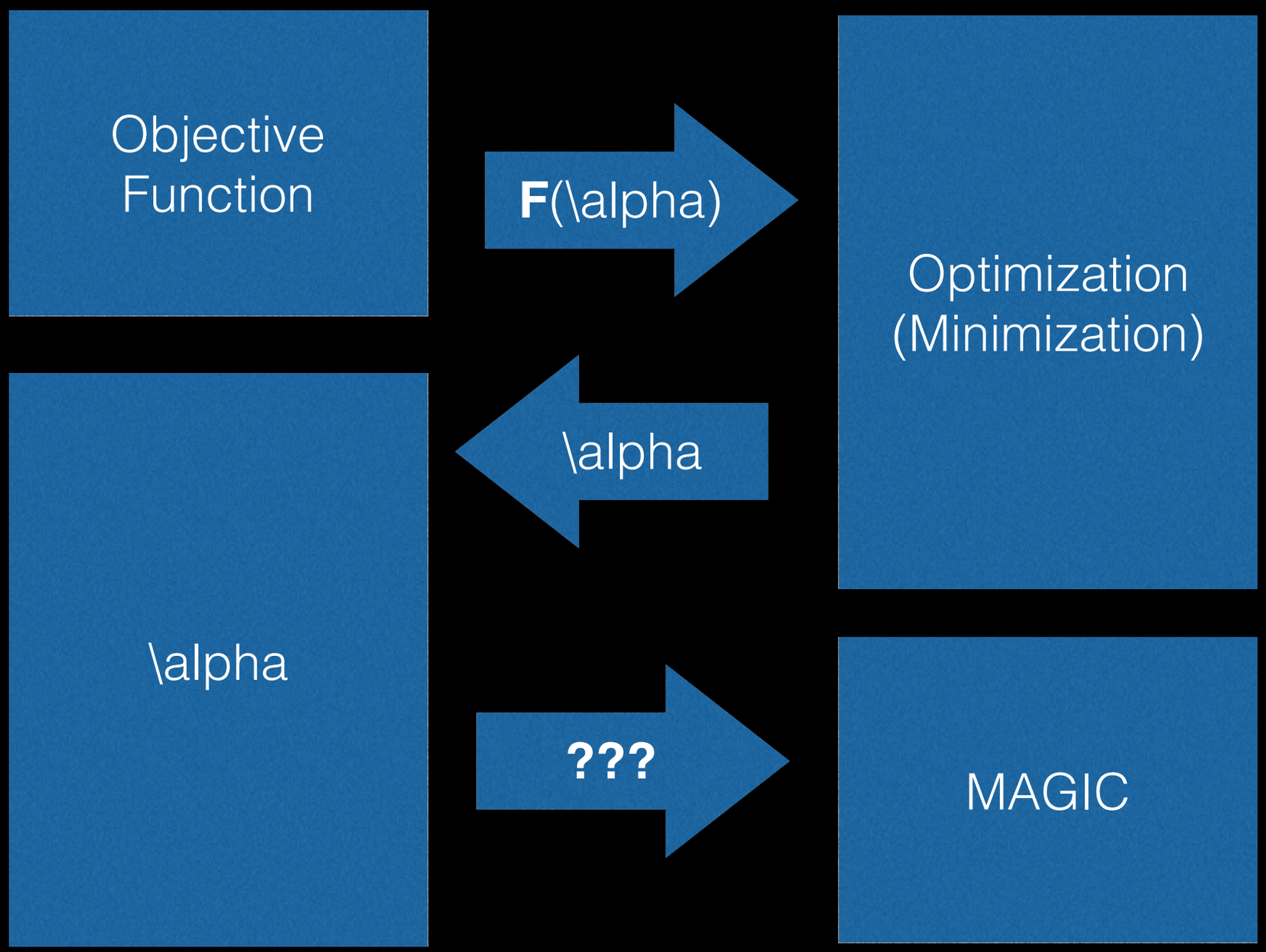
Optimizing Subjective Functions Subject to Complaints

Eston Schweickart, Alan Turing*, Paul Erdős*,
Sir Isaac Newton*

*not actually an author

Machine Learning

Machine Learning



What is **F**?

