Can Go AIs Be Adversarially Robust?*

Tom Tseng[1], **Euan McLean**[1], **Kellin Pelrine**[†1,2], **Tony T. Wang**[†3], **Adam Gleave**[†1]

[1]FAR.AI
[2]Mila
[3]Massachusetts Institute of Technology
{tom, kellin, adam}@far.ai, emc031@gmail.com, twang6@mit.edu

Abstract

Prior work found that superhuman Go AIs can be defeated by simple adversarial strategies, especially "cyclic" attacks. In this paper, we study whether adding natural countermeasures can achieve robustness in Go, a favorable domain for robustness since it benefits from incredible average-case capability and a narrow, innately adversarial setting. We test three defenses: adversarial training on hand-constructed positions, iterated adversarial training, and changing the network architecture. We find that though some of these defenses protect against previously discovered attacks, none withstand freshly trained adversaries. Furthermore, most of the reliably effective attacks these adversaries discover are different realizations of the same overall class of cyclic attacks. Our results suggest that building robust AI systems is challenging even with extremely superhuman systems in some of the most tractable settings, and highlight two key gaps: efficient generalization of defenses, and diversity in training.

Website — https://goattack.far.ai
Code — https://github.com/AlignmentResearch/go_attack

1 Introduction

It is essential that AI systems work robustly, especially when they are deployed at a societal scale or are used in safety-critical systems. Unfortunately, although the *average-case* performance of AI systems is rapidly improving, building AI systems with good *worst-case* performance remains an unsolved problem. Indeed, Go AIs (Wang et al. 2023a), image classifiers (Liu et al. 2023; Croce et al. 2021), and large language models (Mazeika et al. 2024) all fail catastrophically when presented with adversarial inputs.

However, Go seems like it ought to be especially favorable for robustness. First, unlike computer vision models and LLMs where average-case performance struggles to surpass humans, Go AIs' average-case play massively outstrips human ability (Silver et al. 2016; Wang et al. 2023a). Second, the attack surface is narrow, with only a limited set of valid moves available to the adversary each turn. Third, unlike many non-game domains, Go is inherently adversarial and zero-sum, and there is no fundamental trade-off between clean and robust accuracy (Tsipras et al. 2019). Furthermore, there is even a clear initial target: robustness to the "cyclic attack" found by Wang et al. (2023a). This class of attacks is well-defined and consistent enough for humans to understand it, replicate it, and easily tell novice players how to defend against it—suggesting it should be feasible for a superhuman AI to defend against it. The Go models Wang et al. (2023a) attacked were not designed with robustness in mind, giving us hope that with more effort they could be made robust.

So, in this paper, we study if it is possible to make superhuman agents robust—that is, to make them have good worst-case performance—by focusing on the domain of Go. Go has a proven track record of driving progress in AI, motivating the development of algorithms like AlphaZero (Silver et al. 2016) and MuZero (Schrittwieser et al. 2020). If we could solve robustness here, it would provide a natural starting point for designing robust AI systems more broadly. In particular, we aim for a Go AI that a) does not make mistakes that humans can easily correct and b) cannot be reliably defeated by adversaries trained with a small amount of compute. These criteria, elaborated in Section 2, reflect two critical types of robustness: assurance that a system will not open up new vulnerabilities that humans could avoid, and that it will not be cheaply exploitable.

Leveraging the state-of-the-art open-source Go AI KataGo (Wu 2021), we investigate three natural defenses (Figure 1). But we find that all three are ineffective— given query access to the defended systems, it is relatively cheap to train new adversaries that reliably defeat them and cause them to blunder in ways that humans would not. In fact, never mind all attacks—each defended system is still vulnerable to cyclic attacks, demonstrating the defenses' poor generalization even in the favorable domain of Go.

More specifically, the first defense we study is **positional adversarial training**, which augments KataGo's training data with examples of Wang et al. (2023a)'s cyclic attack (Section 3). This intervention produces a defended agent that never loses against Wang et al.'s adversary. Unfortunately, we find that adaptively fine-tuning Wang et al.'s adversary against the defended agent brings its win rate from 0% back

*See the full version (Tseng et al. 2024) of this paper for appendices, which also contain some referenced figures and tables.

†Equal advising contribution.

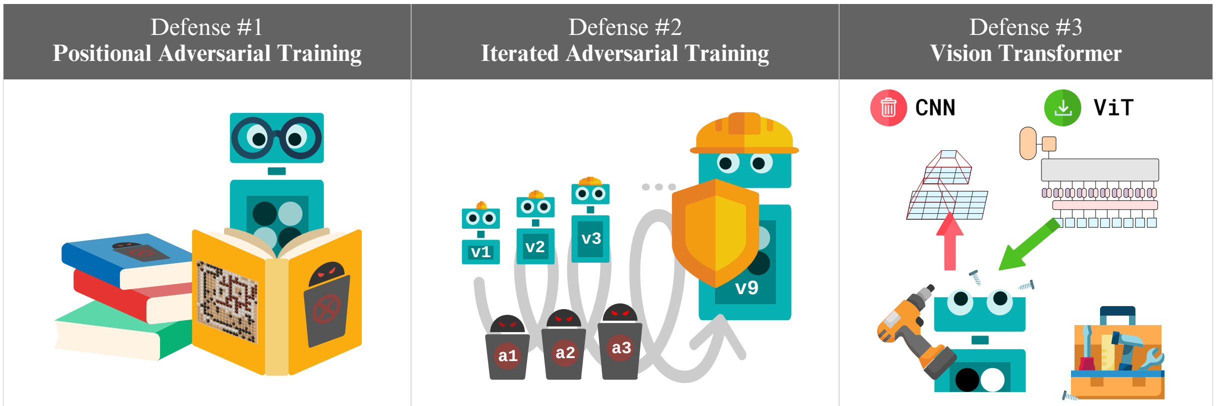


Figure 1: Three strategies for defending Go AIs against adversarial attack. **Left**: Positional adversarial training has an agent "study" adversarial positions by performing self-play starting from those positions. **Middle**: Iterated adversarial training consists of multiple rounds of an adversary finding attacks and a victim learning to defend. **Right**: We replace KataGo's convolutional neural network (CNN) backbone with a vision transformer (ViT) backbone to see which vulnerabilities of Go AIs are caused by the inductive biases of CNNs.

up to 92%. Moreover, this fine-tuned adversary wins using only a slight variant of its original strategy, and can be fine-tuned using just 8% of the compute used for the defense. (Appendix A lists the compute used for each attack and defense.) Furthermore, the defended agent is also vulnerable to a qualitatively new "gift attack" (Figure 3a).

