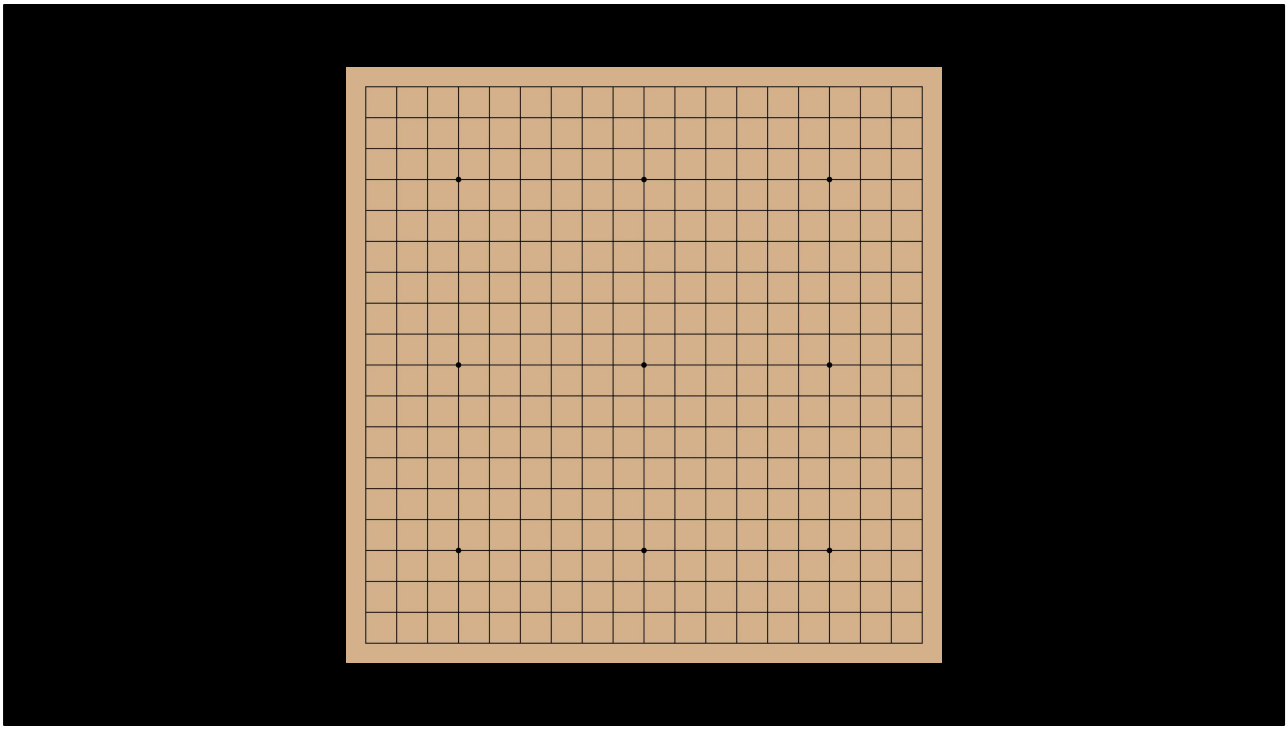
Adversarial Policies Beat Superhuman Go AIs

Tony Wang* Adam Gleave* Tom Tseng Kellin Pelrine
Nora Belrose Joseph Miller Michael Dennis Yawen Duan
Viktor Pogrebniak Sergey Levine Stuart Russell

