General search techniques without common knowledge for imperfect-information games, and application to superhuman Fog of War chess

Tuomas Sandholm

Carnegie Mellon University

Strategy Robot, Inc.

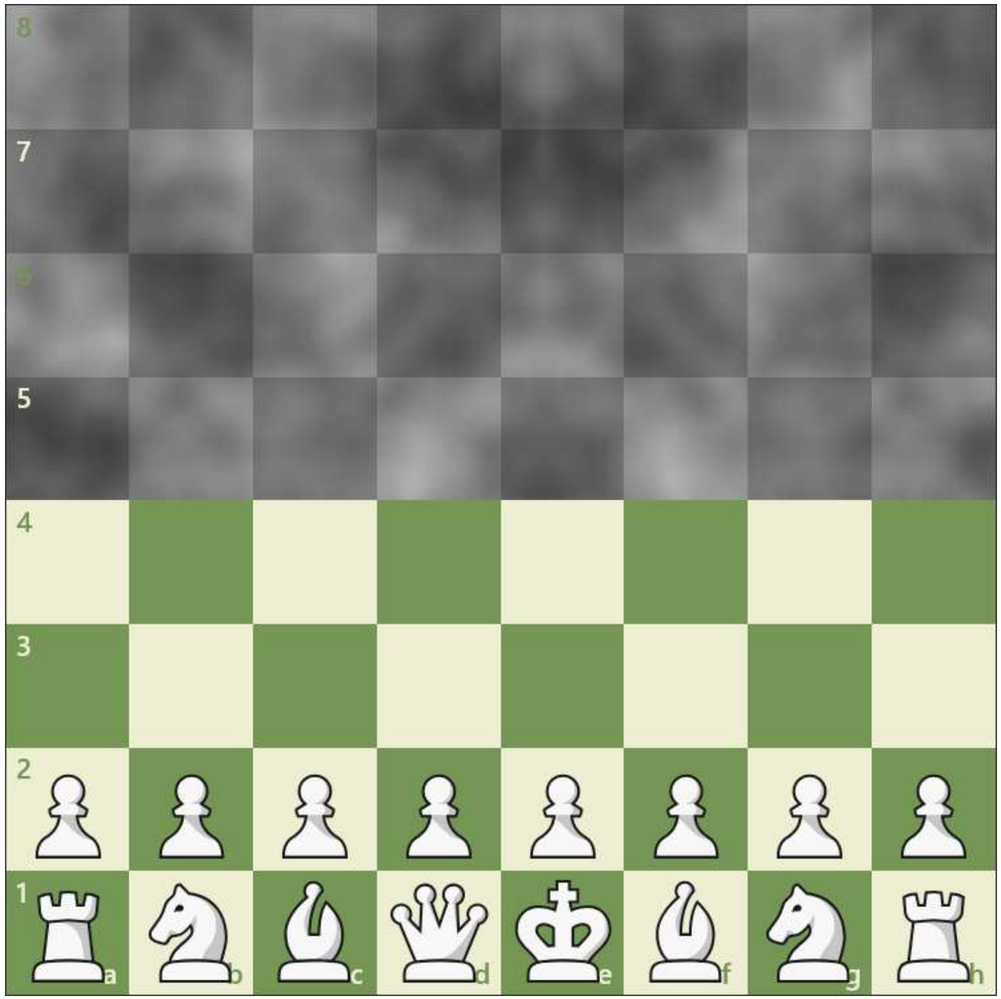
Strategic Machine, Inc.

Optimized Markets, Inc.

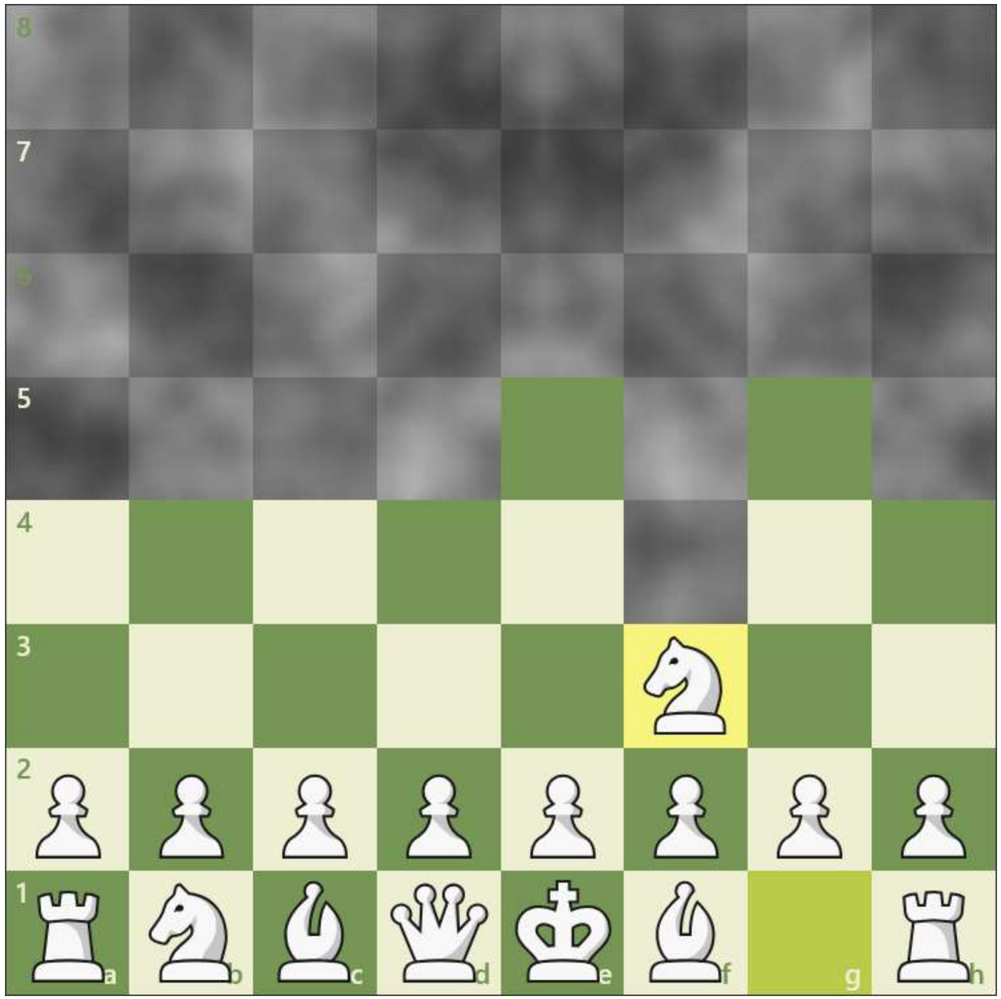
Joint work with my PhD student **Brian Hu Zhang**.

History of AI for Games

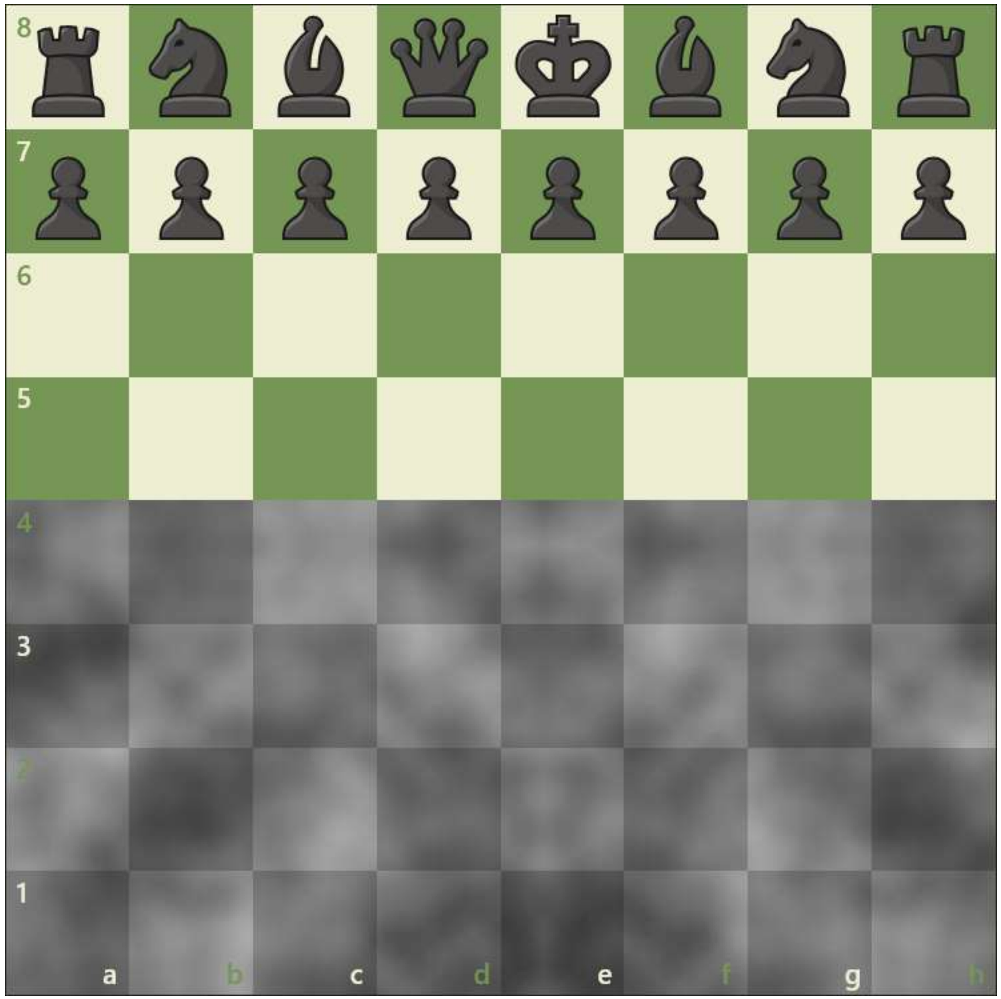
Fog of War Chess (aka. Dark Chess)