Though ultimately unsuccessful, this first defense shows that defending against a *fixed* attack is possible. This motivates our second defense, **iterated adversarial training**, which simulates an "arms race" between an adversary continuously searching for new attacks and a victim continuously building defenses against those attacks (Section 4). Regrettably, we find that this scheme has the same weakness as positional adversarial training. The defended agent is robust to Wang et al. (2023a)'s original cyclic attack, but through fine-tuning we find a variant of the cyclic attack that defeats the defended agent 90% of the time using 26% of the total compute used to train the defended agent (Figure 3c). Another variant wins 81% of the time using 5% of the compute, though it loses efficacy when the defended agent uses more search (Figure 20).

The final defense we test is replacing the convolutional neural network (CNN) backbone used by KataGo with a **vision transformer** (ViT) backbone (Section 5). Is the cyclic vulnerability caused by poor inductive biases of CNNs? We disprove that hypothesis by training the world's first professional-level ViT-based Go AI and show that it too is beaten by cyclic attacks. Given Wang et al. (2023a) found the cyclic vulnerability was widespread across different Go AIs, with CNNs ruled out as the cause, a likely remaining candidate is the self-play training process. Playing only one opponent identical in strength and reasoning process may discourage exploration and get the model stuck in local equilibria, learning a way to assess vulnerability of groups that fails catastrophically on cyclic ones.

Our negative results in Go are evidence that achieving neural net robustness in other domains will require concerted effort—we cannot assume that merely boosting average-case performance to superhuman levels and adding simple defenses will be enough to solve robustness. State-of-the-art Go agents fail to represent a feature (cyclic shapes) that amateur humans easily learn, even when using methods such as adversarial training that provide numerous training examples where this feature is critical to solving the task correctly. This failure makes them easily exploitable, even by humans (Appendix J). Building robust Go AIs may be possible but will likely require significant design changes (Section 7).

2 Threat Model, Robustness, Attack Method

Threat Model We follow the threat model of Wang et al. (2023a) set in a two-player zero-sum Markov game (Shapley 1953). A threat actor trains an *adversary* agent to win against a *victim* agent. The threat actor has *gray-box access* to the victim: they can run inference on the victim's policy network on arbitrary inputs. However, the adversary does not have direct access to the victim's weights and cannot take gradients through the victim.

Defining Robustness We aim to make agents that are robust. Unlike settings like ϵ-ball-robust image classification (Croce et al. 2021), it is not immediately obvious what it means for a Go agent to be robust. In this work, we introduce and use three complementary definitions of robustness centered around the notion of being minimally exploitable by adversaries.

Firstly, we aim to make our agents *human robust*, meaning that they cannot be made to commit game-losing blunders that a human would not commit (Appendix B.1). Secondly, we aim to make our agents have high *training-compute robustness*, meaning that it should take a large amount of compute to train an adversary that can defeat a victim (Appendix B.2). Finally, and more speculatively, we aim to make our agents have high *inference-compute* *robustness*, meaning that our agents should be able to efficiently overcome vulnerabilities by using additional compute at inference-time (Appendix B.3). We pick these definitions of robustness because they are applicable to both Go policies and more general AI agents.

Attack Method The victims are based on KataGo (Wu 2020), the strongest open-source Go AI system. Like AlphaZero (Silver et al. 2018), it trains with self-play and selects moves using Monte Carlo tree search (MCTS; the amount of search is quantified by *visits*) combined with a neural net.

We use Wang et al. (2023a)'s state-of-the-art attack method to produce adversaries for adversarial training and testing defenses. Wang et al. train an adversary with *victim-play* where the adversary plays games against a frozen copy of the victim, and training data is saved only from the adversary's moves. The adversary selects moves using Adversarial MCTS (A-MCTS), a modification of MCTS that queries the victim's network when traversing MCTS nodes corresponding to the opponent's move. We follow Wang et al. by evaluating adversaries with 600 A-MCTS visits per move.

We typically train our adversaries by initializing them to `base-adversary` (Wang et al.'s original cyclic adversary), which achieved a 97% win rate against a 2022 KataGo network `base-victim` (which Wang et al. calls `Latest`) at 4096 victim visits. To find diverse attacks, we initialize some experiments to the non-cyclic `base-adv-early`, which is the first `base-adversary` checkpoint that beats `base-victim` at 1 victim visit. See Appendices A and C for details on these networks and training parameters.

3 Positional Adversarial Training

KataGo's official training run performed adversarial training on board positions exhibiting the cyclic attack. We show that KataGo's adversarially trained networks remain exploitable despite this by training new adversaries that beat the strongest KataGo network from the end of 2023, which we call `dec23-victim`, and a stronger model `may24-victim` from May 2024. One adversary, `gift-adversary`, defeats `dec23-victim` in 91% of games (at 512 victim visits) using a new non-cyclic exploit where the victim repeatedly gifts the adversary two stones, though it does not scale to high victim visits as well as cyclic attacks. We then find that `dec23-victim` and `may24-victim` are both vulnerable to cyclic attacks, with another adversary `big-adversary` winning 56% of games against `may24-victim` (65,536 victim visits). The attacks can be replicated by a human expert (Appendix J).

3.1 Defense Methodology

We target models from KataGo's main training run, which began to include adversarial training against cyclic positions after the discovery of `base-adversary`. Since December 2022, 0.08% of KataGo's self-play games have been initialized from a set of hand-written positions based on `base-adversary`'s strategy (Wu 2022a, 2023b). Other positions were added as online Go players found different configurations of cyclic positions, growing the fraction of seeded self-play games to a few tenths of a percent (Wu 2023a). The resulting models defended well against `base-adversary`.

Despite this positive result, Wang et al. (2023a) were able to fine-tune `base-adversary` to produce `may23-adversary` achieving a 47% win rate against an adversarially trained KataGo checkpoint `may23-victim` at 4096 visits. Building on Wang et al.'s evaluation, we test `dec23-victim` and `may24-victim` which have had over twice as much adversarial training as `may23-victim`.

3.2 The Gift Adversary

The `gift-adversary` was initialized from the early `base-adv-early` checkpoint, encouraging exploration, and fine-tuned using victim-play against `dec23-victim`. Appendix D.1 gives more training details. The attack wins 91% of games against `dec23-victim` (at 512 victim visits—above the superhuman threshold of 64 visits, see Appendix I) after training with just 6% as much compute as the victim. The `gift-adversary` does not scale to high victim visits as well as the cyclic adversaries from Section 3.3 (Figure 2). However, the attack reveals a non-cyclic, qualitatively new exploit against KataGo (Figure 3a).