- Whatever you can quantify!
- Cuteness of cats
- Humor of jokes
- How well an image matches a model of Shia LaBeouf's left ear

What if you can't quantify something?

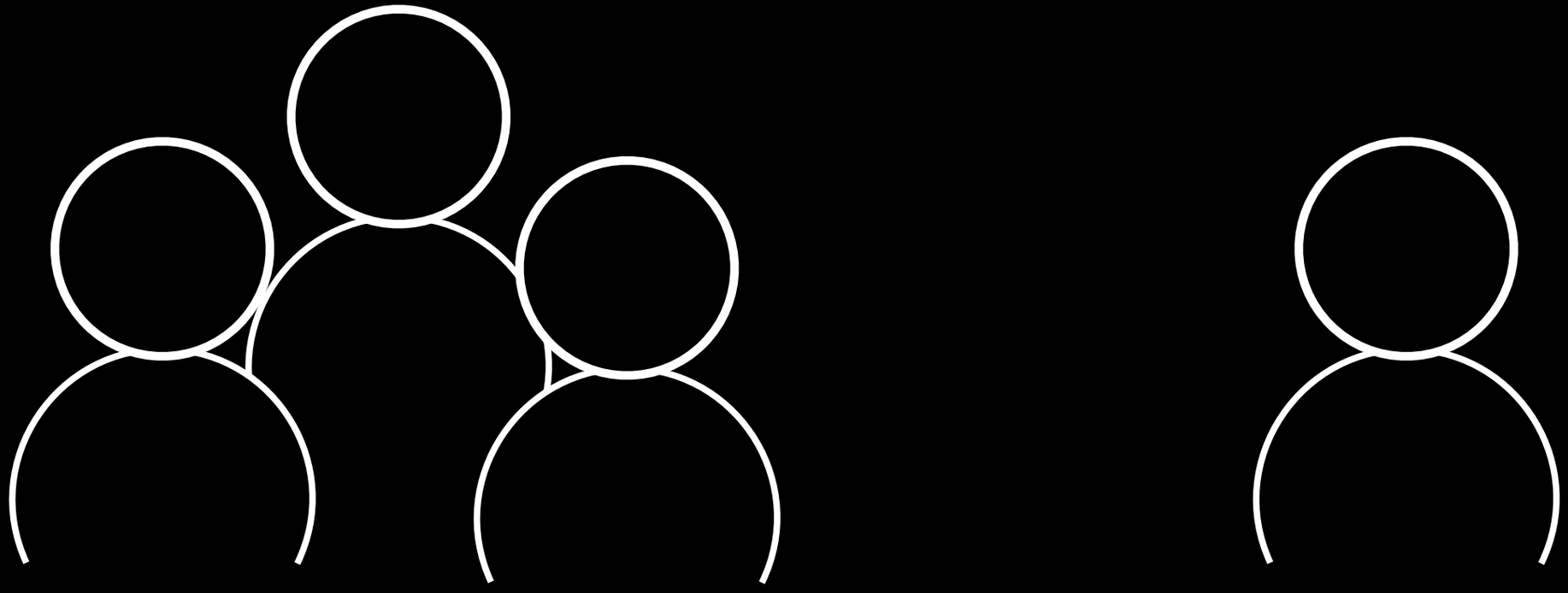
- Standard answer: "try harder, dammit"
- Our answer: "give up, you'll never succeed"