twang6@mit.edu

goattack.far.ai

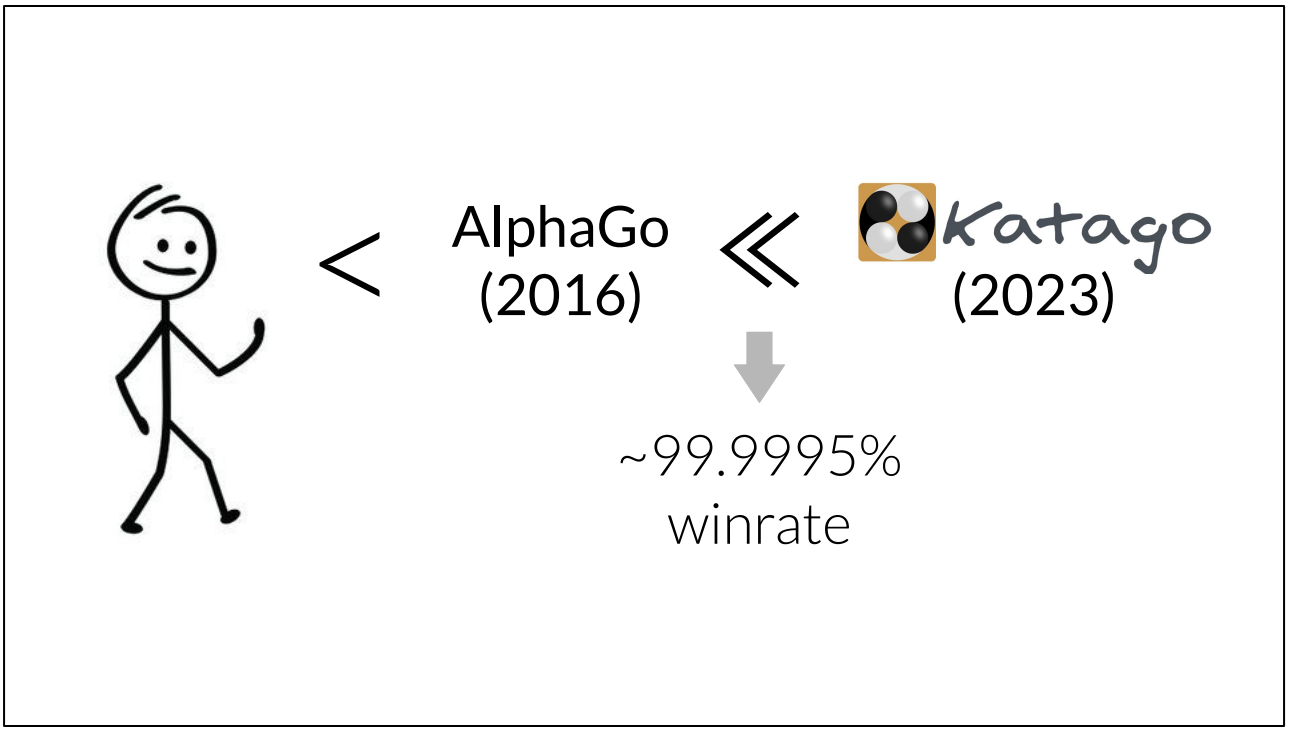
Hi, my name is Tony and I want to teach you about how adversarial policies beat superhuman Go AIs.
[click]



Go is an ancient Chinese board game invented over 2500 years ago. Two players take turns placing black and white stones on a square board, trying to surround territory, and kill each other’s stones. At the end of the game, whoever controls more of the board wins.
[click]

AlphaGo def. Lee Sedol (4-1)

The game you just saw was part of a match between Lee Sedol and the AI AlphaGo. Lee (on the right here) was one of the strongest humans to ever play the game. But, by leveraging deep neural networks, AlphaGo was able to pull off an upset, and beat Lee 4-1.
[click]



AlphaGo showed, that deep neural networks could be better at Go than any human. But progress did not stop here and even more powerful systems were eventually developed. [click] For example KataGo, which is currently the strongest open-source Go AI, can beat AlphaGo roughly 99.9995% of the time. Systems like KataGo show that at least in certain domains, AI systems can far surpass human capabilities.

Okay, so that's some background on Go and Go AIs. [click]

Sources:

1. AlphaGo vs. AlphaZero (https://www.nature.com/articles/nature24270), 5185 - 3739 = 1446 => 99.97% winrate
2. From our paper, AlphaZero_s800 has 3813 elo of goratings.org. cp505_s1 has 2738 elo on goratings.org. cp505_s800 has 4500 elo on goratings.org. So 700 elo gap => 98% winrate.
3. From reddit.com/r/baduk/comments/hma3nx/unified_elo_rating_for_ais/, AlphaZero is 2065 - 1330 = 735 elo weaker that cp505.

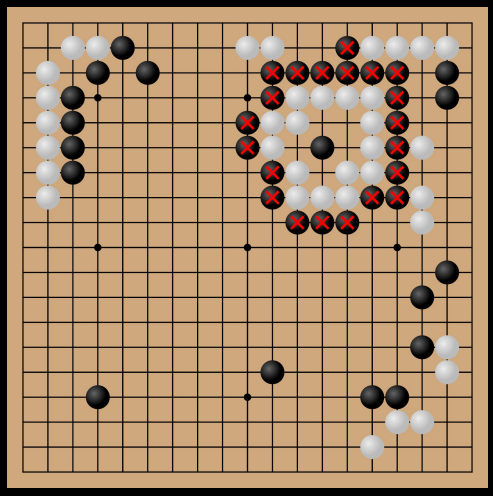
Do current superhuman AI systems have good **worst-case** performance?

No, for Go AIs.

The key question we tried to answer with our project is this, do current superhuman AI systems have good worst-case performance? The reason we care about this, is that if we deploy AI systems in high stakes settings (for example, as self-driving cars, automated traders, or autonomous weapons systems), it doesn't matter if our systems are superhuman most of the time, if they can still fail catastrophically some of the time.

We attack Go AIs to get at this question. And, at least in this domain, our answer is no. Let me show you why.