In particular, the adversary sets up a "sending-two-returning-one" situation where the victim unnecessarily gifts the adversary two stones and needs to capture one back. However, the victim's recapture is blocked by positional superko rules.[1] The adversary sets up the position such that it resurrects of one of its dead groups, leading to a disaster for the victim. The victim's gift is strange given that the victim was trained with superko rules and has a neural net input feature marking moves that are illegal due to superko, and moreover the scenario would not benefit the victim even if superko rules were not in play.

3.3 Cyclic Attacks

We also find that `dec23-victim` is still vulnerable to cyclic attacks by fine-tuning `may23-adversary` to produce a cyclic adversary `continuous-adversary` (Appendix D.2) that wins 65% of games against `dec23-victim` at 4096 victim visits. In February 2024, KataGo's developer added positions from `gift-adversary` and `continuous-adversary` into KataGo' adversarial training data set, but still we were able to fine-tune `may23-adversary` to produce yet another cyclic adversary, `big-adversary`, that beats `may24-victim` (see Appendix D.3 for training details).

Cyclic attacks still work even at high victim visits, as `big-adversary` achieves a 56% win rate against `may24-victim` at 65536 victim visits. The shape of the cyclic group is oblong and bigger than Wang et al.'s original cyclic attack, hence the name `big-adversary` (Figure 3b). The win rate is not as high as Wang et al. achieved

[1]To prevent infinite loops, most rule sets include a *superko rule* forbidding repetition of a previous board state ("positional superko", or "situational superko" if the repeated state is expanded to include not only the board position but also whose turn it is to move).

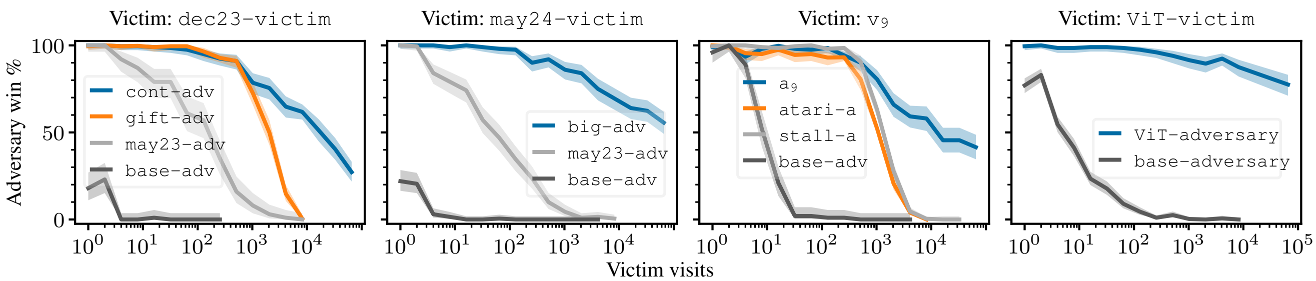
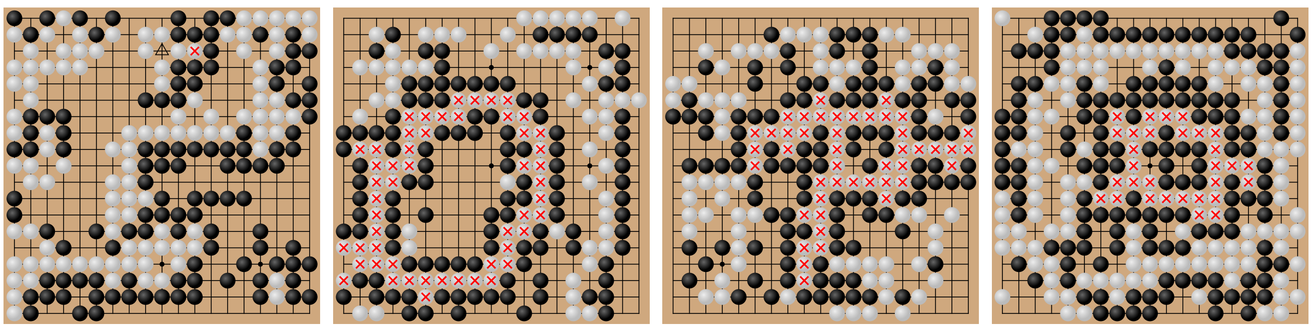


Figure 2: Win rate (y-axis) of adversaries (legend) for varying amounts of search visits (x-axis) given to victims (plot title). The adversary win rate declines with victim search budget; however, some adversaries generalize better to higher victim visit counts than others. Shaded regions are 95% Clopper-Pearson confidence intervals in this and following figures.



(a) `dec23-victim` (W) vs. `gift-adversary` (B)

(b) `may24-victim` (W) vs. `big-adversary` (B)

(c) $\mathtt{v}_9$ (W) vs. $\mathtt{a}_9$ (B)

(d) `ViT-victim` (W) vs. `ViT-adversary` (B)

Figure 3: Our learned adversarial strategies are qualitatively distinct. b, c, d show cyclic attacks with the × groups soon to be captured; these attacks use different styles of inside shapes, though these shapes have little impact on optimal play and are all easy for a human to navigate correctly. The `gift-adversary` in a follows a different strategy, inducing the victim (white) to play the stone marked × "gifting" the adversary two stones it can capture by playing at △. See full games on our website.

against the non-adversarially trained `base-victim`, suggesting that adversarial training complicates and may narrow the range of attacks. But at its current scale in KataGo, adversarial training has failed to comprehensively eliminate the cyclic vulnerability.

4 Iterated Adversarial Training

The previous section shows that an adversarially trained agent can still be vulnerable to new attacks. Can we create a robust agent by repeatedly defending against new attacks until the space of possible attacks is exhausted? In this section, we design an iterated adversarial training procedure that alternately trains a victim and an adversary. Our procedure produced a victim that was largely robust to the attacks it observed, losing only a low single-digit percentage of games. However, the victim still was not robust to new attacks, as we were able to train a new adversary to exploit the final victim.

4.1 Methodology

Our approach differs from KataGo's adversarial training (Section 3.1) in three key ways. First, we perform *iterated* adversarial training, with multiple rounds of attack and defense to train against a broader range of attacks. Second, we include a higher proportion of adversarial games in the training data: since our priority is robustness, we are more willing to take a hit in average-case capabilities than KataGo's developer. Third, we play games directly against the adversary: this method is less sample-efficient than starting from hand-curated positions, but is more scalable and does not require domain-specific knowledge.