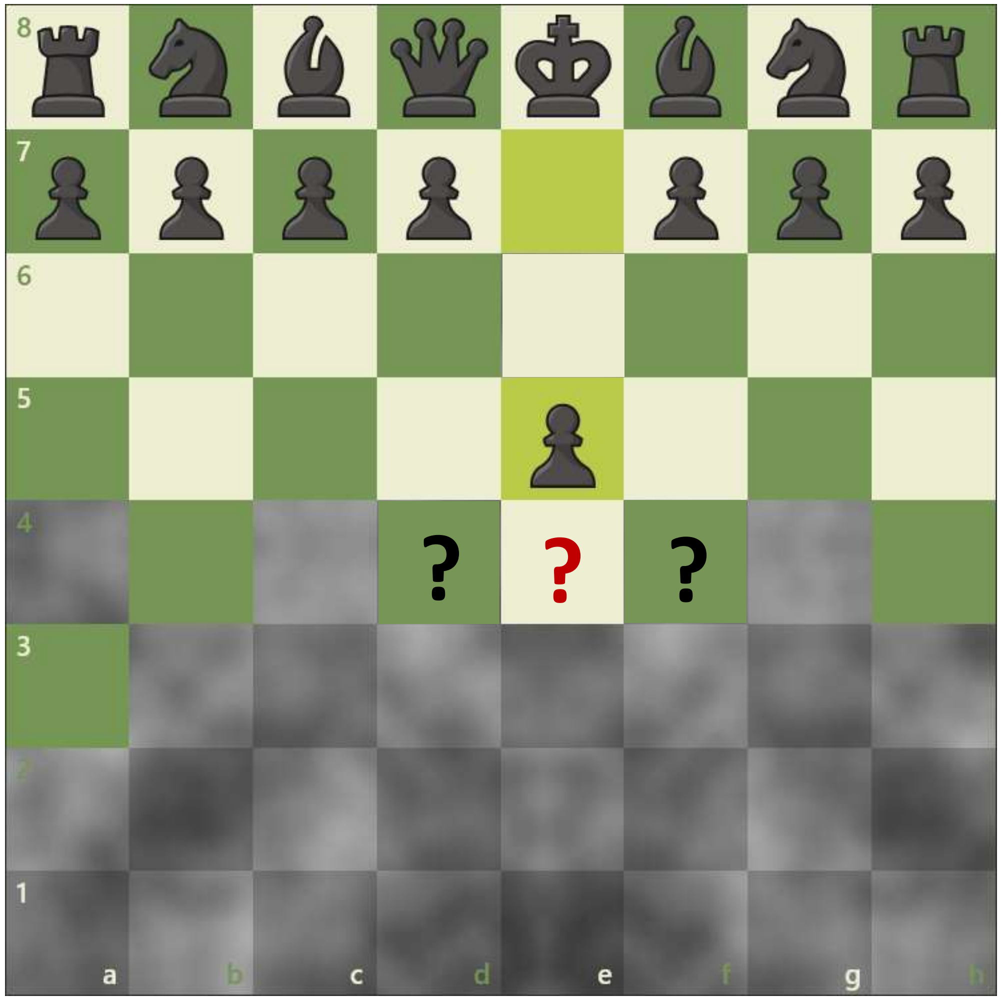
Fog of War Chess (aka. Dark Chess)



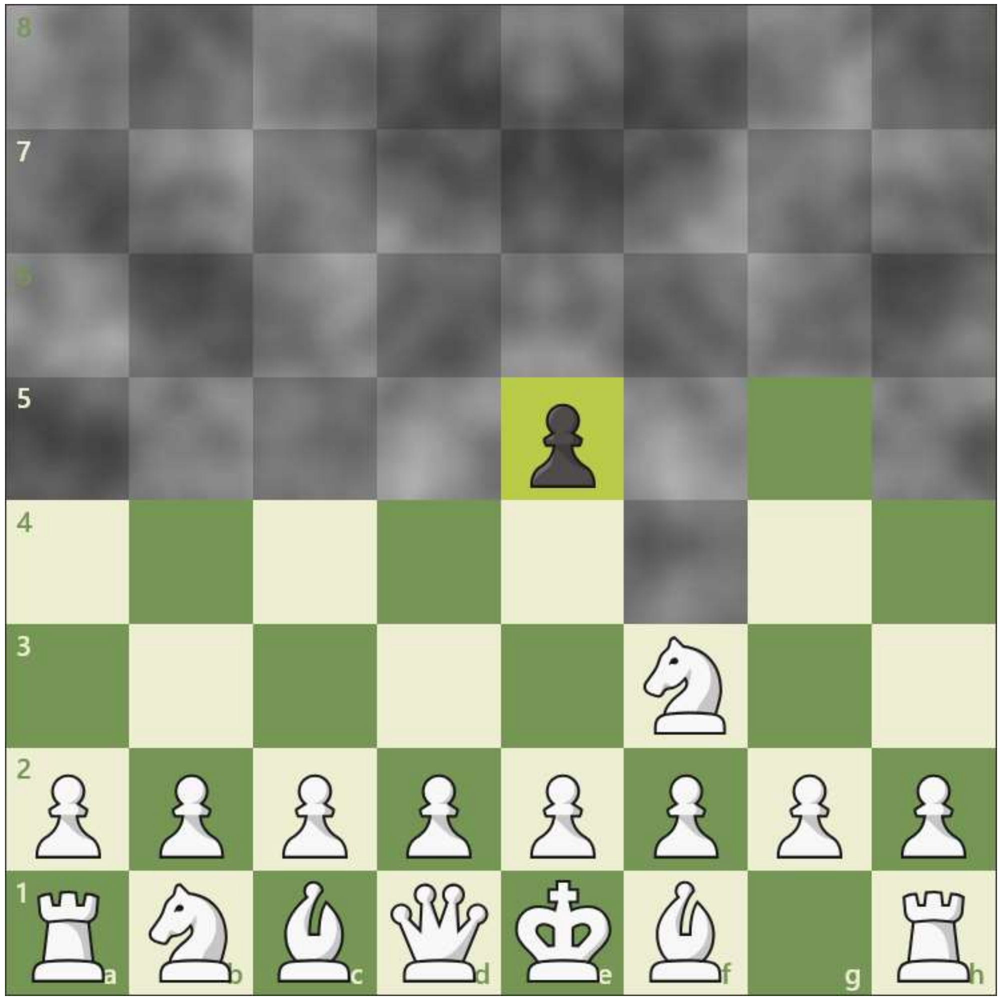
Fog of War Chess (aka. Dark Chess)



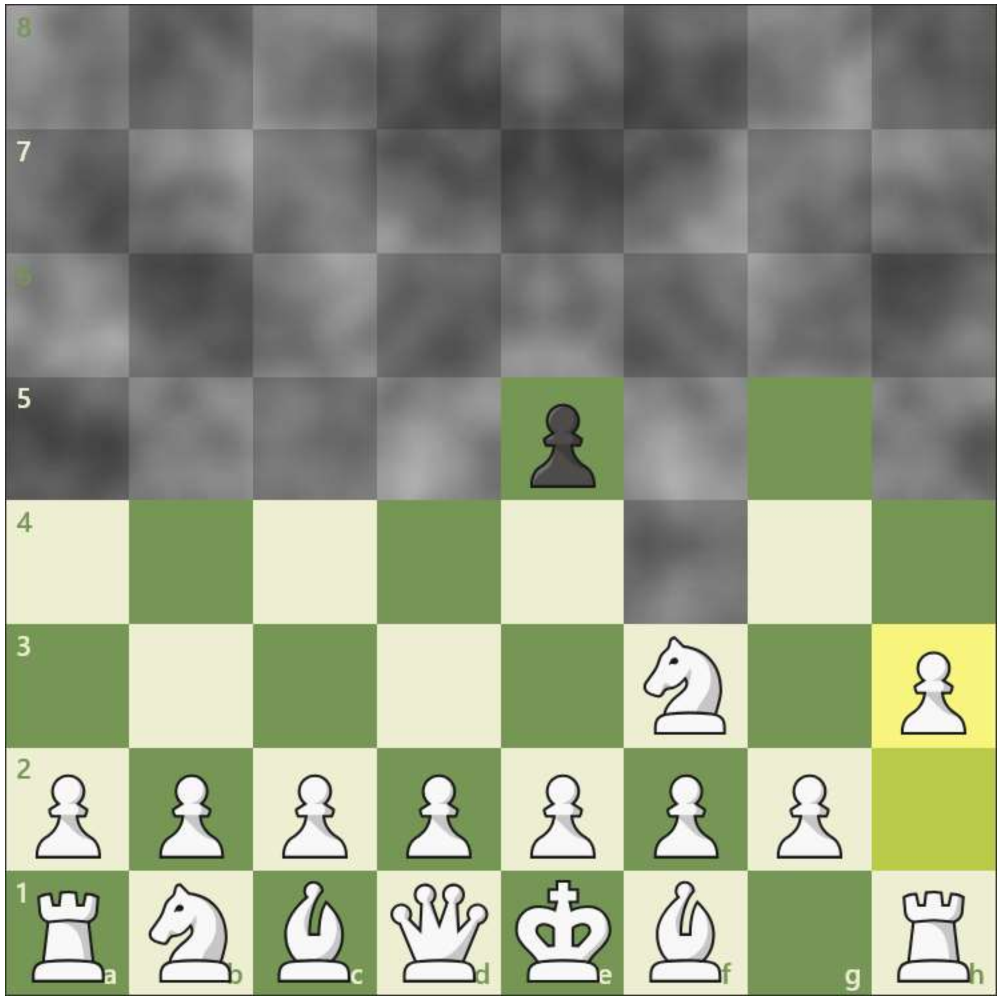
Fog of War Chess (aka. Dark Chess)



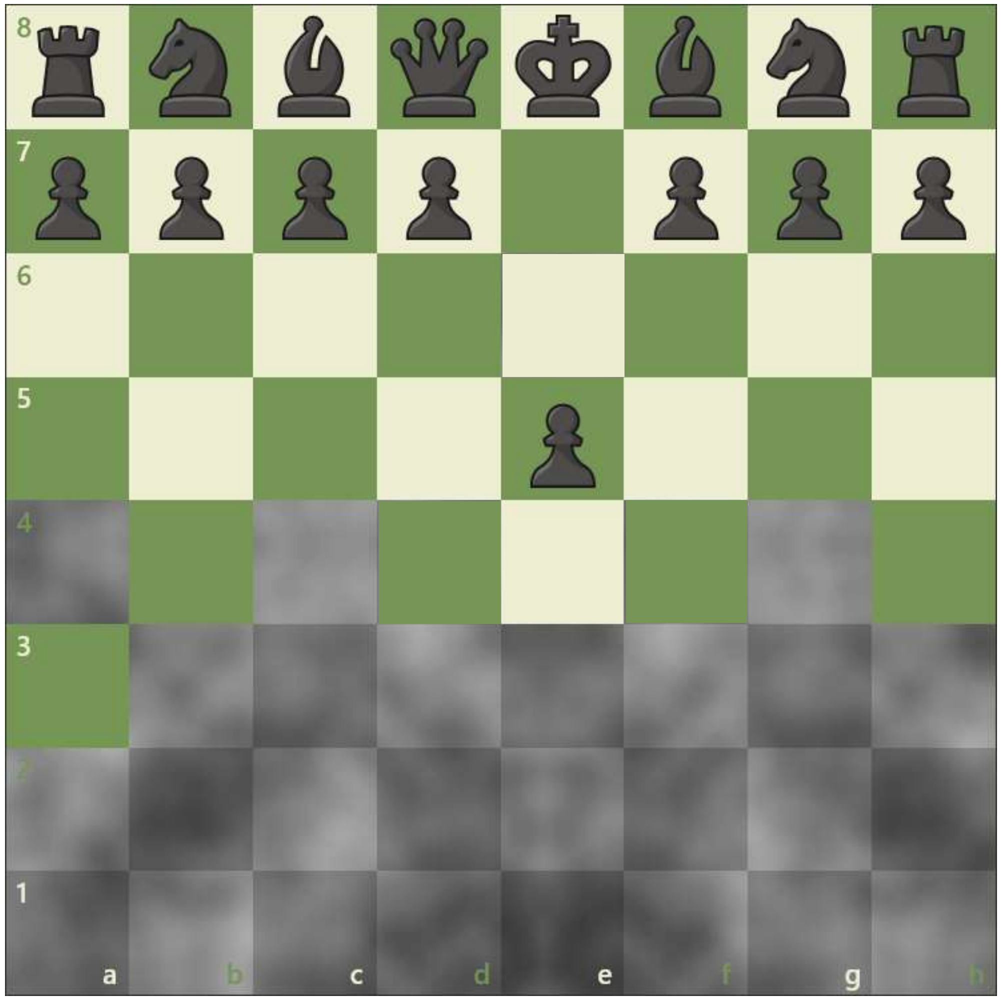
Fog of War Chess (aka. Dark Chess)