The Cyclic-Exploit

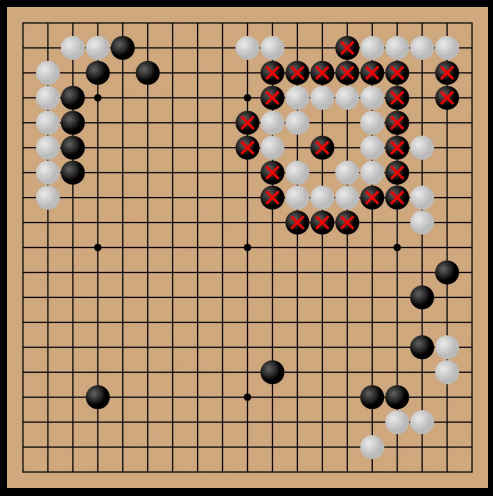


KataGo AI

Adversary

Existing "superhuman" Go AIs have a hidden weakness—they don't understand cyclic shapes. If you get the AI to make a cyclic-shape, it thinks the shape is invulnerable and won't defend it even though it can be killed. Here's the superhuman KataGo making a circle as black. Because KataGo doesn't realize its circle can be killed, an adversary AI we trained can slowly smother the circle from the inside and outside. Let's see this action.
[click]

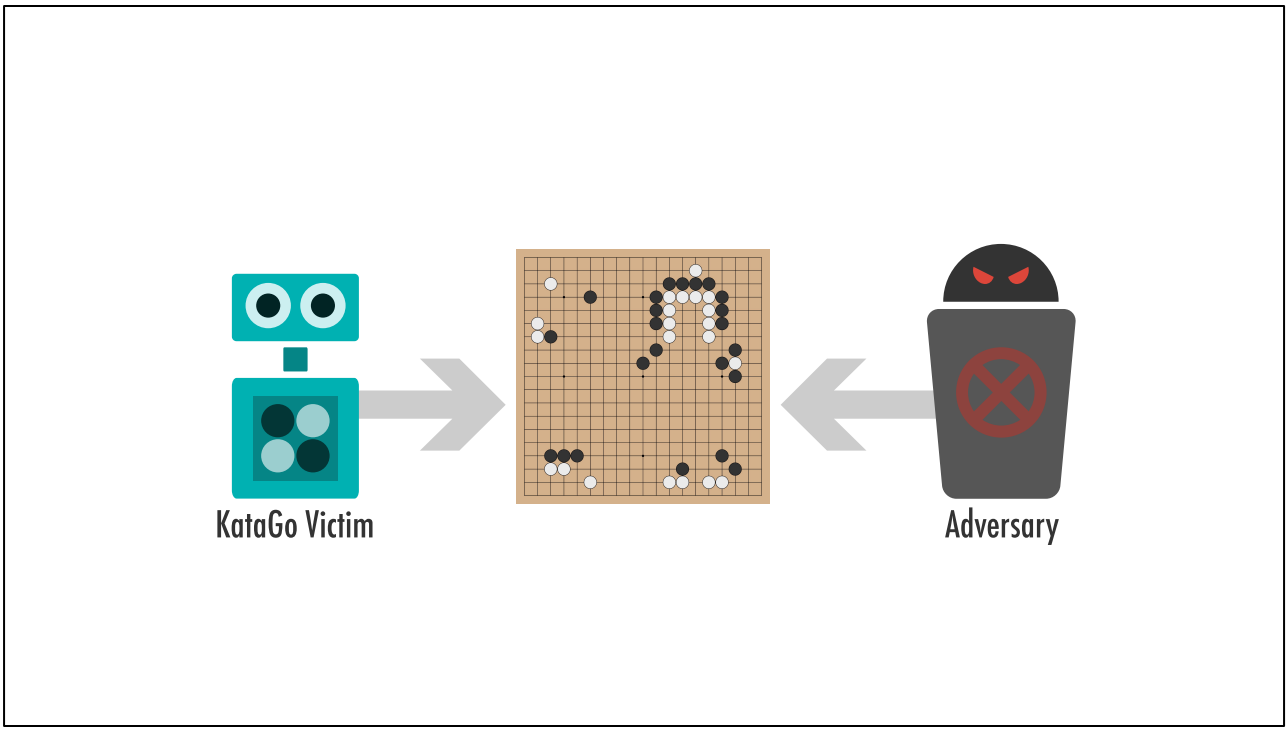
The Cyclic-Exploit



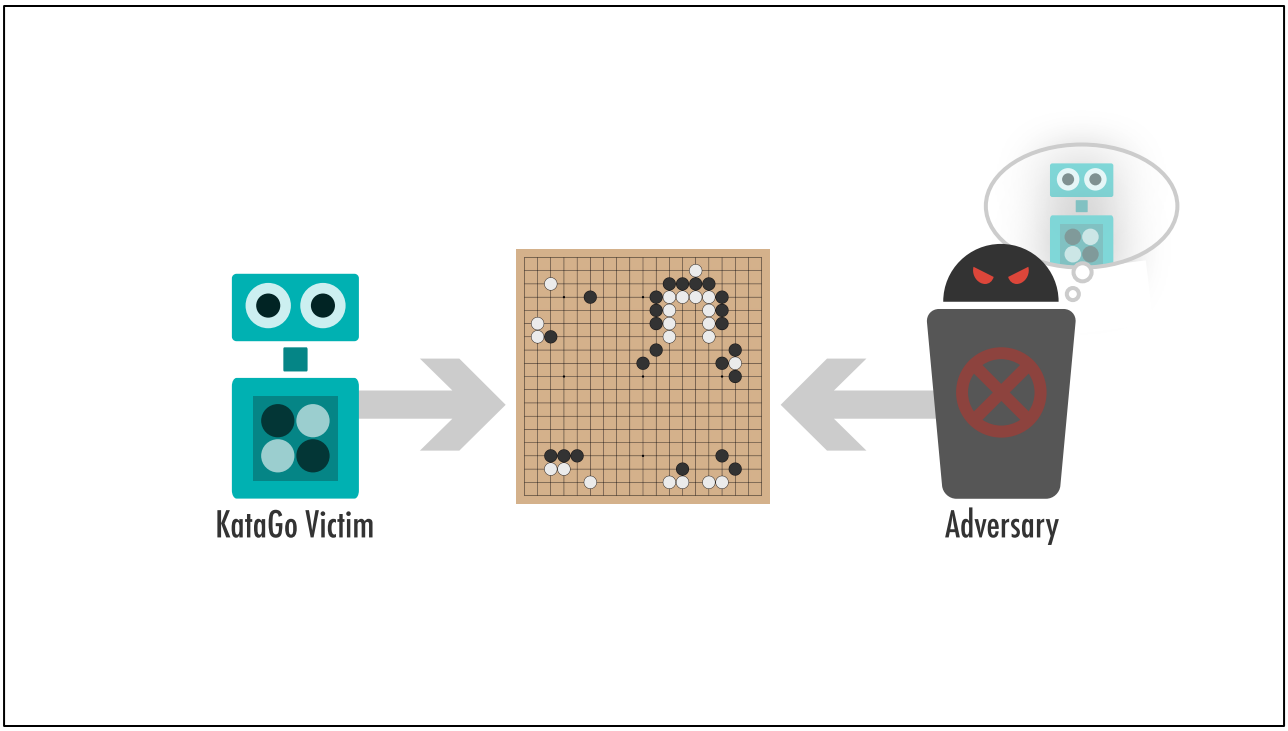
KataGo AI

Adversary

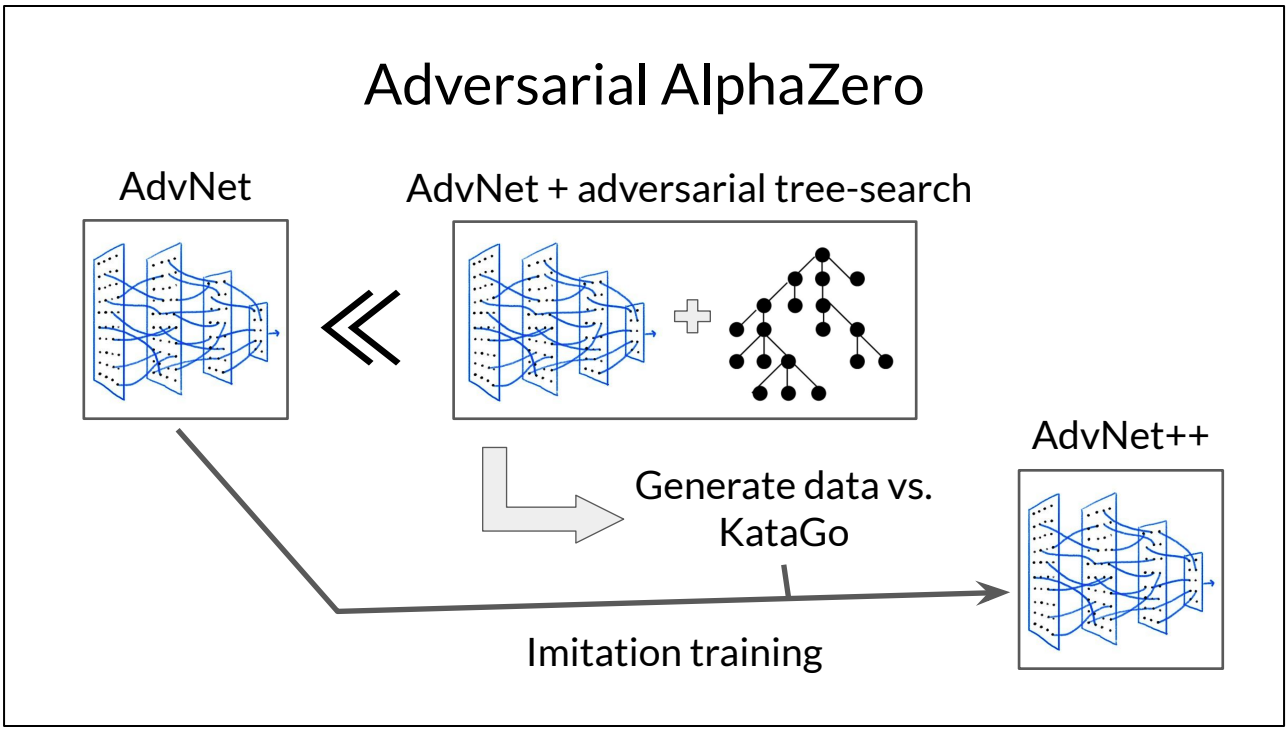
So all of the black stones marked with an X here will eventually die. And even though this is an adversary AI playing right now, this strategy is actually simple enough for a human to pull it off manually.
[click]