We label the adversary and victim at iteration n of adversarial training "$\mathtt{a}_n$" and "$\mathtt{v}_n$". We initialize the victim to $\mathtt{v}_0$ = `base-victim` and the adversary to $\mathtt{a}_0$ = `base-adversary` which Wang et al. trained to defeat `base-victim`. Each subsequent iteration involves training the victim to be robust against the latest adversary, then training an adversary to attack this hardened victim. We repeat this process for 9 iterations.

$\mathtt{v}_n$ is fine-tuned from $\mathtt{v}_{n-1}$. 18% of games were played against a frozen copy of $\mathtt{a}_{n-1}$ and all other games were self-play. This mixture teaches the victim to be robust to the attack while preserving its Go capabilities. We stop the training when the victim's win rate plateaus. $\mathtt{a}_n$ is fine-tuned from $\mathtt{a}_{n-1}$ using victim-play. We stop its training after either the adversary wins most games at a threshold victim visit count, or a set maximum compute budget is reached. See Appendix E for details.

4.2 Results

Each defender $\mathtt{v}_n$ learned an effective defense against the adversary $\mathtt{a}_{n-1}$ but rarely reached a 100% win rate despite $\mathtt{a}_{n-1}$ following a weak, degenerate strategy. We tested the final victim $\mathtt{v}_9$ by pitting it against the final attacker $\mathtt{a}_9$ as well as validation adversaries trained separately from the iterated adversaries $\mathtt{a}_1, \ldots, \mathtt{a}_9$. We found the victim was still vulnerable to both attacks but is somewhat less so at high visit counts, indicating that iterated adversarial training offers partial protection against attacks.

Robustness Against the Iterated Adversaries Each victim $\mathtt{v}_n$ achieves a high win rate against the adversary $\mathtt{a}_{n-1}$ it was trained against when $\mathtt{v}_n$ uses at least 16 visits of search (Figure 4, middle). $\mathtt{v}_n$ quickly learns to beat $\mathtt{a}_{n-1} > 95\%$ of the time (Figure 66; 256 visits). However, our defense runs rarely reached a 100% win rate, and the victims were persistently vulnerable at extremely low visits (Figure 4, left).

Both victims and adversaries are able to beat opponents from all previous iterations. This is clearly shown by the adversary win rate in Figure 4 being much higher *above* the diagonal (adversary playing against older victim) than *below* the diagonal (victim playing against older adversary). In the victim case, this can be explained by the training window being large enough to contain data from all previous iterations. By contrast, since adversary training takes many more time steps, all data from one iteration exits the training window during the next iteration. This suggests that the adversary strategy transfers well to previous victims.

The final victim $\mathtt{v}_9$ is still vulnerable even at high visits. Our final adversary $\mathtt{a}_9$ wins 42% of the time against $\mathtt{v}_9$ even at 65,536 visits. We trained $\mathtt{a}_9$ for longer than preceding adversaries, but its total training compute was still only 26% of $\mathtt{v}_9$'s. (Though $\mathtt{a}_5$, $\mathtt{a}_6$, and $\mathtt{a}_8$ have relatively poor win rates against their corresponding victims, $\mathtt{a}_9$ is able to beat every $\mathtt{v}_n$ even at high victim visits (Figure 38). Therefore we think that $\mathtt{v}_5$, $\mathtt{v}_6$, and $\mathtt{v}_8$ are no more robust than $\mathtt{v}_9$ but instead benefited from their adversaries getting stuck in local minima during training.)

All adversaries $\mathtt{a}_n$ exploit a cyclic group, but there are qualitative variations in the size and location of that group, other stones, and especially the adversary's group inside the victim's cyclic group (elaborated in Appendix E.2). For example, $\mathtt{a}_9$ favors a small, minimal interior group (Figure 3c). To humans, the differences are subtle, and the difficulty of defending against them does not vary significantly. But to the victims $\mathtt{v}_n$, the representations learned do not appear to generalize smoothly between these variations.

Robustness Against New Adversaries Though the final iterated victim $\mathtt{v}_9$ bests all previous adversaries $\mathtt{a}_1$ to $\mathtt{a}_8$, the ultimate judge of a defense is whether it works against real attacks. To evaluate this, we train a new adversary `atari-adversary` (initialized to `base-adv-early`) against $\mathtt{v}_9$ (Figure 5). This is analogous to a randomly initialized adversary trained to first beat the publicly available KataGo checkpoint `base-victim` at 1 visit and then—without access to any intermediate adversarial training checkpoints $\mathtt{v}_1,\ldots,\mathtt{v}_8$—trained to attack $\mathtt{v}_9$. `atari-adversary` still learns a cyclic attack, and we name it after its tendency to leave many stones in "atari" (capturable next move) while forming an elongated cyclic shape (Fig. 20).

`atari-adversary` wins 81% of the time against $\mathtt{v}_9$ playing with 512 visits despite being trained with less than 5% of $\mathtt{v}_9$'s compute, and even without any A-MCTS search, `atari-adversary` wins 13% of games (Appendix G). The attack quickly learns to exploit $\mathtt{v}_9$ at low visits, winning over 60% of the time against $\mathtt{v}_9$ at 256 visits after just 500 V100 GPU days (Figure 5), sooner than our original `base-adversary` adversary learned to exploit `base-victim`.

However, $\mathtt{v}_9$ proves harder to attack at high visits than `base-victim`. Quadrupling to 1024 visits takes slightly more than $4\times$ the compute, largely due to the increased cost of playing training games at higher visits. `atari-adversary` plateaus after 1401 GPU days (⧫) with a meager 4% win rate at 4096 visits. We also trained another adversary `stall-adversary` (Appendix E.3), initialized from the later checkpoint `base-adversary`, to attack $\mathtt{v}_9$, but despite learning a distinct cyclic attack, its attack fails at high victim visits and its training plateaus like `atari-adversary`. In contrast, `base-adversary` generalized rapidly to beat `base-victim` at higher visits.

5 Vision Transformers

Wang et al. (2023a)'s attack works not only against KataGo but also against a range of other superhuman Go AIs such as ELF OpenGo (Tian et al. 2019), Leela Zero (Pascutto 2019a), Sai (Morandin et al. 2019), Golaxy (Beijing Shenke Technology Co. 2018), and FineArt (Tencent 2017). Though it is possible that each of these systems is vulnerable to the cyclic attack for a different reason, it is more likely that shared properties such as their convolutional neural network (CNN) backbone cause their shared vulnerability.[2] Indeed, KataGo's developer proposed that vulnerability to cyclic attacks may be a result of the CNN backbone learning a local algorithm for classifying if a group is alive that fails to generalize to larger groups (polytope 2023). However, we demonstrate that superhuman Go AIs with vision transformer (ViT) backbones are also susceptible to cyclic attacks. This suggests the shared weakness is either AlphaZero-style training or deep learning more generally.