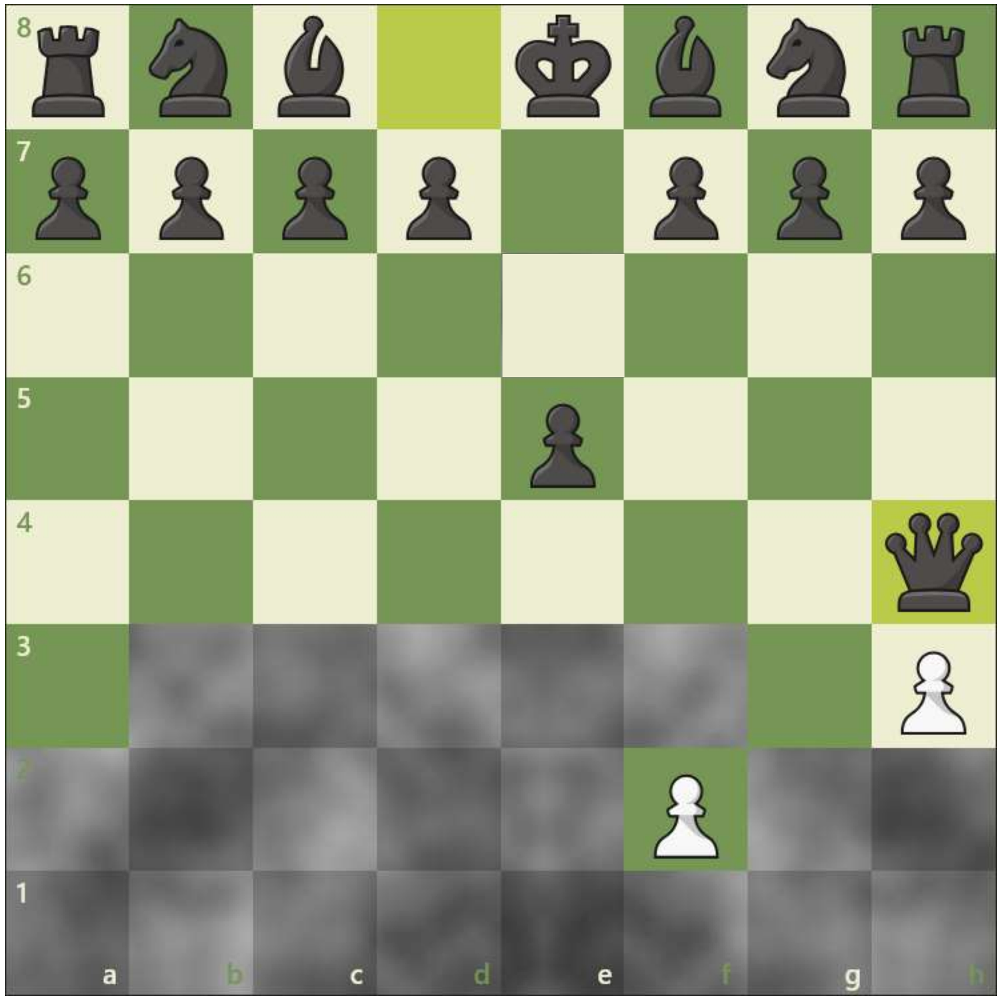
Fog of War Chess (aka. Dark Chess)



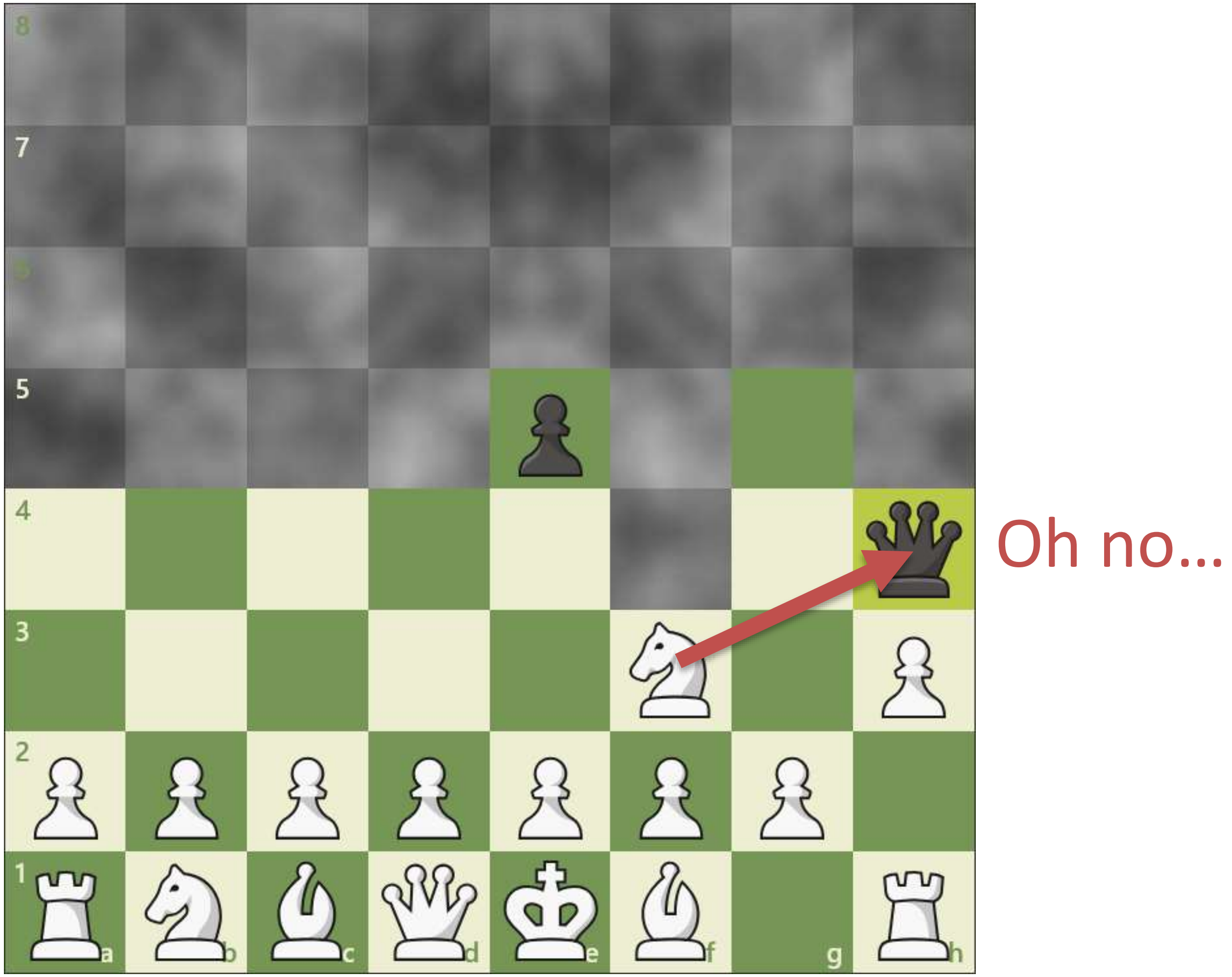
Fog of War Chess (aka. Dark Chess)



Fog of War Chess (aka. Dark Chess)



Fog of War Chess (aka. Dark Chess)



*k*th-order knowledge

A

Position after 1. Nf3 e5 2. h3 Qh4

B

Position after 1. Nc3 g5 2. Nh3 d5

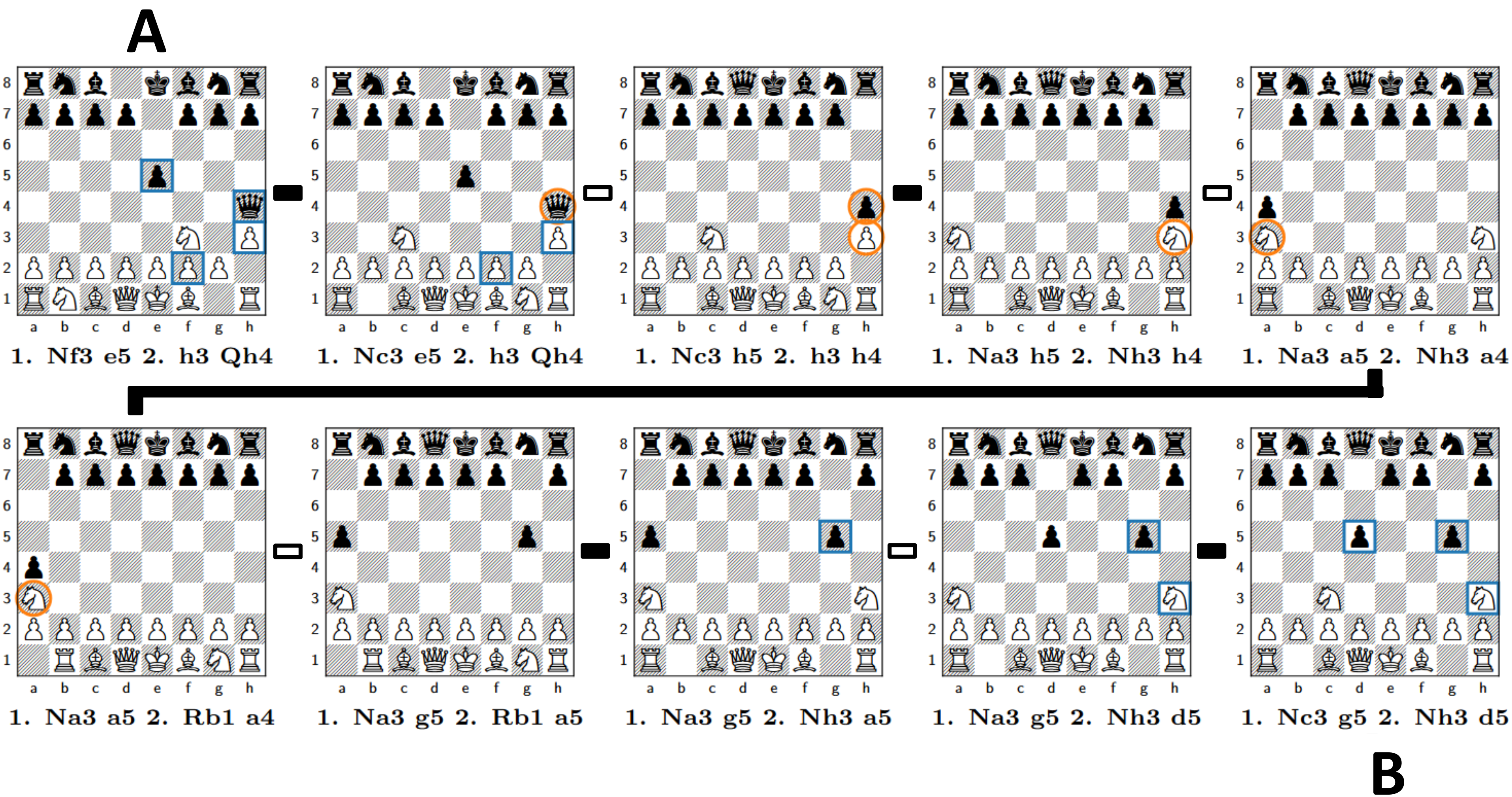
Question: In Position **A**, is it common knowledge that the position is *not* **B**?