How did we discover this exploit? Well, we did so by training an adversary AI to defeat KataGo. [click]

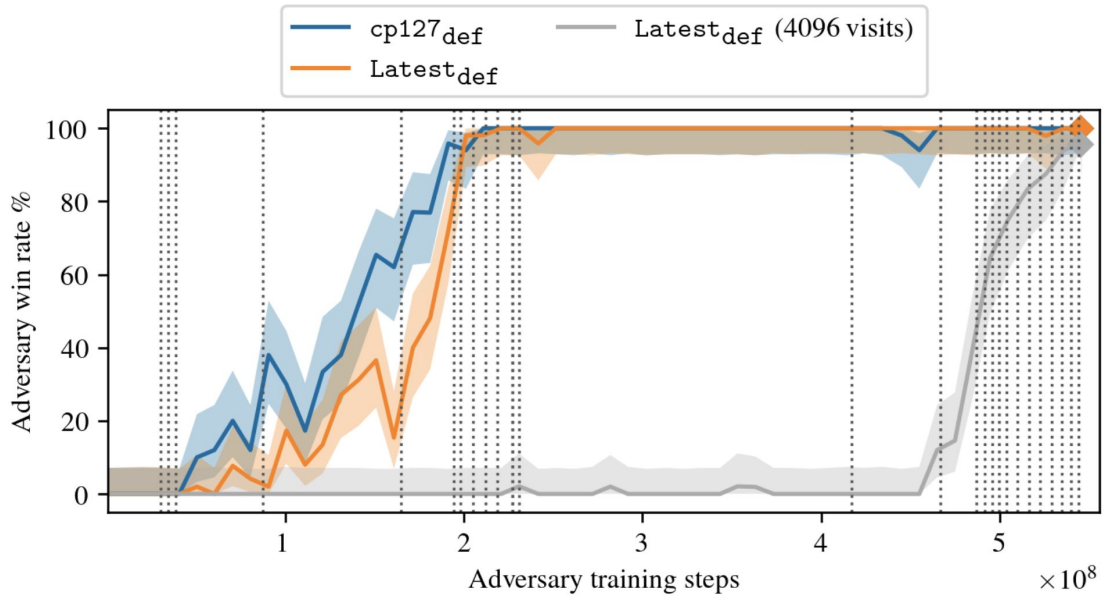


Our adversary has a special ability. Namely, it can simulate the victim's behavior when it searches over future moves.



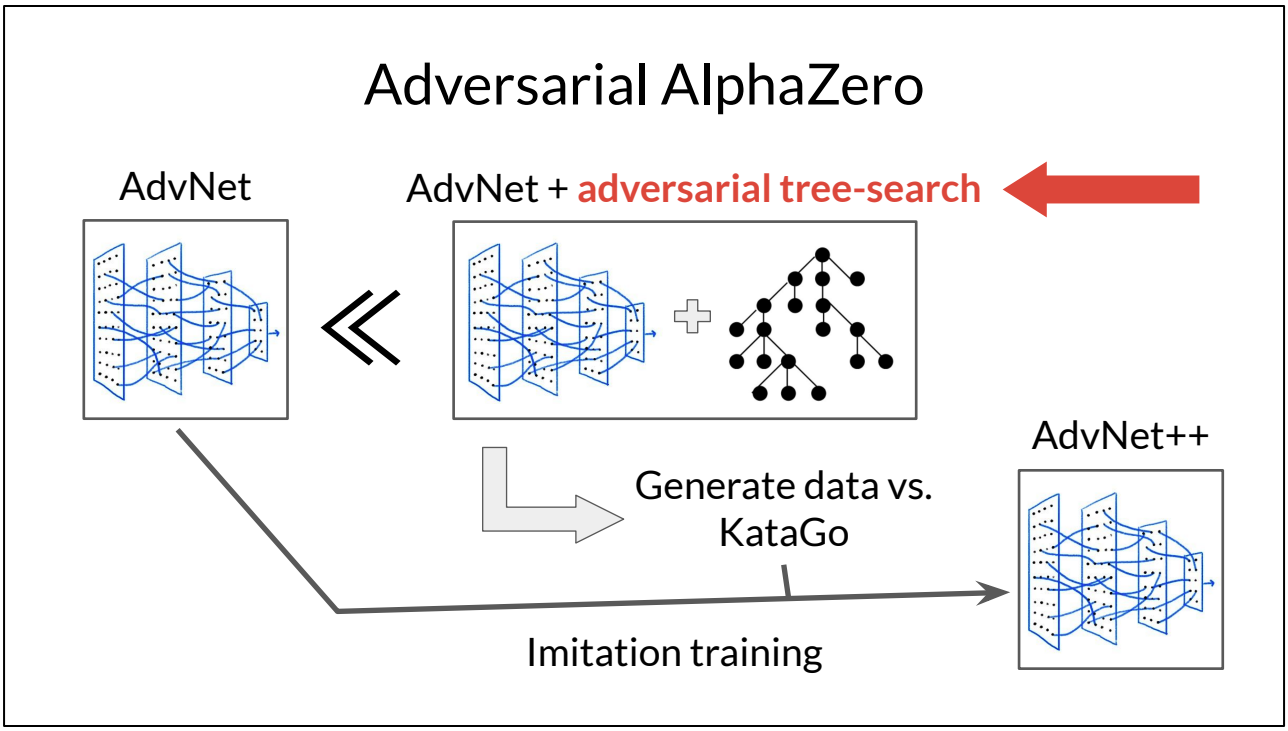
To train our adversary, we use an adversarial variant of the AlphaZero algorithm. Here's how it works. [click] We start with a randomly initialized adversary neural network [click] Next, we augment this network with an adversarial variant of Monte-Carlo Tree Search. Tree-search is a policy improvement operator, meaning the network with tree search is a stronger adversary than the network alone. This adversarial tree-search is also where the adversary simulates possible victim responses. [click] We then pit our search-augmented adversary against KataGo, generating a dataset of behavior. [click] Finally, we train the adversary network to mimic the behavior of the search-augmented adversary. This imitation training yields a slightly stronger network.