Since no prior work has trained strong Go AIs with a ViT architecture,[3] we first trained the world's first professional-level ViT-based Go AI. We follow a training recipe similar to the one used by KataGo (Wu 2022b), except we replace the CNN backbone with a ViT (Appendix F). Our strongest

[2]Golaxy and FineArt are closed-source but likely use the same design principles as other Go AIs.

[3]Sagri et al. (2024) and Wu et al. (2024) train ViT-based and hybrid CNN-transformer Go AIs but did not validate the strength of their systems or release weights. Moreover, they used supervised learning on games generated by CNN agents and humans, whereas we trained our ViT agents only on ViT-generated self-play data.

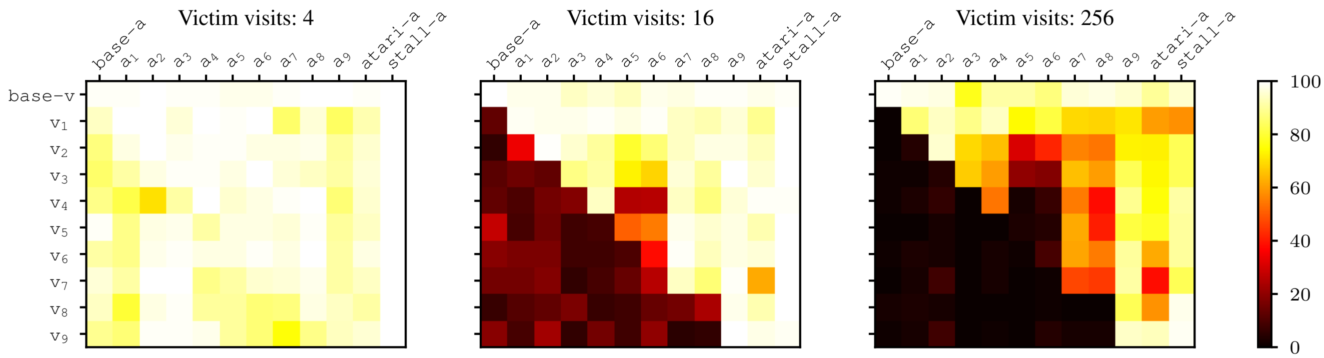


Figure 4: Win rate of all adversaries (x-axis) against all victims (y-axis) throughout iterated adversarial training for varying victim visits (plot title). The adversary $\mathtt{a}_n$ is typically able to beat the victim $\mathtt{v}_n$ it is trained to exploit (top-left-to-bottom-right diagonal), especially at 16 visits or less (middle and left plots). However, given at least 16 visits (middle and right) the victim $\mathtt{v}_n$ is typically able to beat the adversary $\mathtt{a}_{n-1}$ it trained against (elements immediately below main diagonal) along with all previous iterations $\mathtt{a}_{n-2}$, $\mathtt{a}_{n-3}$, $\cdots$. See Figure 38 for an extended version including other adversaries, victims and visit counts.

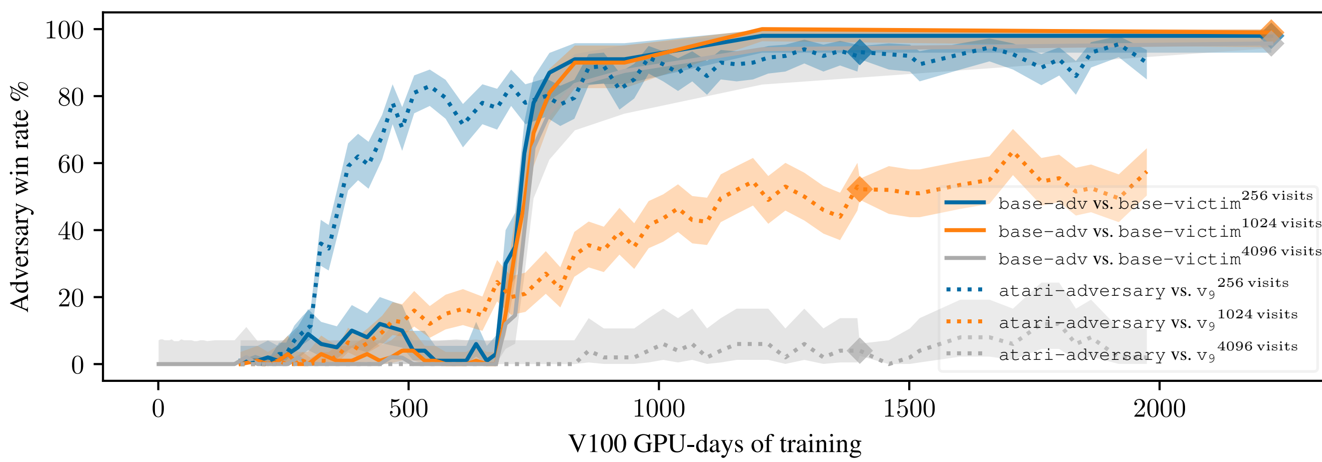


Figure 5: Win rate (y-axis) of `base-adversary` vs `base-victim` (—) and `atari-adversary` vs $\mathtt{v}_9$ ($\cdots$) by training compute (x-axis), including the 164 GPU days training `atari-adversary`'s initialization checkpoint `base-adv-early`. The checkpoint marked ♦ is used for evaluation.

ViT network, a 16-layer model `ViT-victim`, was trained for 563 V100 GPU-days. It is slower to train than a CNN agent and weaker at the same inference budget, but by our estimation it still reaches top-human levels when playing with 32,768 visits. This estimate is derived from benchmarking against KataGo, pitting our agent against players on the KGS Go Server, and winning two out of three games against Go professionals (Appendix I).

Despite the new architecture, Figure 2 shows that our `ViT-victim` (at 65536 visits) remains vulnerable to the cyclic attack, losing 78% of games to a fine-tuned version of `base-adversary`, which we call `ViT-adversary` (Appendix F.5). `ViT-adversary`'s strategy resembles other cyclic attacks but is qualitatively distinct in its tendency to produce small groups inside the cyclic one, and dense board states with limited open space (Figure 3d). This attack can be replicated by a human expert (Appendix J).