No… but it is 8th-order knowledge

Everyone knows that everyone knows that everyone knows that… the position is not **B**

True for 8 repetitions, but false for 9

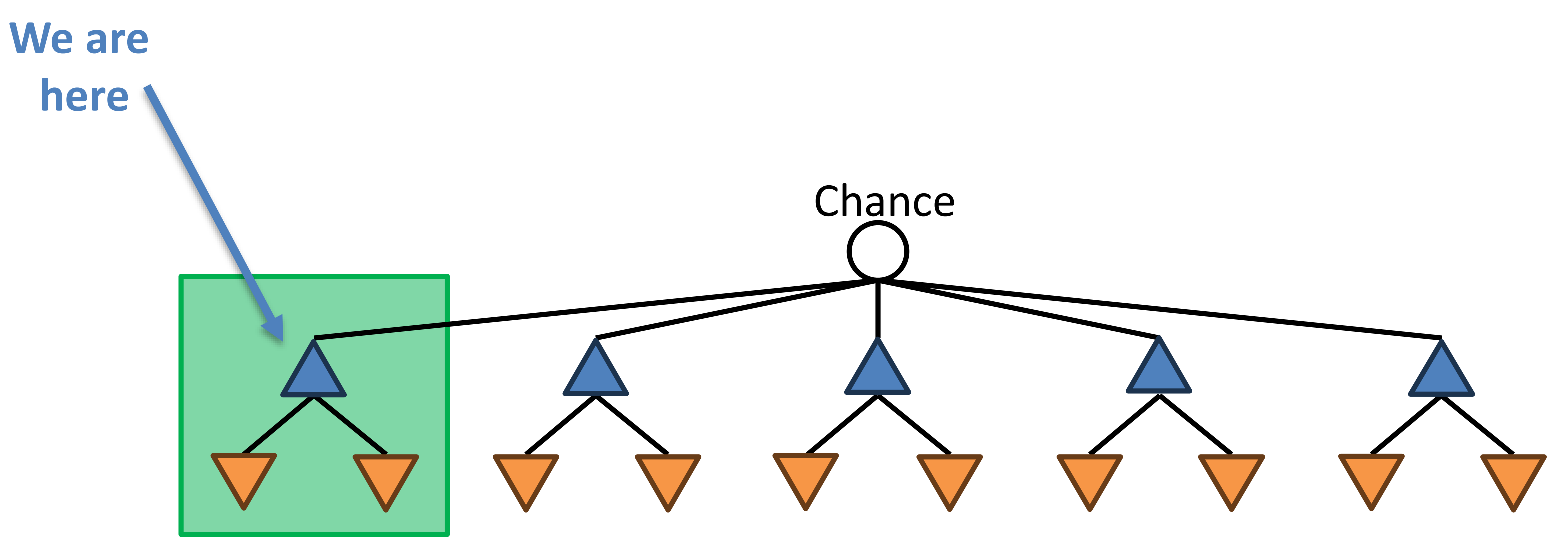
*k*th-order knowledge



Everyone knows that everyone knows that everyone knows that… the position is not **B**

True for 8 repetitions, but false for 9

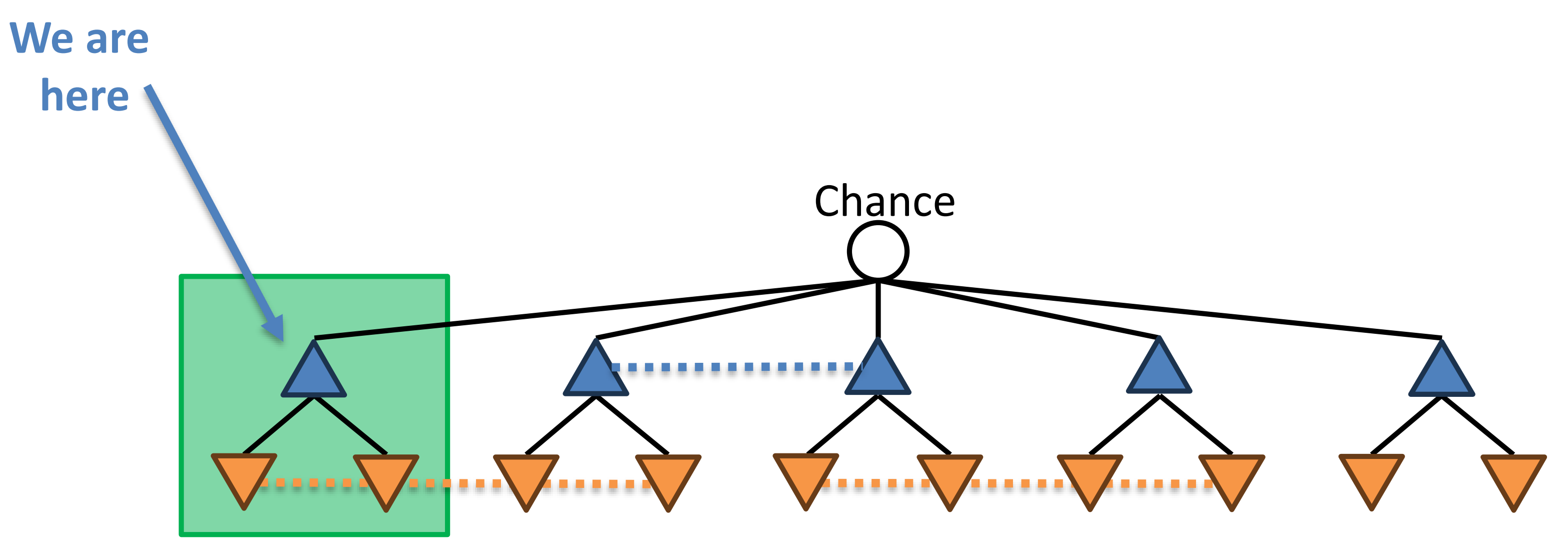
Real-time subgame solving in *perfect*-information games



Real-time subgame solving in *imperfect*-information games

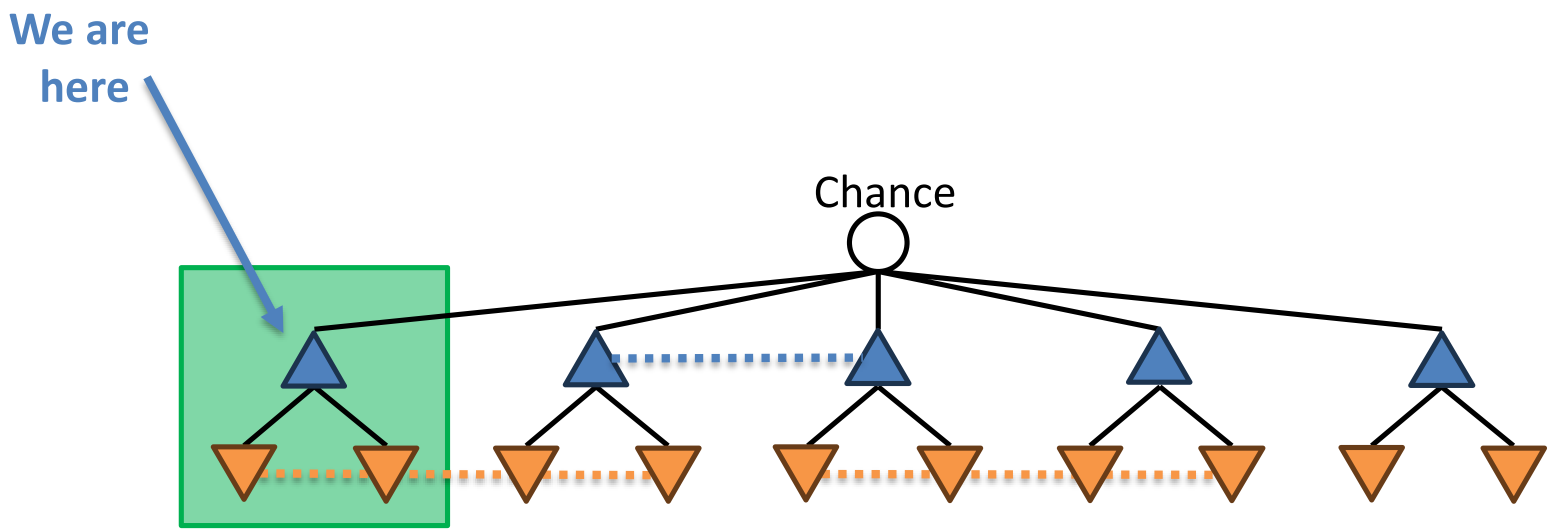
Can't solve the subgame in isolation, because the solution may depend on other parts of the game!

What now?



Real-time subgame solving in *imperfect*-information games

Idea [Gilpin & Sandholm AAAI-06]: Start with a ***blueprint strategy*** (*e.g.*, computed in a coarse abstraction, via deep learning, or human heuristics). When we arrive at a decision point, try to **refine** the strategy



Real-time subgame solving in *imperfect*-information games

History

- **“Unsafe”** methods
  [Gilpin & Sandholm AAAI-06, Gilpin & Sandholm AAMAS-07, Gilpin, Sandholm & Sørensen AAMAS-08, Ganzfried & Sandholm AAMAS-15…]

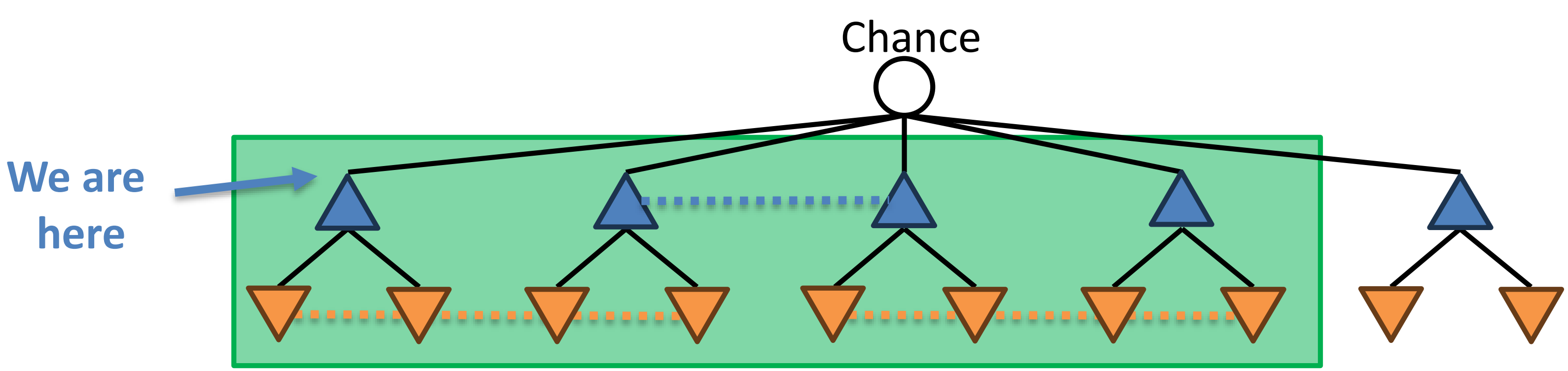
- **“Safe”** methods (subgame solving doesn’t make the agent more exploitable than the blueprint)
  [Burch *et al.* AAAI-14, Moravcik *et al.* AAAI-16, Schmid *et al.* AAAI-16, Brown & Sandholm NeurIPS-17…]

Real-time subgame solving in *imperfect*-information games

Prior subgame solving approaches are based on the ***common-knowledge subgame***

Definition:

- Two nodes in the same layer of the game tree are *connected* if there is an information set connecting some descendant of the first node to some descendant of the second node
- The ***common-knowledge subgame*** at a node h consists of all nodes *recursively* connected to h, and all their descendants.