Repeating this process, we eventually get an adversary that is able to reliably defeat KataGo via the cyclic-exploit I showed previously.

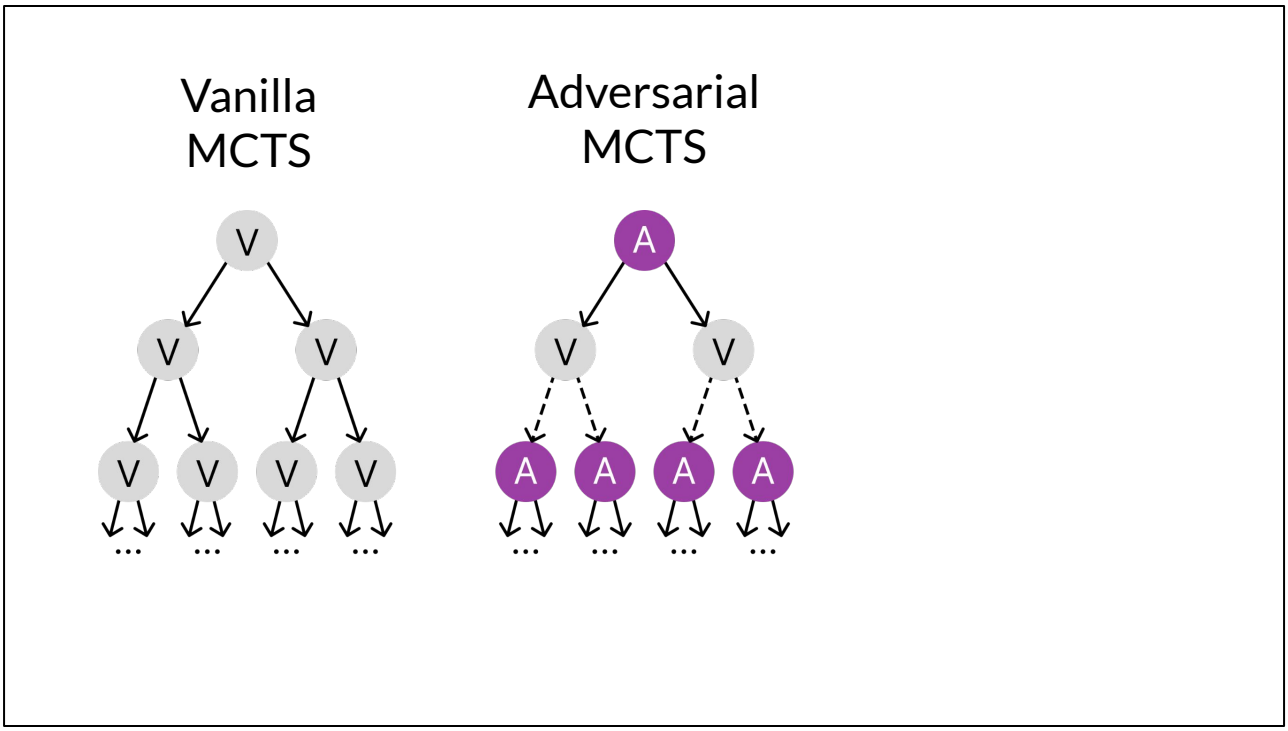


Here's what running our adversarial alphazero algorithm looks like.
On the horizontal axis, we have the number of gradient descent steps performed on our adversary.
The different colored lines represent the win-rate of our adversary against different versions of KataGo of varying strength.
We see that over the course of our training, our adversary goes from losing to winning.
And to give you a sense of scale, the entire x-axis encompasses roughly 2000 V100 GPU days.