Remarkably, `ViT-victim` (at 512 visits) also loses 2.5% of games to the original `base-adversary`—similar to the zero-shot transfer to CNN Go AIs such as ELF OpenGo reported by Wang et al. (2023a). `base-adversary` certainly does not win through strong Go ability: it is a very weak strategy that loses to amateur human players (Wang et al. 2023a). `ViT-victim` even misevaluates cyclic positions on boards as small as 11×11 (Appendix H). We conclude that CNN architectures are not the cause of the cyclic vulnerability, though we do not rule out the possibility that a much deeper ViT or some other architecture would solve the cyclic vulnerability.

6 Related Work

We focus on robustness against *adversarial policies*: strategies designed to make an opponent perform poorly. Adversarial policies give an empirical lower bound for an agent's *exploitability*: its worst-case loss relative to Nash equilibria (Timbers et al. 2022). Gleave et al. (2020) previously explored such policies in a zero-sum game between simulated humanoids trained with self-play. The policies (Bansal et al. 2018) attacked by Gleave et al. were below human performance, raising the question: were the agents vulnerable because of their limited capability? To investigate this, Wang et al. (2023a) searched for adversarial policies against the superhuman Go AI KataGo (Wu 2020), finding a strategy that beats KataGo in 97% of games.

We focus on KataGo (Wu 2020) as it is the most capable open-source Go AI. Moreover, other superhuman open-source Go AIs such as ELF OpenGo (Tian et al. 2019) and Leela OpenZero (Pascutto 2019b) all follow the same basic AlphaZero-style training architecture. However, alternative multi-agent reinforcement learning methods may be more robust. Approaches that maintain a population of strategies are promising (Vinyals et al. 2019; Czempin and Gleave 2022; Lanctot et al. 2017). Another alternative, counterfactual regret minimization (Zinkevich et al. 2007), has been used to beat professional human poker players (Brown and Sandholm 2018). Furthermore, Perolat et al. (2022) found a way to approximate Nash equilibria (Perolat et al. 2021) that scaled to the board game Stratego, whose game tree is 10^{175} times larger than Go's.

We replace the CNN backbone of KataGo with a vision transformer (ViT) and train the ViT agent to a superhuman level, finding it to be slower to train than a CNN agent and weaker at the same inference budget. By contrast, Sagri et al. (2024) found the transformer-based EfficientFormer architecture (Li et al. 2022) performed similarly to CNNs for Go—however, their models were trained only with supervised learning, not self-play. Transformers have been investigated more thoroughly in chess. Our results are consistent with Czech, Blüml, and Kersting (2023) who found that CNNs are stronger at chess than both ViTs and a ViT-CNN hybrid at a given inference budget. Yet transformers have shown strong performance, with the transformer-based Leela Chess Zero (Pascutto and Linscott 2019; Monroe 2023) winning the Top Chess Engine Championship Cup 11 (TCEC 2023).

Although we find ViTs weaker than CNNs in average-case capabilities, our primary metric is *robustness*. Past research in image classification indicates ViTs are modestly more robust than CNNs against adversarial perturbations and other out-of-distribution inputs (Benz et al. 2021; Shao et al. 2022; Bhojanapalli et al. 2021; Zhang et al. 2022; Paul and Chen 2022), although some research contests this (Bai et al. 2021; Mahmood, Mahmood, and van Dijk 2021; Tang et al. 2022; Pinto, Torr, and K. Dokania 2022; Wang et al. 2023b). Even if ViTs are not more robust, their differing inductive biases might cause them to fail in *different* ways to CNNs, with prior work finding that ViTs are more vulnerable to patch perturbations (Fu et al. 2022). Surprisingly, we find that Wang et al. (2023a)'s attack transfers zero-shot to our ViT agent. In concurrent work, Wu et al. (2024) find that a hybrid CNN-transformer architecture beats a similarly sized CNN in the average case and is more robust yet still vulnerable against a set of cyclic positions from Wang et al. (2023a).

7 Discussion and Future Work

We explore three natural approaches for defending against adversarial attacks in Go: adversarial training with hand-constructed positions, iterated adversarial training, and using a ViT instead of a CNN. The defenses make attacks harder, increasing the amount of compute needed to successfully beat the defended systems. However, none of the defenses make attacks impossible—the attack algorithm from Wang et al. (2023a) is always able to find a successful attack, often with a small fraction of the compute used to train the victims. Moreover, none of the defenses achieve the robustness of a human (Appendix B.1), and humans are even able to execute several attacks against the defenses (Appendix J). Our results highlight the challenge of defending against all possible attacks, or even all possible cyclic attacks, suggesting an offense-defense balance (Jervis 1978) favoring attackers.

We do, however, find it cheap to defend (relative to the cost to attack) against any *fixed* (i.e. non-adaptive) attack. This suggests that it may be possible to fully defend a Go AI by training against a large enough corpus of attacks. We propose two complementary routes to fully robust Go AIs.

The first approach is to increase the size of the attack corpus by developing new attack algorithms that require less compute to train a variety of adversaries. Our version of iterated adversarial training was bottlenecked by the attack component, which took 18x more compute than the defense component (see Table 5). Reducing the time taken to find new attacks, e.g., with relaxed (Christiano 2019; Hubinger 2019) or latent adversarial training (Sankaranarayanan et al. 2018; Casper et al. 2024), would allow for training against many more attacks.

The second approach is to increase the sample efficiency of adversarial training by making the victim generalize from a limited number of adversarial strategies. Existing algorithms for training Go AIs do not generalize in this way: `may24-victim` and v_9 remain vulnerable to cyclic attacks even after training against many variants of cyclic attacks.

There are also routes to robustness besides adversarial training. For example, multi-agent reinforcement learning schemes like PSRO (Lanctot et al. 2017) or DeepNash (Perolat et al. 2022) may be able to automatically discover strategies like the cyclic attack and train them away. Another possibility is to change the threat model and use online or *stateful* defenses (Chen, Carlini, and Wagner 2020) which can dynamically update the victim in tandem with adversaries who are trying to learn an exploit.

Our results highlight that existing difficulties in building robust AI systems may not be resolved by scaling up average-case capabilities and applying straightforward defenses. We recommend further systematic evaluations of robustness and how it behaves under increasing capabilities in other domains, especially ones where frontier models are beginning to attain new superhuman levels. But if we are unable to achieve robustness in strongly superhuman agents in the well-defined and self-contained domain of Go, achieving robustness in more open-ended real-world applications will likely be even more challenging.

