Differentiable Games in the Era of Machine Learning

Gauthier Gidel
Mila and DIRO
Differentiable Games in the Era of Machine Learning
This talk
New Formulations for learning

Understanding the Learning in Differentiable Games
Adversarial Example Games


Mila, McGill University, Université de Montréal, Facebook AI Research
Standard Adversarial Attack Setting:

\[ x' \in \text{argmax}_{x' \in \mathcal{X}} \ell(f(x'), y), \quad \text{s.t.} \quad d(x, x') \leq \epsilon. \]

- \( f \): function to attack.
- \( x \in \mathcal{X} \): input datapoint.
- \( x' \in \mathcal{X} \): adversarial example.
- \( y \in \mathcal{Y} \): true label.
- \( \ell \): loss function.

We Need to know the function to optimize.

Usually \( L_p \) norm.
Standard Adversarial Attack Setting:

\[ x' \in \arg\max_{x' \in x} \ell(f(x'), y), \quad \text{s.t.} \quad d(x, x') \leq \epsilon. \]

- **\( f \)**: function to attack.
- **\( \ell \)**: loss function.
- **Threat model**: what we assume to have access to. (e.g. gradients, softmax values)
- **Optimization Problem**: We Need to know the function to optimize.

**Threat model**
- **Whitebox threat model**
- **Blackbox threat model**

**Usually** \( L_p \) norm.
Intuitions

- Adversarial examples are **features**. [Ilyas et al. 2019]
- Adversarial examples **always exist** with Neural Nets [Bubeck, Cherapanamjeri, Gidel, Tachet des Combes 2021] [Daniely and Schacham 2020]

- These features can be learned.
- Modifying them can attack a whole class $\mathcal{F}$ function.

**Conclusion:** the generator can learn to detect and change these features **without querying** $f_t \Rightarrow \text{NoBox attack.}$
A Realistic (and challenging) threat model: Non-interactive blackBox (NoBox) threat model

- **Target model** $f_t$ : we want to break that model.
- **Target examples** $\mathcal{D}$ : the data we want to corrupt.
- **Model hypothesis class** $\mathcal{F}$ : our knowledge on the target model. **New!**
- **Representative classifier** $f_c$ : we assume we can optimize over the hypothesis class using that representative classifier. **New!**
- **A Reference Dataset** $\mathcal{D}_{ref}$ : similar to the training set of $f_t$ **New!**

IDEA: Optimize over $\mathcal{F}$ to get adversarial examples that can attack any function in $\mathcal{F}$
Adversarial Example Games Framework

Game Between:

- A generator that generate adversarial examples conditioned on (x,y):

\[(x', y) \sim p_g \iff x' = g(x, y, z), (x, y) \sim D, z \sim p_z \text{ with } d(x', x) \leq \epsilon.\]

- A Classifier \(f_c\) that aims at getting robust against adversarial examples:

Classification loss of an adversarial example of (x,y):

\[\ell(f_c(g(x, y, z))), y)\]
Adversarial Example Games Framework

Game Between:

- A generator that generates adversarial examples conditioned on \((x,y)\):

\[
(x', y) \sim p_g \Leftrightarrow x' = g(x, y, z), \quad (x, y) \sim \mathcal{D}, \quad z \sim p_z \quad \text{with} \quad d(x', x) \leq \epsilon.
\]

- A Classifier \(f_c\) that aims at getting robust against adversarial examples:

\[
\max_{g \in G_\epsilon} \min_{f_c \in \mathcal{F}} \mathbb{E}_{(x,y) \sim \mathcal{D}, z \sim p_z} [\ell(f_c(g(x, y, z)), y)] =: \varphi(f_c, p_g)
\]
# Attacking in the Wild: CIFAR 10

<table>
<thead>
<tr>
<th>Source</th>
<th>Attack</th>
<th>VGG-16</th>
<th>RN-18</th>
<th>WR</th>
<th>DN-121</th>
<th>Inc-V3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Clean</td>
<td>11.2 ± 0.9</td>
<td>13.1 ± 2.0</td>
<td>6.8 ± 0.7</td>
<td>11.2 ± 1.4</td>
<td>9.9 ± 1.3</td>
</tr>
<tr>
<td></td>
<td>MI-Attack</td>
<td>63.9 ± 1.3</td>
<td>74.6 ± 0.4</td>
<td>63.1 ± 1.2</td>
<td>72.5 ± 1.3</td>
<td>67.9 ± 1.6</td>
</tr>
<tr>
<td></td>
<td>DI-Attack</td>
<td>77.4 ± 1.7</td>
<td>90.2 ± 0.8</td>
<td>74.0 ± 1.0</td>
<td>87.1 ± 1.3</td>
<td>85.8 ± 0.8</td>
</tr>
<tr>
<td></td>
<td>TID-Attack</td>
<td>21.6 ± 1.3</td>
<td>26.5 ± 2.2</td>
<td>14.0 ± 1.5</td>
<td>22.3 ± 1.6</td>
<td>19.8 ± 0.9</td>
</tr>
<tr>
<td></td>
<td>SGM-Attack</td>
<td>68.4 ± 1.8</td>
<td>79.5 ± 0.5</td>
<td>64.3 ± 1.6</td>
<td>73.8 ± 1.0</td>
<td>70.6 ± 1.7</td>
</tr>
<tr>
<td></td>
<td>AEG (Ours)</td>
<td>89.0 ± 2.1</td>
<td>96.8 ± 0.7</td>
<td>80.9 ± 2.4</td>
<td>91.6 ± 1.7</td>
<td>87.2 ± 1.6</td>
</tr>
</tbody>
</table>