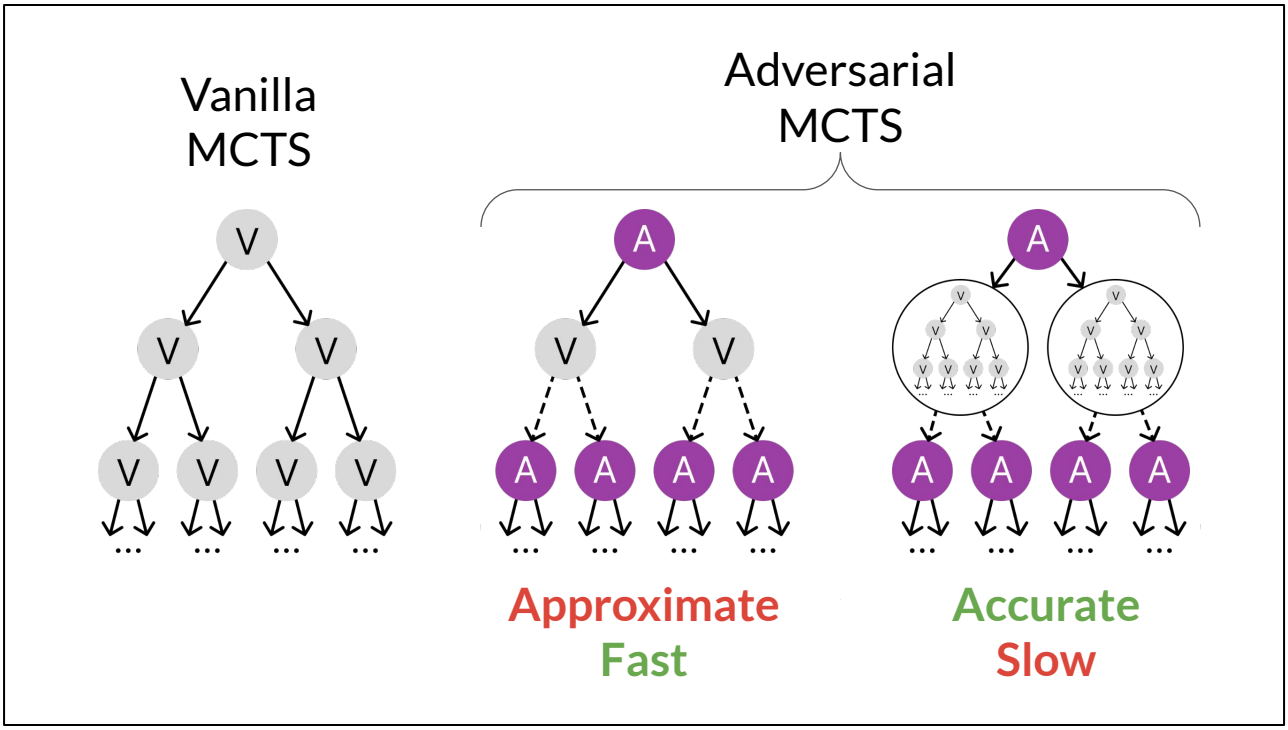
The vertical dashed lines here actually denote a curriculum, which is an important detail.
We couldn't just train our adversary against the strongest version of KataGo right off the bat,
because it would lose every game and we would have no training signal.
So instead, we started our training against really weak versions of KataGo,
and only once we achieve a sufficiently high win-rate, do we move on to a slightly stronger adversary.
~~We were able to do this in part because KataGo released a full training checkpoint history of its networks, ranging from very weak to very strong.~~



Okay now I want to loop back to our adversarial AlphaZero training procedure, and give you a bit more detail on how we do our adversarial tree search.