Acknowledgments

We thank David Wu for discussing KataGo and its adversarial training with us and for helping us qualitatively describe the two attacks we found in Section 3, Adrià Garriga-Alonso for infrastructure support when running experiments, and ChengCheng Tan for helping create Figure 1. We also thank ChengCheng Tan, Derik Kauffmann, Siao Si Looi, David Wu, Micah Carroll, and Daniel Filan for feedback on early drafts. Finally, we thank Yilun Yang (7 dan professional) and Ryan Li (4 dan professional) for playing `ViT-victim` to evaluate its strength, and Matthew Harwit for helping connect the authors with professional Go players.

Disclosure of funding

Tom Tseng, Euan McLean, and Adam Gleave were employed by FAR.AI, a non-profit research institute, and supported by FAR.AI's unrestricted funds. Kellin Pelrine was supported by funding from IVADO and by the Fonds de recherche du Quebéc. Tony Wang was supported by a Lightspeed grant.

Author contributions

Tom Tseng was the primary technical individual contributor, implementing the majority of the code and running the majority of the experiments. Euan McLean prepared an initial draft of the paper from high-level comments provided by technical contributors, edited the resulting paper, and coordinated the write-up. Kellin Pelrine, Tony Wang, and Adam Gleave were joint co-advisors throughout the project, and helped with paper writing and editing. In addition, Kellin Pelrine analysed the Go games and replicated attacks by hand, Tony Wang set up KGS bots for human evaluation, and Adam Gleave managed the project.

References

Bai, Y.; Mei, J.; Yuille, A. L.; and Xie, C. 2021. Are transformers more robust than CNNs? *Advances in Neural Information Processing Systems*, 34: 26831–26843.

Bansal, T.; Pachocki, J.; Sidor, S.; Sutskever, I.; and Mordatch, I. 2018. Emergent Complexity via Multi-Agent Competition. In *International Conference on Learning Representations*.

Beijing Shenke Technology Co., L. 2018. Golaxy. https://www.19x19.com/. Accessed: 2024-03-28.

Benz, P.; Ham, S.; Zhang, C.; Karjauv, A.; and Kweon, I. S. 2021. Adversarial Robustness Comparison of Vision Transformer and MLP-Mixer to CNNs. In *British Machine Vision Conference*, 25. BMVA Press.

Bhojanapalli, S.; Chakrabarti, A.; Glasner, D.; Li, D.; Unterthiner, T.; and Veit, A. 2021. Understanding robustness of transformers for image classification. In *IEEE/CVF International Conference on Computer Vision*, 10231–10241.

Brown, N.; and Sandholm, T. 2018. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374): 418–424.

Casper, S.; Schulze, L.; Patel, O.; and Hadfield-Menell, D. 2024. Defending Against Unforeseen Failure Modes with Latent Adversarial Training. arXiv:2403.05030.

Chen, S.; Carlini, N.; and Wagner, D. 2020. Stateful detection of black-box adversarial attacks. In *Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence*, 30–39.

Christiano, P. 2019. Worst-case guarantees. https://ai-alignment.com/training-robust-corrigibility-ce0e0a3b9b4d. Accessed: 2024-06-15.

Croce, F.; Andriushchenko, M.; Sehwag, V.; Debenedetti, E.; Flammarion, N.; Chiang, M.; Mittal, P.; and Hein, M. 2021. RobustBench: a standardized adversarial robustness benchmark. In *Advances in Neural Information Processing Systems*.

Czech, J.; Blüml, J.; and Kersting, K. 2023. Representation Matters: The Game of Chess Poses a Challenge to Vision Transformers. arXiv:2304.14918.

Czempin, P.; and Gleave, A. 2022. Reducing Exploitability with Population Based Training. In *International Conference on Machine Learning Workshop on New Frontiers in Adversarial Machine Learning*.

Fu, Y.; Zhang, S.; Wu, S.; Wan, C.; and Lin, Y. 2022. Patch-Fool: Are Vision Transformers Always Robust Against Adversarial Perturbations? In *International Conference on Learning Representations*.

Gleave, A.; Dennis, M.; Wild, C.; Kant, N.; Levine, S.; and Russell, S. 2020. Adversarial Policies: Attacking Deep Reinforcement Learning. In *International Conference on Learning Representations*.

Hubinger, E. 2019. Relaxed adversarial training for inner alignment. https://www.alignmentforum.org/posts/9Dy5YRaoCxH9zuJqa. Accessed: 2024-06-15.

Jervis, R. E. 1978. Cooperation under the Security Dilemma. *World Politics*, 30: 167 – 214.

Lanctot, M.; Zambaldi, V.; Gruslys, A.; Lazaridou, A.; Tuyls, K.; Perolat, J.; Silver, D.; and Graepel, T. 2017. A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 30, 4190–4203.

Li, Y.; Yuan, G.; Wen, Y.; Hu, J.; Evangelidis, G.; Tulyakov, S.; Wang, Y.; and Ren, J. 2022. EfficientFormer: Vision transformers at MobileNet speed. *Advances in Neural Information Processing Systems*, 35: 12934–12949.

Liu, C.; Dong, Y.; Xiang, W.; Yang, X.; Su, H.; Zhu, J.; Chen, Y.; He, Y.; Xue, H.; and Zheng, S. 2023. A Comprehensive Study on Robustness of Image Classification Models: Benchmarking and Rethinking. arXiv:2302.14301.

Mahmood, K.; Mahmood, R.; and van Dijk, M. 2021. On the Robustness of Vision Transformers to Adversarial Examples. In *IEEE/CVF International Conference on Computer Vision*, 7838–7847.

Mazeika, M.; Phan, L.; Yin, X.; Zou, A.; Wang, Z.; Mu, N.; Sakhaee, E.; Li, N.; Basart, S.; Li, B.; Forsyth, D.; and Hendrycks, D. 2024. HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal. arXiv:2402.04249.

Monroe, D. 2023. Leela Chess Zero Training README. https://github.com/Ergodice/lczero-training/blob/0c10d4e19fbfd28abb167f9134fee74c983ef6db/README.md. Accessed: 2024-02-06.

Morandin, F.; Amato, G.; Gini, R.; Metta, C.; Parton, M.; and Pascutto, G.-C. 2019. SAI a Sensible Artificial Intelligence that plays Go. In *International Joint Conference on Neural Networks*. IEEE.

Pascutto, G.-C. 2019a. Leela Zero. https://github.com/leela-zero/leela-zero/. Accessed: 2024-03-27.