Real-time subgame solving in *imperfect*-information games

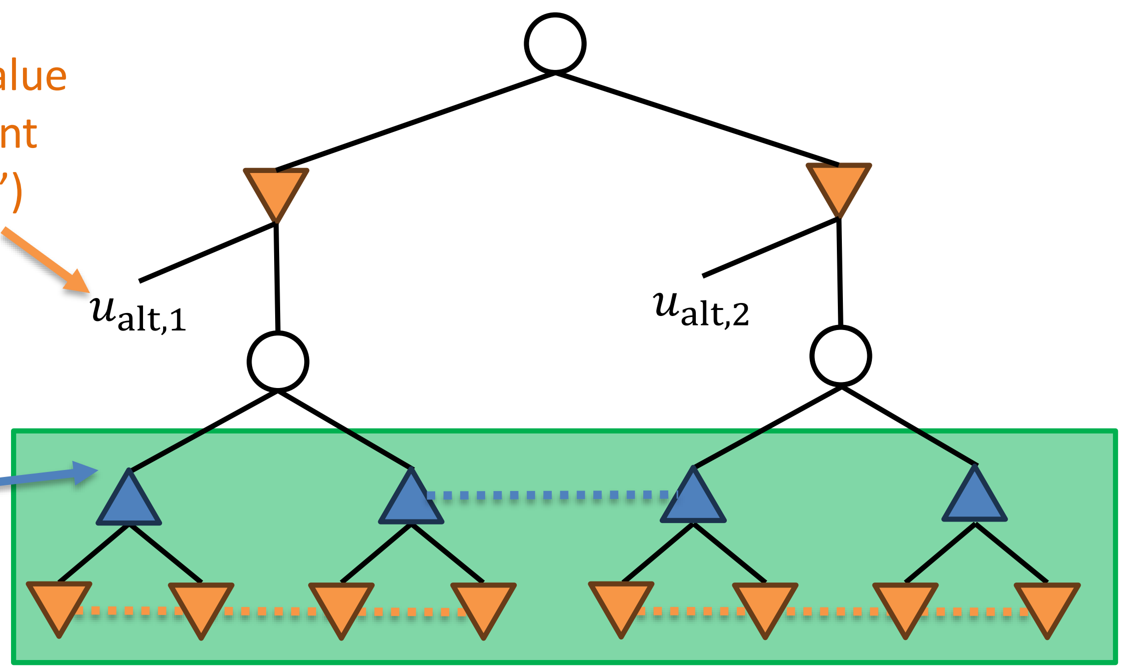
"Margin" for P2 at entry point $M_j := u_{\text{enter},j} - u_{\text{alt},j}$

= best response value against P1's new strategy

– best response value against P1's blueprint strategy

Theorem (informal)**:** Any P1 strategy that ensures that all P2 margins are nonnegative is safe

P2 best response value against P1 blueprint ("alternate value")



We are here

Burch, Johanson, Bowling (2014); Moravcik *et al.* (2016); Brown & Sandholm (2017)

Real-time subgame solving in *imperfect*-information games

"Resolve refinement":
[Burch, Johanson, Bowling 2014]

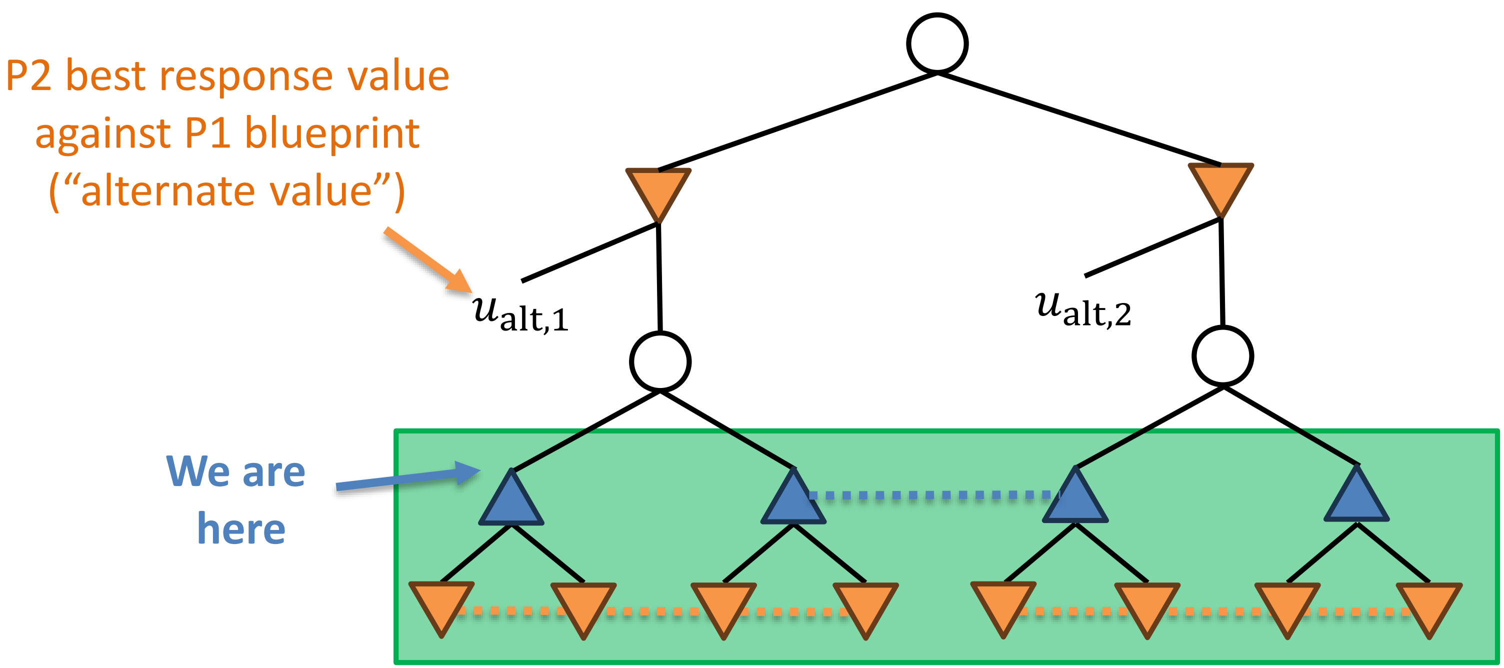
Find subgame strategy that maximizes

$$\sum_{j \in \text{Information sets}} \min\{0, M_j\}$$

"Maxmargin subgame solving":
[Moravcik *et al.*, 2016]

Find subgame strategy that maximizes

$$\min_{j \in \text{Information sets}} M_j$$



Real-time subgame solving in *imperfect*-information games

- Latest & greatest prior safe subgame solving technique: *"**reach** maxmargin subgame solving"* [Brown & Sandholm NeurIPS-17]
- **Idea:** Potentially give back the gifts the opponent has given to us on the path to any information set *j*
- Led to superhuman performance in 2-player no-limit Texas hold'em [Brown & Sandholm *Science* 2018]
- We use this idea in our new algorithm

Now moving to our new work …

So, what is the problem?

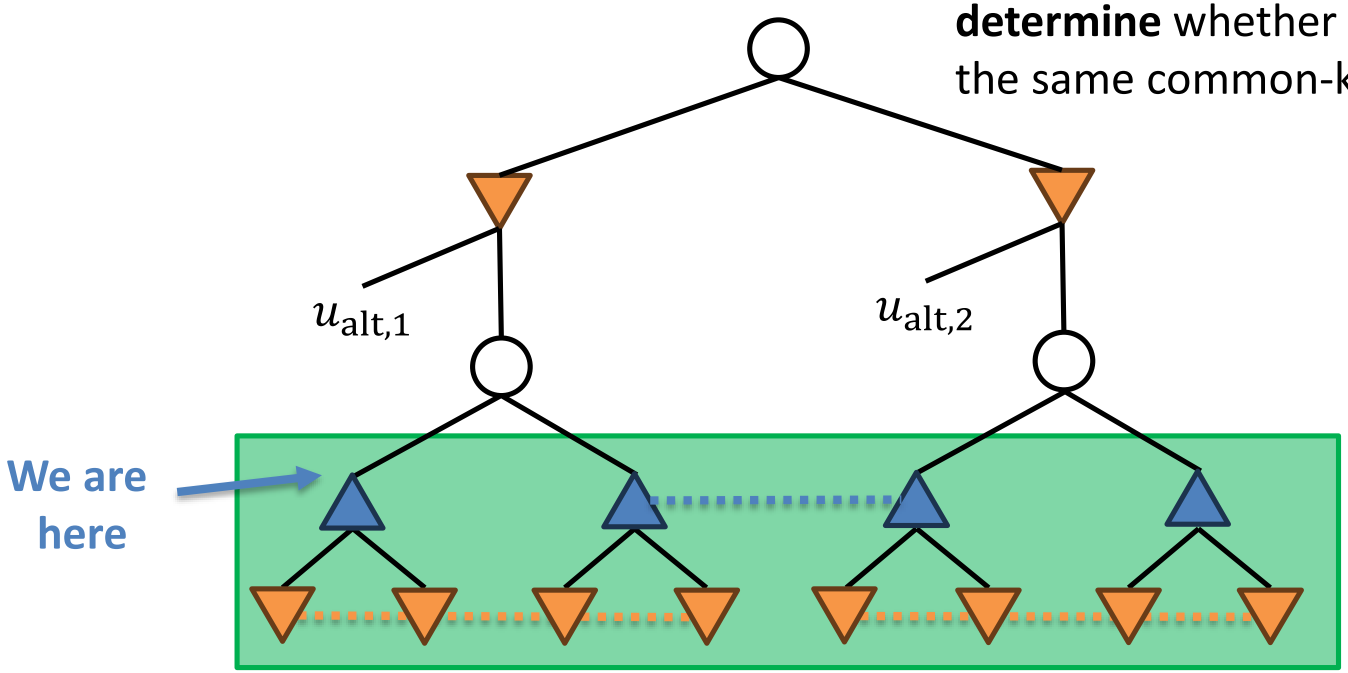
Beyond poker, common-knowledge sets can be very large!

