theta}_i} f_{\boldsymbol{\theta}}(\mathbf{x}) \\
\|\nabla_{\boldsymbol{\theta}} L\|^2 &= \sum_{i=1}^{i=|\boldsymbol{\theta}|} \left(\nabla_{\boldsymbol{\theta}_i} L\right)^2 = \sum_{i=1}^{i=|\boldsymbol{\theta}|} \left(\mathbb{E}_{p(\mathbf{x})} \nabla_{\boldsymbol{\theta}_i} f_{\boldsymbol{\theta}}(\mathbf{x})\right)^2
\end{aligned}$$

to obtain an unbiased estimate of the above using samples, we have that:

$$\begin{aligned}
\|\nabla_{\boldsymbol{\theta}} L\|^2 &= \sum_{i=1}^{i=|\boldsymbol{\theta}|} \left(\mathbb{E}_{p(\mathbf{x})} \nabla_{\boldsymbol{\theta}_i} f_{\boldsymbol{\theta}}(\mathbf{x})\right)^2 \\
&= \sum_{i=1}^{i=|\boldsymbol{\theta}|} (\mathbb{E}_{p(\mathbf{x})} \nabla_{\boldsymbol{\theta}_i} f_{\boldsymbol{\theta}}(\mathbf{x}))(\mathbb{E}_{p(\mathbf{x})} \nabla_{\boldsymbol{\theta}_i} f_{\boldsymbol{\theta}}(\mathbf{x})) \\
&\approx \sum_{i=1}^{i=|\boldsymbol{\theta}|} \left(\frac{1}{N} \sum_{k=1}^{N} \nabla_{\boldsymbol{\theta}_i} f_{\boldsymbol{\theta}}(\widehat{\mathbf{x}_{1,k}})\right) \left(\frac{1}{N} \sum_{j=1}^{N} \nabla_{\boldsymbol{\theta}_i} f_{\boldsymbol{\theta}}(\widehat{\mathbf{x}_{2,j}})\right)
\end{aligned}$$

so we have to use two sets of samples $\mathbf{x}_{1,k} \sim p(\mathbf{x})$ and $\mathbf{x}_{2,j} \sim p(\mathbf{x})$ from the true distribution (by splitting the batch into two or using a separate batch) to obtain the correct norm. To compute an estimator for $\nabla_{\boldsymbol{\phi}} \|\nabla_{\boldsymbol{\theta}} L\|^2$, we can compute the gradient of the above unbiased estimator of $\|\nabla_{\boldsymbol{\theta}} L\|^2$. However, to avoid computing gradients for two sets of samples, we derive another unbiased gradient estimator, which we use in all our experiments:

$$\frac{2}{N} \sum_{i=1}^{i=|\boldsymbol{\theta}|} \sum_{k=1}^{N} \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}_i} f_{\boldsymbol{\theta}}(\widehat{\mathbf{x}_{1,j}}) \nabla_{\boldsymbol{\theta}_i} f_{\boldsymbol{\theta}}(\widehat{\mathbf{x}_{2,j}}) \tag{A.68}$$

References

Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., and Graepel, T. The mechanics of n-player differentiable games. In *International Conference on Machine Learning*, pp. 354–363. PMLR, 2018.

Barrett, D. G. and Dherin, B. Implicit gradient regularization. In *International Conference on Learning Representations*, 2021.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Hairer, E. and Lubich, C. The life-span of backward error analysis for numerical integrators. *Numerische Mathematik*, 76: 441–457, 06 1997.

Hairer, E., Lubich, C., and Wanner, G. *Geometric numerical integration*, volume 31. Springer-Verlag, Berlin, second edition, 2006. ISBN 3-540-30663-3; 978-3-540-30663-4.

Hennigan, T., Cai, T., Norman, T., and Babuschkin, I. Haiku: Sonnet for JAX, 2020. URL http://github.com/deepmind/dm-haiku.

Hessel, M., Budden, D., Viola, F., Rosca, M., Sezener, E., and Hennigan, T. Optax: composable gradient transformation and optimisation, in jax!, 2020. URL http://github.com/deepmind/optax.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802, 2017.

Mescheder, L., Nowozin, S., and Geiger, A. The numerics of gans. In *Advances in Neural Information Processing Systems*, pp. 1825–1835, 2017.

Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for gans do actually converge? In *International conference on machine learning*, pp. 3481–3490. PMLR, 2018.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Nagarajan, V. and Kolter, J. Z. Gradient descent gan optimization is locally stable. In *Advances in neural information processing systems*, pp. 5585–5595, 2017.

Qin, C., Wu, Y., Springenberg, J. T., Brock, A., Donahue, J., Lillicrap, T. P., and Kohli, P. Training generative adversarial networks by solving ordinary differential equations. 2020.