Pascutto, G.-C. 2019b. Leela Zero. https://zero.sjeng.org/. Accessed: 2022-06-16.

Pascutto, G.-C.; and Linscott, G. 2019. Leela Chess Zero. https://lczero.org/. Accessed: 2024-02-06.

Paul, S.; and Chen, P.-Y. 2022. Vision transformers are robust learners. In *AAAI Conference on Artificial Intelligence*, volume 36(2), 2071–2081.

Perolat, J.; De Vylder, B.; Hennes, D.; Tarassov, E.; Strub, F.; de Boer, V.; Muller, P.; Connor, J. T.; Burch, N.; Anthony, T.; et al. 2022. Mastering the game of Stratego with model-free multiagent reinforcement learning. *Science*, 378(6623): 990–996.

Perolat, J.; Munos, R.; Lespiau, J.-B.; Omidshafiei, S.; Rowland, M.; Ortega, P.; Burch, N.; Anthony, T.; Balduzzi, D.; De Vylder, B.; Piliouras, G.; Lanctot, M.; and Tuyls, K. 2021. From Poincaré Recurrence to Convergence in Imperfect Information Games: Finding Equilibrium via Regularization. In *International Conference on Machine Learning*, volume 139, 8525–8535.

Pinto, F.; Torr, P. H.; and K. Dokania, P. 2022. An impartial take to the CNN vs transformer robustness contest. In *European Conference on Computer Vision*, 466–480. Springer.

polytope. 2023. Comment on There are (probably) no superhuman Go AIs: strong human players beat the strongest AIs. https://www.lesswrong.com/posts/Es6cinTyuTq3YAcoK/there-are-probably-no-superhuman-go-ais-strong-human-players?commentId=gAEovdd5iGsfZ48H3. Accessed: 2024-02-14.

Sagri, A.; Cazenave, T.; Arjonilla, J.; and Saffidine, A. 2024. Vision Transformers for Computer Go. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, 376–388. Springer.

Sankaranarayanan, S.; Jain, A.; Chellappa, R.; and Lim, S. N. 2018. Regularizing deep networks using efficient layerwise adversarial training. arXiv:1705.07819.

Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; Lillicrap, T.; and Silver, D. 2020. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609.

Shao, R.; Shi, Z.; Yi, J.; Chen, P.-Y.; and Hsieh, C.-J. 2022. On the Adversarial Robustness of Vision Transformers. *Transactions on Machine Learning Research*.

Shapley, L. S. 1953. Stochastic Games. *PNAS*, 39(10): 1095–1100.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587): 484–489.

Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; Lillicrap, T.; Simonyan, K.; and Hassabis, D. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419): 1140–1144.

Tang, S.; Gong, R.; Wang, Y.; Liu, A.; Wang, J.; Chen, X.; Yu, F.; Liu, X.; Song, D.; Yuille, A.; Torr, P. H. S.; and Tao, D. 2022. RobustART: Benchmarking Robustness on Architecture Design and Training Techniques. arXiv:2109.05211.

TCEC. 2023. LCZero 0.30-dag-dcb4ece9-BT2-3250000 vs Stockfish dev16_202301021914 - TCEC - Archived Game. https://tcec-chess.com/#game=1&round=fl&season=cup11. Accessed: 2024-04-28.

Tencent. 2017. FineArt. https://en.wikipedia.org/wiki/Fine_Art_(software). Accessed: 2024-03-28.

Tian, Y.; Ma, J.; Gong, Q.; Sengupta, S.; Chen, Z.; Pinkerton, J.; and Zitnick, L. 2019. ELF OpenGo: an analysis and open reimplementation of AlphaZero. In *International Conference on Machine Learning*.

Timbers, F.; Bard, N.; Lockhart, E.; Lanctot, M.; Schmid, M.; Burch, N.; Schrittwieser, J.; Hubert, T.; and Bowling, M. 2022. Approximate exploitability: Learning a best response in large games. arXiv:2004.09677.

Tseng, T.; McLean, E.; Pelrine, K.; Wang, T. T.; and Gleave, A. 2024. Can Go AIs be adversarially robust? arXiv:2406.12843.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness May Be at Odds with Accuracy. In *International Conference on Learning Representations*.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.

Wang, T. T.; Gleave, A.; Tseng, T.; Pelrine, K.; Belrose, N.; Miller, J.; Dennis, M. D.; Duan, Y.; Pogrebniak, V.; Levine, S.; and Russell, S. 2023a. Adversarial Policies Beat Superhuman Go AIs. In *International Conference on Machine Learning*, 35655–35739. PMLR.

Wang, Z.; Bai, Y.; Zhou, Y.; and Xie, C. 2023b. Can CNNs Be More Robust Than Transformers? In *The Eleventh International Conference on Learning Representations*.

Wu, D. 2022a. Discord comment on the initial KataGo adversarial training. https://discord.com/channels/417022162348802048/583775968804732928/1052951418685882408. Accessed: 2025-01-24.

Wu, D. 2023a. Discord comment on KataGo adversarial training data sources. https://discord.com/channels/417022162348802048/723268423588642948/1131951228495081543. Accessed: 2025-01-24.

Wu, D. 2023b. KataGo should be partially resistant to cyclic groups now. https://www.reddit.com/r/baduk/comments/14prv4f/katago_should_be_partially_resistant_to_cyclic/. Accessed: 2025-01-24.

Wu, D. J. 2020. Accelerating Self-Play Learning in Go. In *AAAI Workshop on Reinforcement Learning in Games*.

Wu, D. J. 2021. KataGo's Supported Go Rules (Version 2). https://lightvector.github.io/KataGo/rules.html. Accessed: 2022-09-27.

Wu, D. J. 2022b. KataGo - Networks for kata1. https://katagotraining.org/networks/. Accessed: 2022-09-26.

Wu, T.-L.; Wu, T.-R.; Shih, C.-C.; Ju, Y.-R.; Wu, I.; et al. 2024. ResTNet: Defense against Adversarial Policies via Transformer in Computer Go. arXiv:2410.05347.

Zhang, C.; Zhang, M.; Zhang, S.; Jin, D.; Zhou, Q.; Cai, Z.; Zhao, H.; Liu, X.; and Liu, Z. 2022. Delving deep into the generalization of vision transformers under distribution shifts. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 7277–7286.

Zinkevich, M.; Johanson, M.; Bowling, M.; and Piccione, C. 2007. Regret Minimization in Games with Incomplete Information. In *Advances in Neural Information Processing Systems*, volume 20.