2-player Texas hold'em: $|\mathrm{C}| < 2 \times 10^6$

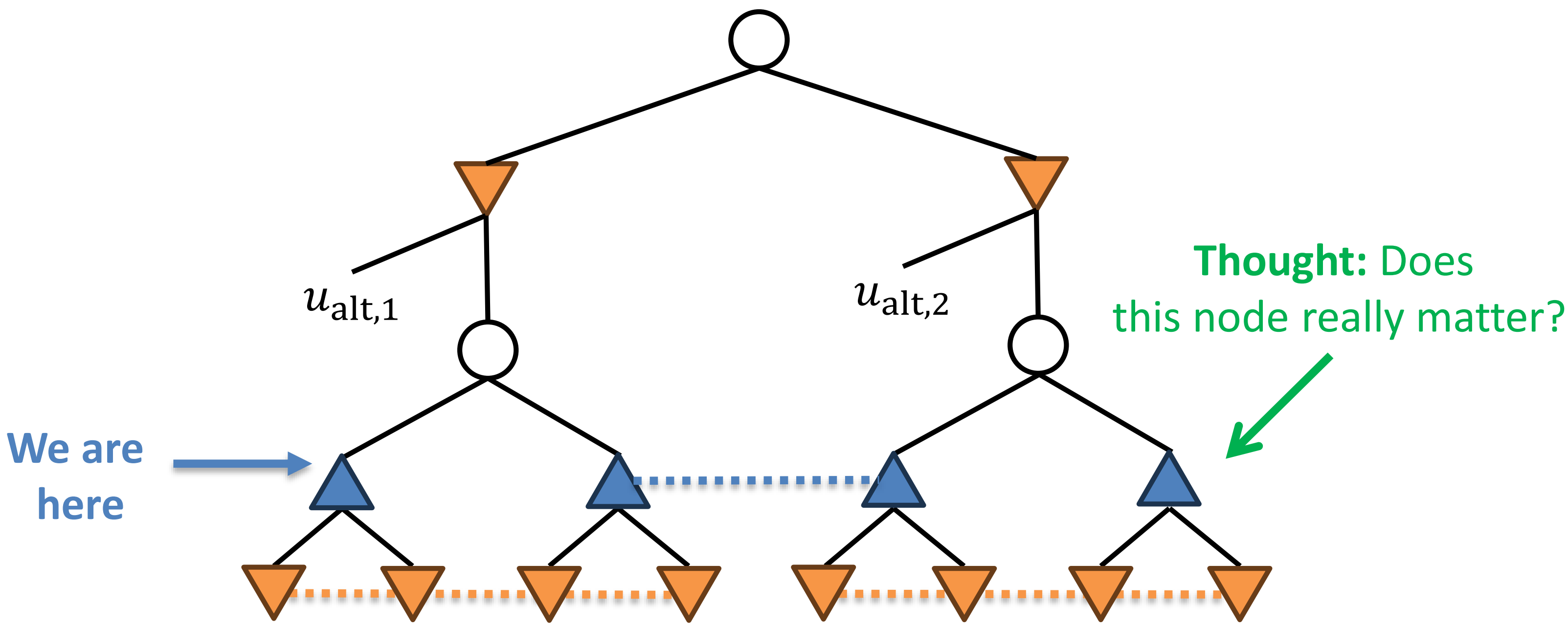
- Practical poker-specific tricks mean that, effectively, $|C| \approx 10^3$ [Johanson *et al*. *IJCAI*-11]
- Manageable in real time

Fog of War chess:

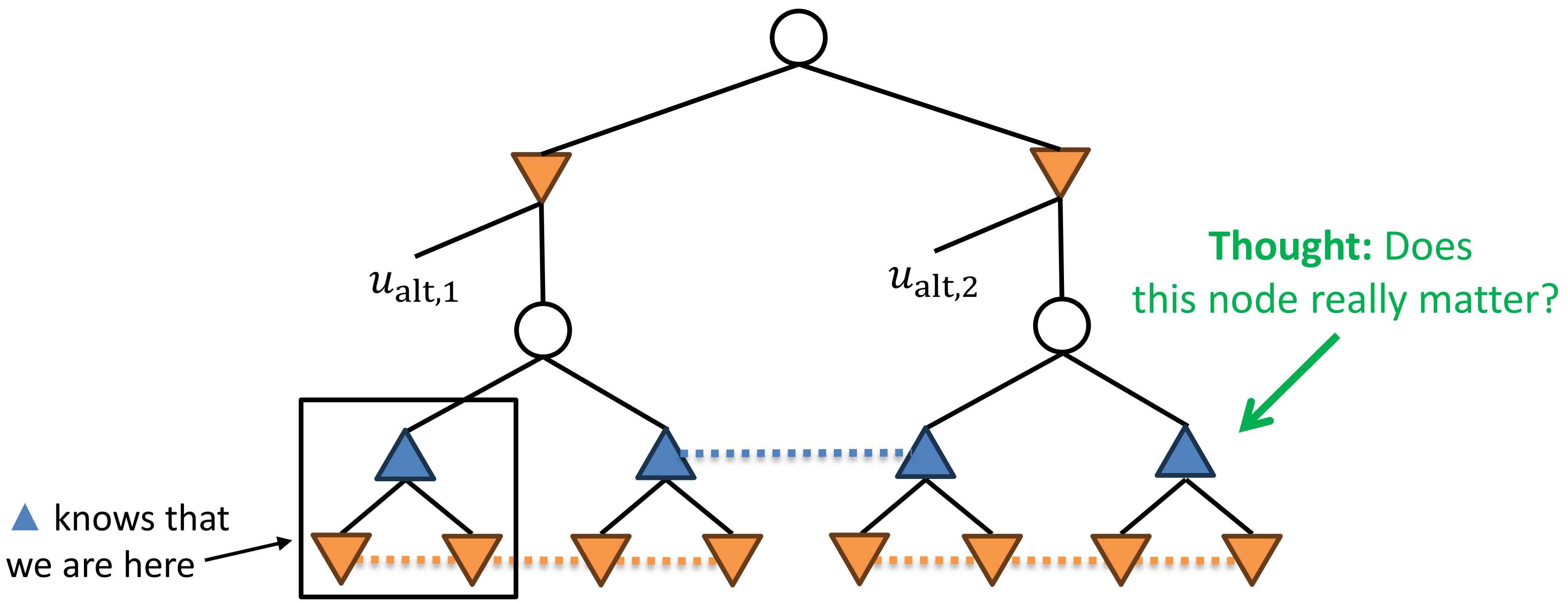
- C is often too large to store in memory, much less work with in real time
- (Not even obvious how to efficiently **determine** whether two nodes are in the same common-knowledge set)