|        | MI-Attack | 54.3 ± 1.1 | 62.5 ± 0.9 | 56.3 ± 1.3 | 66.1 ± 1.5 | 65.0 ± 1.3 |
|        | DI-Attack | 61.1 ± 1.9 | 69.1 ± 0.8 | 61.9 ± 1.1 | 77.1 ± 1.2 | 71.6 ± 1.6 |
|        | TID-Attack | 21.7 ± 1.2 | 23.8 ± 1.5 | 14.0 ± 1.4 | 21.7 ± 1.1 | 19.3 ± 1.2 |
|        | SGM-Attack | 51.6 ± 0.7 | 60.2 ± 1.3 | 52.6 ± 0.9 | 64.7 ± 1.6 | 61.4 ± 1.3 |
|        | AEG (Ours) | 90.5 ± 1.6 | 95.9 ± 1.4 | 80.3 ± 2.3 | 95.9 ± 1.4 | 90.6 ± 2.4 |

|        | MI-Attack | 49.9 ± 0.1 | 50.0 ± 0.2 | 46.7 ± 0.4 | 50.4 ± 0.6 | 50.0 ± 0.3 |
|        | DI-Attack | 65.1 ± 0.1 | 64.5 ± 0.2 | 58.8 ± 0.6 | 64.1 ± 0.3 | 60.9 ± 0.6 |
|        | TID-Attack | 26.2 ± 0.6 | 24.0 ± 0.6 | 13.0 ± 0.2 | 20.8 ± 0.7 | 18.8 ± 0.2 |
|        | AEG (Ours) | 94.2 ± 1.2 | 93.7 ± 1.6 | 77.1 ± 1.1 | 92.3 ± 1.7 | 86.5 ± 1.3 |

Table 2: Error rates on $\mathcal{D}$ for average NoBox architecture transfer attacks with $\epsilon = 0.03125$
Real World Games look like Spinning Tops
Real World Game

A competitive, two-player, symmetric zero-sum game, designed for human enjoyment, engagement and as a mean of challenging each others strategic thinking.
Extensive Form Game / Game Tree

Normal Form Game Payoff

Outcome $f(\square, \square) = +1$
Game of Tic Tac Toe has more than $10^{567}$ behaviourally distinct pure strategies.
Definition 3. Nash clustering $C$ of the finite zero-sum symmetric game strategy $\Pi$ set by setting for each $i \geq 1$: $N_{i+1} = \text{supp}(\text{Nash}(P|\Pi \setminus \bigcup_{j\leq i} N_j))$ for $N_0 = \emptyset$ and $C = (N_j : j \in \mathbb{N} \land N_j \neq \emptyset)$. 
Definition 3. Nash clustering $C$ of the finite zero-sum symmetric game strategy $\Pi$ set by setting for each $i \geq 1$: $N_{i+1} = \text{supp}(\text{Nash}(P|\Pi \setminus \bigcup_{j \leq i} N_j))$ for $N_0 = \emptyset$ and $C = (N_j : j \in \mathbb{N} \land N_j \neq \emptyset)$.

Theorem 2. Nash clustering satisfies $RPP(C_i, C_j) \geq 0$ for each $j > i$. 
Empirical Verification
OpenSpiel [LINK]
Game of Tic Tac Toe has more than $10^{567}$ behaviourally distinct pure strategies.

We rely on empirical game theory through sampling.

An open question: can the analysis be done implicitly through the game tree traversal?
Nash clustering + RPP creates transitive structure (Theorem 2)

Sizes of Nash clusters denote "non-transitivity" at each level
Conclusion:

Empirical and Theoretical evidence that in real world game:

● Huge number of strategies.
● But tiny number of **Good** strategies
● Spinning top shape.

(The worst you get the more strategies there is)
Thank you!