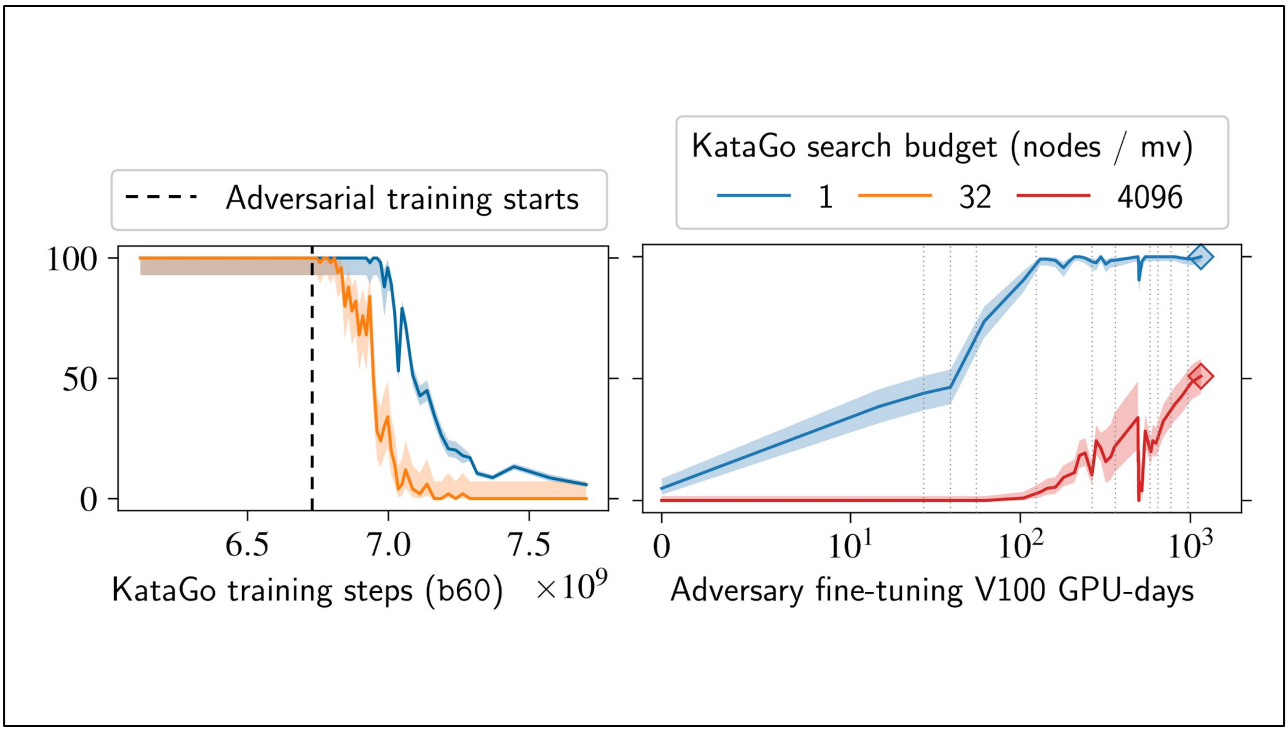
So in regular Monte-Carlo tree search, on the left here, moves are sampled from a single *policy network*, and AI pretends that it is its own opponent. We developed an adversarial version of Monte-Carlo tree search, on the right, where the adversary simulates the opponent when it is the opponent's turn.



We actually have two variants of our adversarial MCTS.

The first variant is an approximate but fast version: we approximate the victim with just its neural network policy.
This is an approximation, because in reality the victim is also doing tree search.

The second variant is accurate but slow: we actually recursively simulate the victim tree search.
This is much more accurate and actually results in a stronger adversary, but is computationally impractical.



Okay, so finally, I want to talk a little bit about how to defend against our adversarial exploit.

After publishing an earlier version of our work late last year, we talked with the creator of KataGo, who actually started to do some adversarial training on KataGo to make it immune to our exploit. And as we see on the graph here, after adversarial training starts, KataGO gradually became immune to our adversary.

[click[
However, we show that this defense is incomplete. We took the defended version of KataGo, and finetuned our adversaries to attack to updated version of KataGo. And on the right here, we see that our adversary once again learns to defeat KataGo, and we checked that it did so actually via the same cyclic exploit.
So defense is still an open problem.

published adversaries. However, we show this defense is incomplete—re-attacking KataGo yields adversaries that are still able to win via the cyclic exploit. So defense is still an open question.

“Superhuman” Go AIs have bad worst-case performance.

Automated red-teaming is a powerful technique.

In summary, we showed what were previously thought to be superhuman Go AIs, actually have pretty bad worst-case performance.
And discovered these vulnerabilities via automated red-teaming, that is, we trained another AI system to find the exploit.

Our main takeaway, is that automated red-teaming is pretty powerful technique. AlphaGo, AlphaZero, and their successors, have been known the world for 6+ years now. However, it wasn’t until we use automated red-teaming that we managed to discover the exploit I showed today. We think that powerful AI systems of the future may also have failure modes that are hard to discover without the assistance of AI.

Nir: Would go back to the big question. **The day is coming where humans will not be able to test systems rigorously.** We’re approaching superhuman systems. How much can we trust these systems? Should also add a summary of what we did. These points are too low level. **2 is the main point.** Maybe add on: we don’t know how to defend.

Website: goattack.far.ai

Paper: arxiv.org/abs/2211.00241

Poster tomorrow at 11am (#300)


Adam Gleave


Tom Tseng


Kellin Pelrine


Nora Belrose


Joseph Miller


Michael Dennis


Yawen Duan


Viktor Pogrebniak


Sergey Levine


Stuart Russell

This work was done in collaboration with all these other fantastic folks on the screen. In particular, Adam and Kellin are here at the conference as well, so feel free to to flag any of us down if you have questions. For more information, check out our website and paper, and do swing by our poster tomorrow (we're number 300).