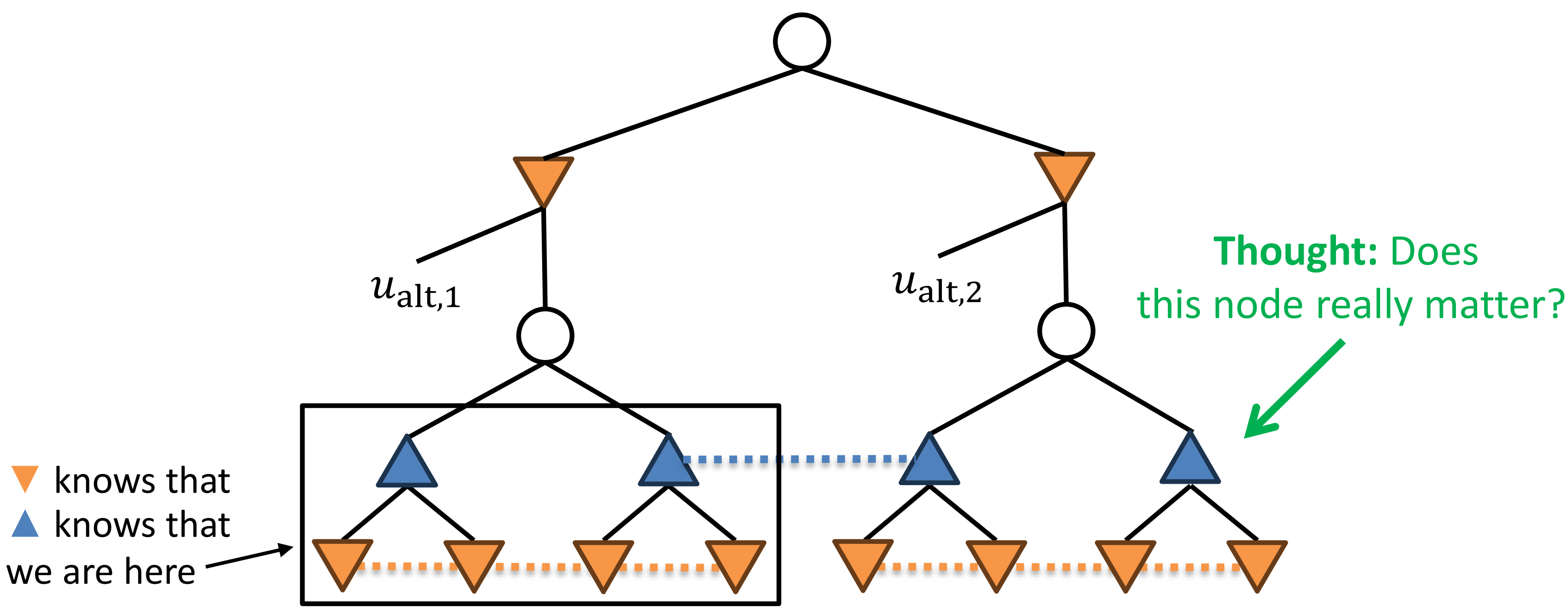
Knowledge-limited subgame solving



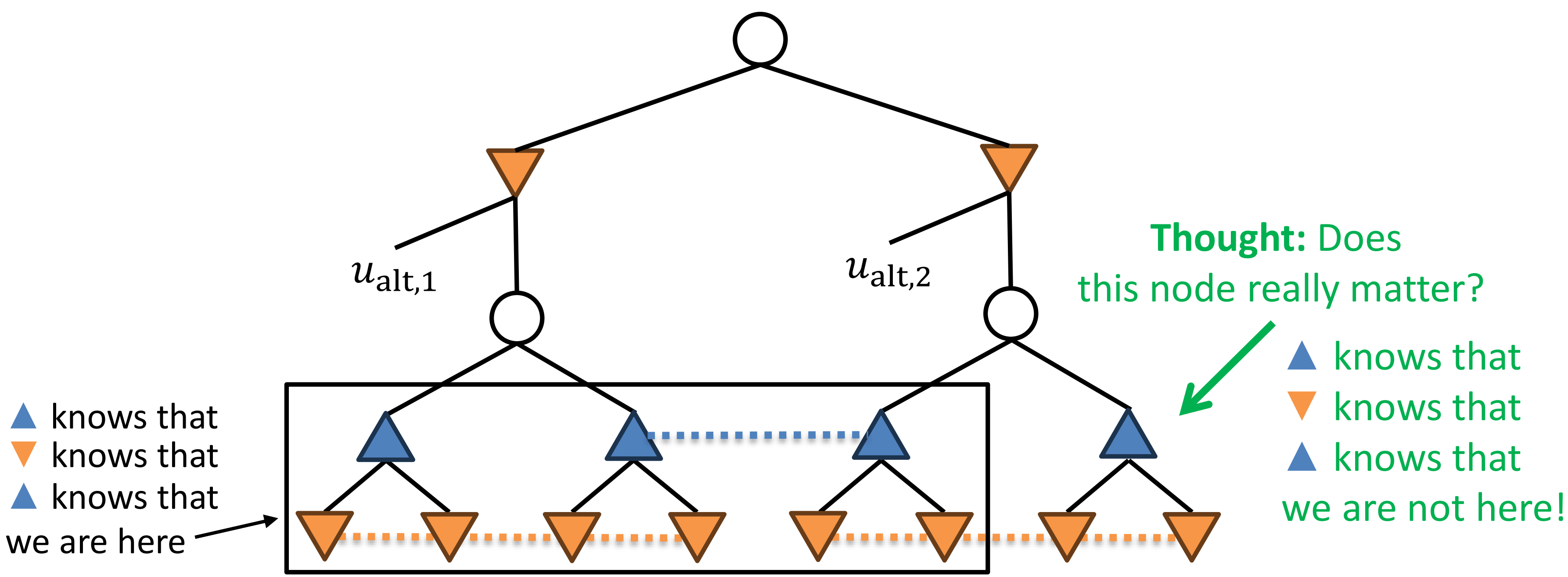
Knowledge-limited subgame solving



Knowledge-limited subgame solving



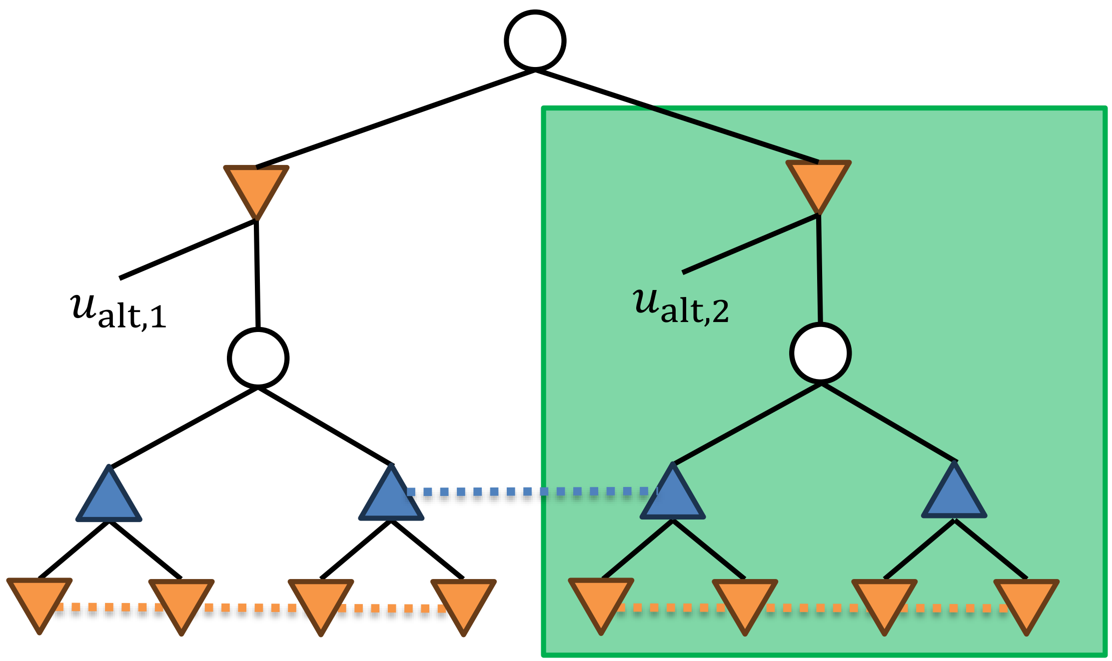
Knowledge-limited subgame solving



Knowledge-limited subgame solving

Idea: Ignore nodes "too far away" in the knowledge graph

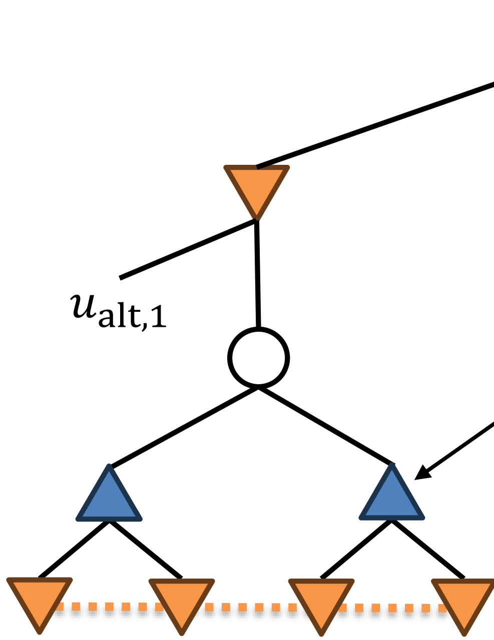
E.g., 2-KLUSS



Knowledge-limited subgame solving

Idea: Ignore nodes “too far away” in the knowledge graph

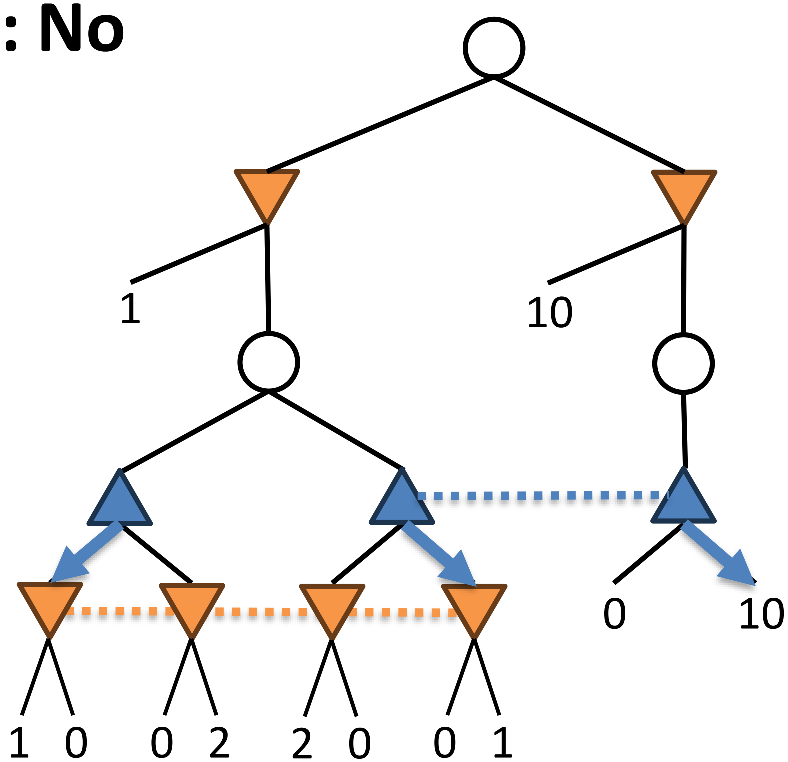
E.g., 2-KLUSS



Minor detail: Old version (*KLSS*) [Zhang & Sandholm NeurIPS-21] would freeze this node to the blueprint strategy, while our new algorithm *Knowledge Limited Unfrozen Subgame Solving (KLUSS)* doesn’t.

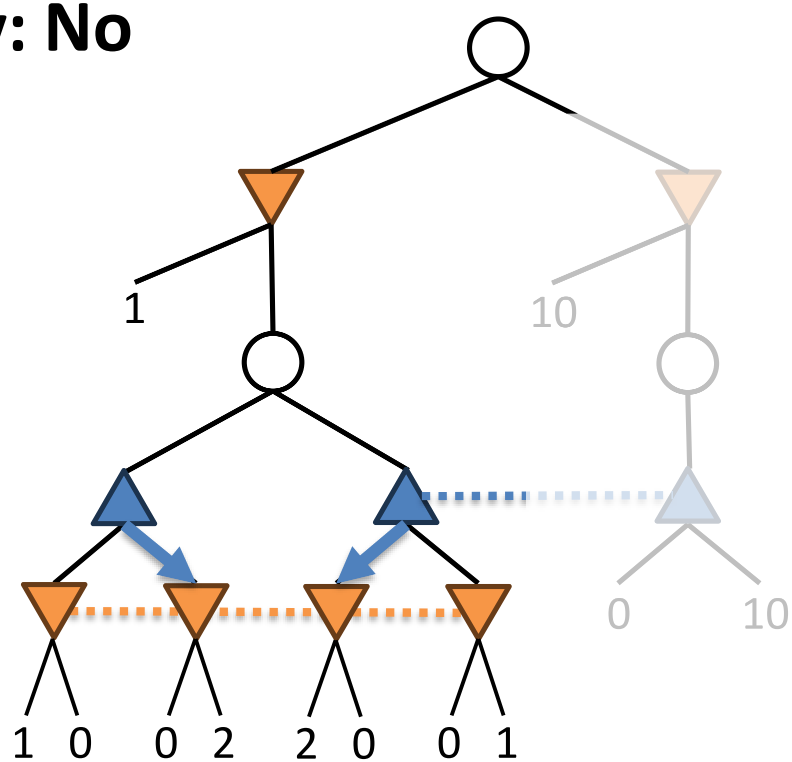
Is it safe?

- **Theory: No**



Is it safe?

- **Theory: No**



- KLSS tends to be safe in practice (based on experiments on small games) [Zhang & Sandholm NeurIPS-21].

(There are also theoretical ways to make KLSS safe [Zhang & Sandholm NeurIPS-21, Liu *et al*. ICML-23].)

Fog of War Chess

- No check or checkmate: capture the king to win ⇒ highly volatile!
- **Large game:** Game tree is almost identical to that of regular chess
- Moderate-sized information sets: size $\approx 10^5$ to 10^6 is common, but $> 10^7$ is very rare ⇒ KLUSS is practically implementable!
- Unmanageable common-knowledge sets: likely have size $> 10^{12}$ in practice and can be $> 10^{18}$; enumeration is impractical in real time

Using KLUSS, we created the first superhuman agent, *Obscuro*, for Fog of War chess!

- 97-3 against amateur humans
- **16-4 against #1-rated human** ⇒ superhuman with statistical significance!

Additional techniques in *Obscuro*

- We use a **perfect-information chess engine** (Stockfish) as an evaluation function
  - Regular chess is not so different from Fog of War chess in terms of what positions are good or bad ⇒ this evaluation function is good enough
- No blueprint ⇒ "Nested subgame solving"
  - First move: Solve from root
  - Every later move: Run 2-KLUSS on the reached subgame, with the solution from the previous move as the blueprint

Additional techniques in *Obscuro*

- Expand game tree during subgame solving using a new variant of *growing-tree CFR* [Schmid et al. 2023]: one player runs one node expansion of MCTS-like exploration algorithm; the other plays its current strategy. Swap between players every iteration.
  - Avoids unnecessarily expanding nodes that **neither player** wants to play to reach
    - MCTS or GT-CFR alone would expand these nodes
  - Still provably sound, i.e., eventually finds an equilibrium
- For any given tree in the process, we use PCFR+ [Farina, Kroer & Sandholm AAAI-21]

Additional techniques in *Obscuro*

"Resolve refinement":

Find subgame strategy that maximizes

$$\sum_j \min\{0, M_j\}$$

- ☹ Does not have any preference among the set of safe subgame strategies
- ☺ Optimistic when margins can't be made all nonnegative (e.g., due to approximation errors)

"**Maxmargin** subgame solving":

Find subgame strategy that maximizes

$$\min_j M_j$$

- ☺ "Aggressively" tries to improve upon the blueprint if possible
- ☹ Extremely pessimistic when margins cannot be made all nonnegative (e.g., due to approximation errors): focuses all attention on the entry point j with minimum margin

Our idea: Use **Maxmargin** when all margins can be made nonnegative, and **Resolve** otherwise. ⇒ "Best of both worlds" behavior!

Additional techniques in *Obscuro*

- In **Resolve** (when margin < 0), the standard objective is

$$\sum_{j\in[m]} \min\{0, M_j\}$$

  but any weighted sum with all positive weights can be used, with the same guarantees! We use

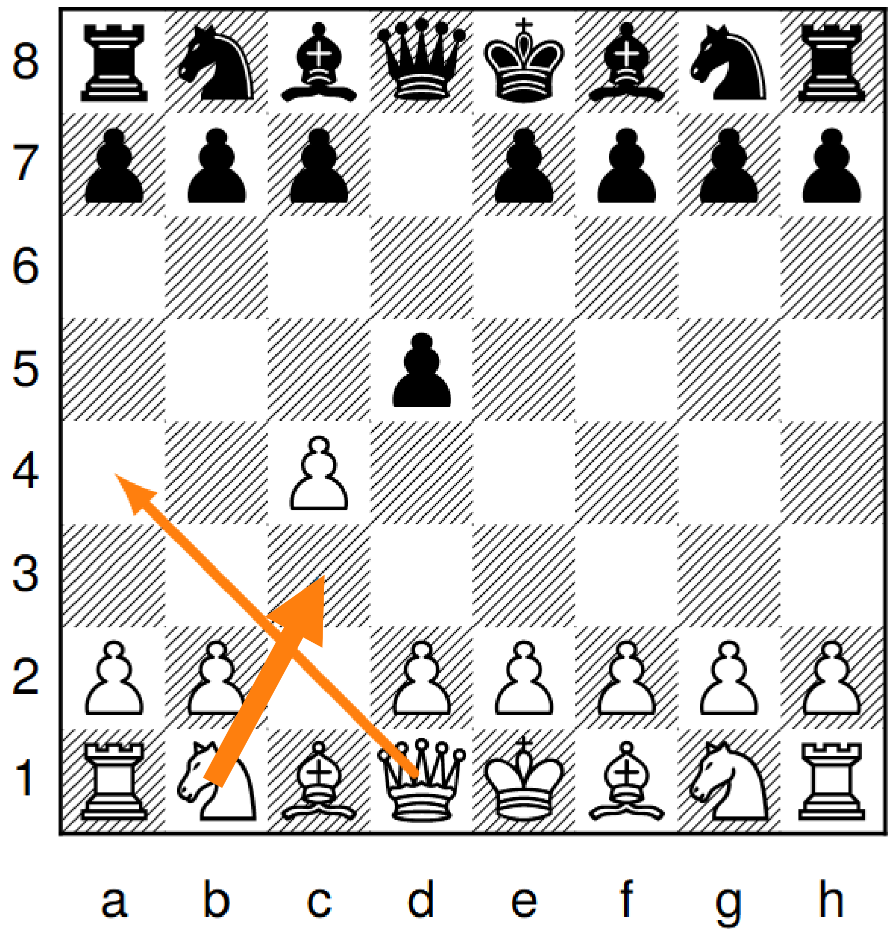
$$\sum_{j\in[m]} \frac{1}{2}\left(\frac{1}{m} + \frac{y_j}{\sum_j y_j}\right) \min\{0, M_j\}$$

- "increase weight on information sets that are more likely to be reached, while maintaining that all weights are positive"

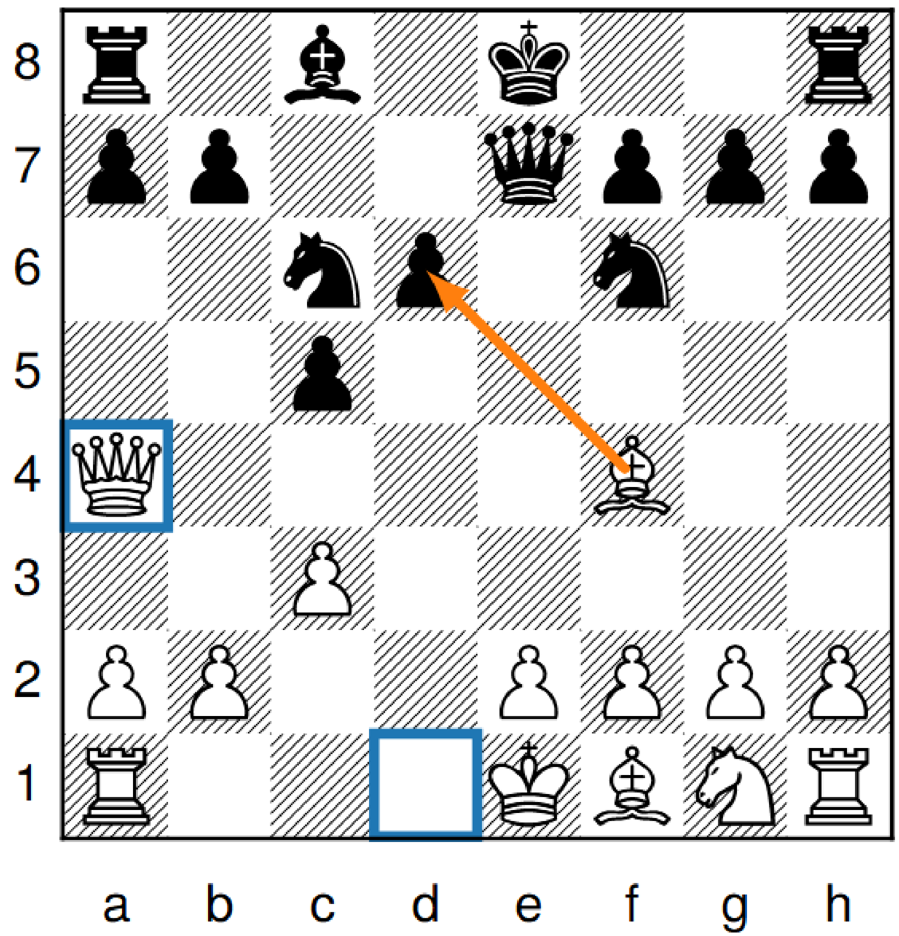
Additional techniques in *Obscuro*

- **~purify** the strategy: Remove all but the *k* most likely moves ($k = 1$ when margin < 0, and $k = 3$ when margin $\geq$ 0)
  - Purification removes the danger of bad actions ending up in the support with small probability
  - But *some* mixing is required for good play
    $\Rightarrow$ allow $k > 1$ at least when it's "provably safe" (margin $\geq$ 0)

Examples of *Obscuro's* behavior: *Smart randomization*



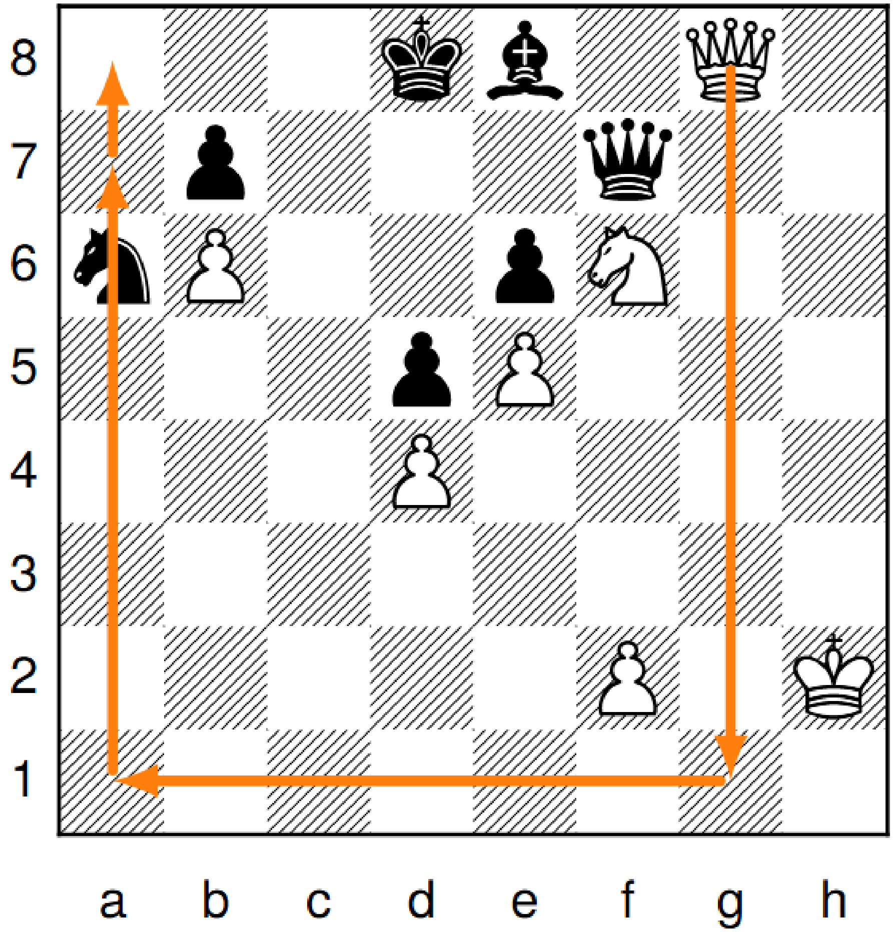
Examples of *Obscuro's* behavior: *Bluffing*



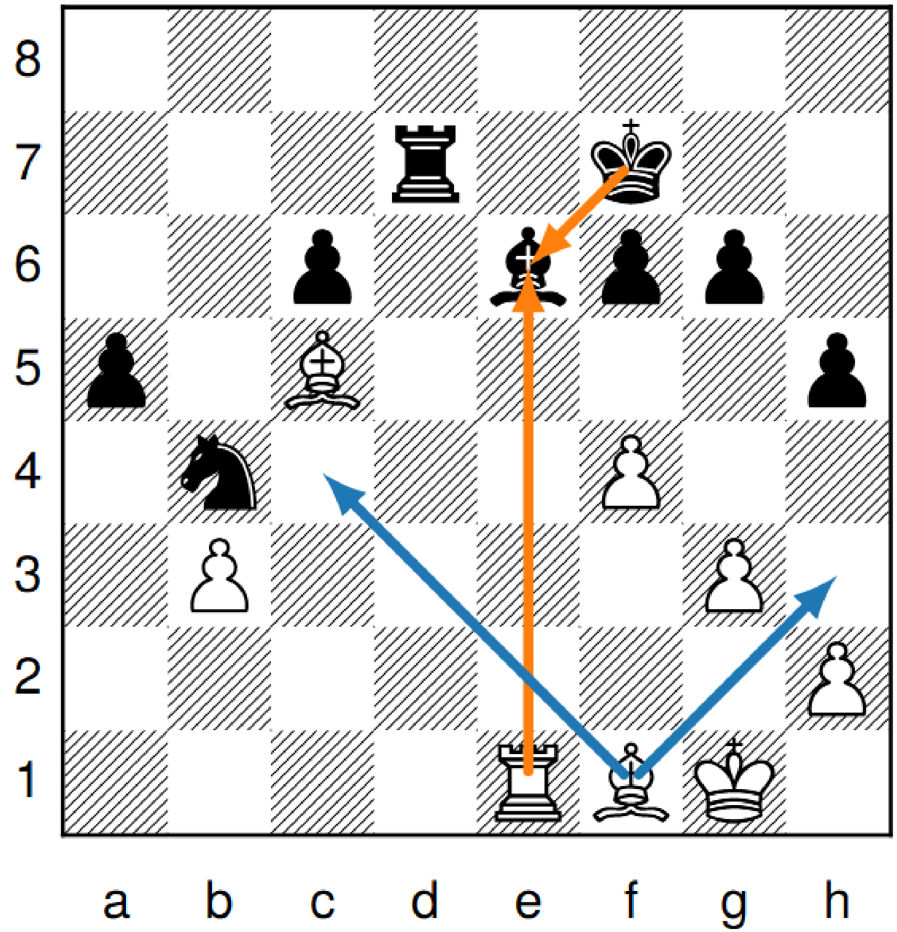
Examples of *Obscuro's* behavior:

Risk-taking when losing

(from play against #1 human)



Examples of *Obscuro's* behavior:
A tactic that relies on randomization



Conclusions

- Fog of War chess is now the
  - largest (measured by amount of imperfect information) turn-based game in which superhuman performance has been achieved, and
  - the largest game in which imperfect-information search techniques have been successfully applied
- Knowledge limited subgame solving avoids the need to generate common knowledge sets
- We introduced many additional techniques

Future research

- Other applications of alternating between MCTS-like exploration & playing current strategy
- Practical, provably safe variants of KLUSS
- Integration with other techniques
  - Using blueprint from deep RL
  - Using continuation strategies [Brown, Sandholm & Amos NeurIPS-18; Brown & Sandholm *Science* 2019]
  - Opponent modeling