- Examples:
  - Goodness of cats
  - Dankness of memes

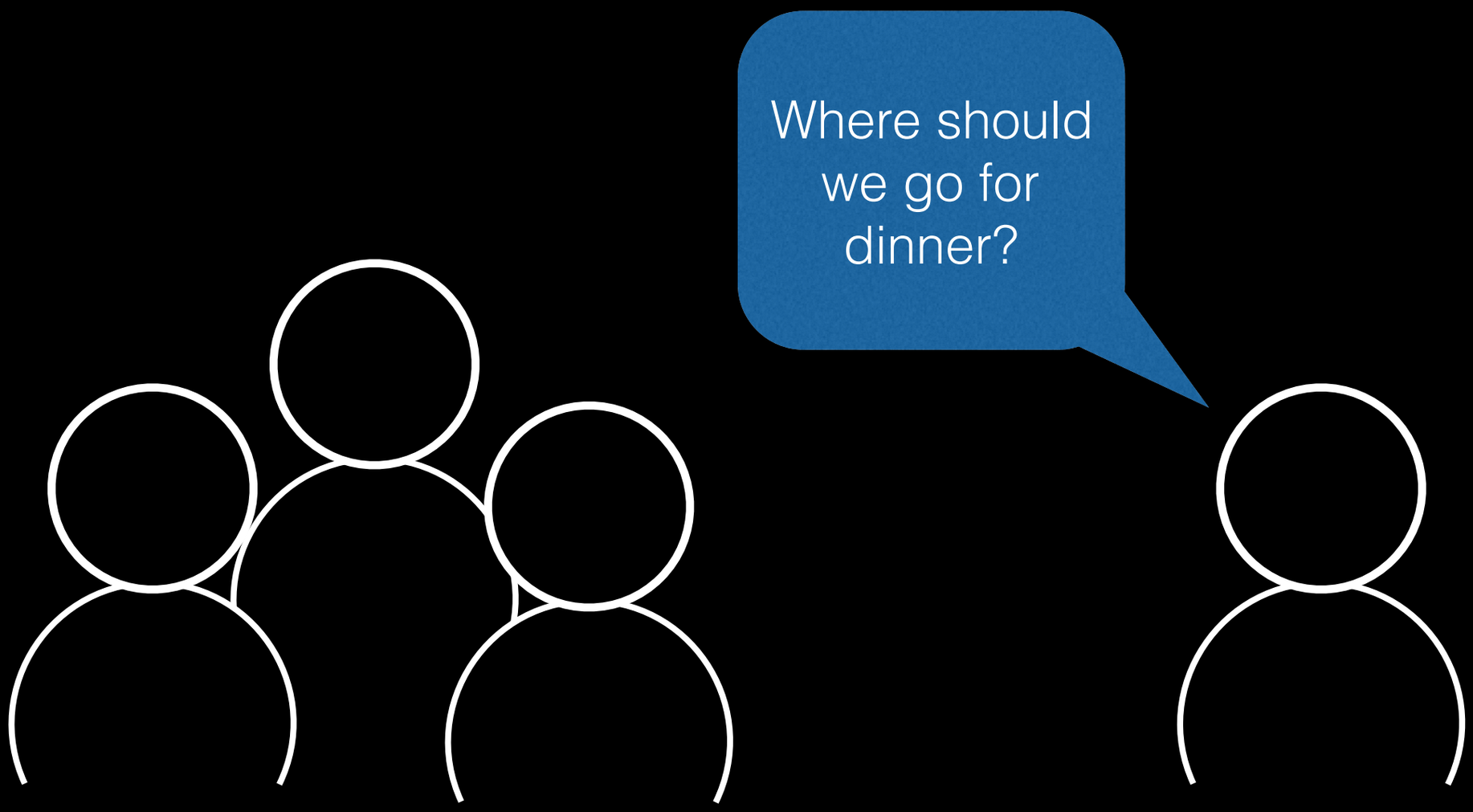
Subjective functions

- Similar to objective functions, but **F** does not give quantifiable output
- Often nondeterministic, poorly behaved
- Some example output:
  - “Yeah, okay”
  - “Sounds lame”
  - “Gnarly!”
- Forget about trying to take a derivative

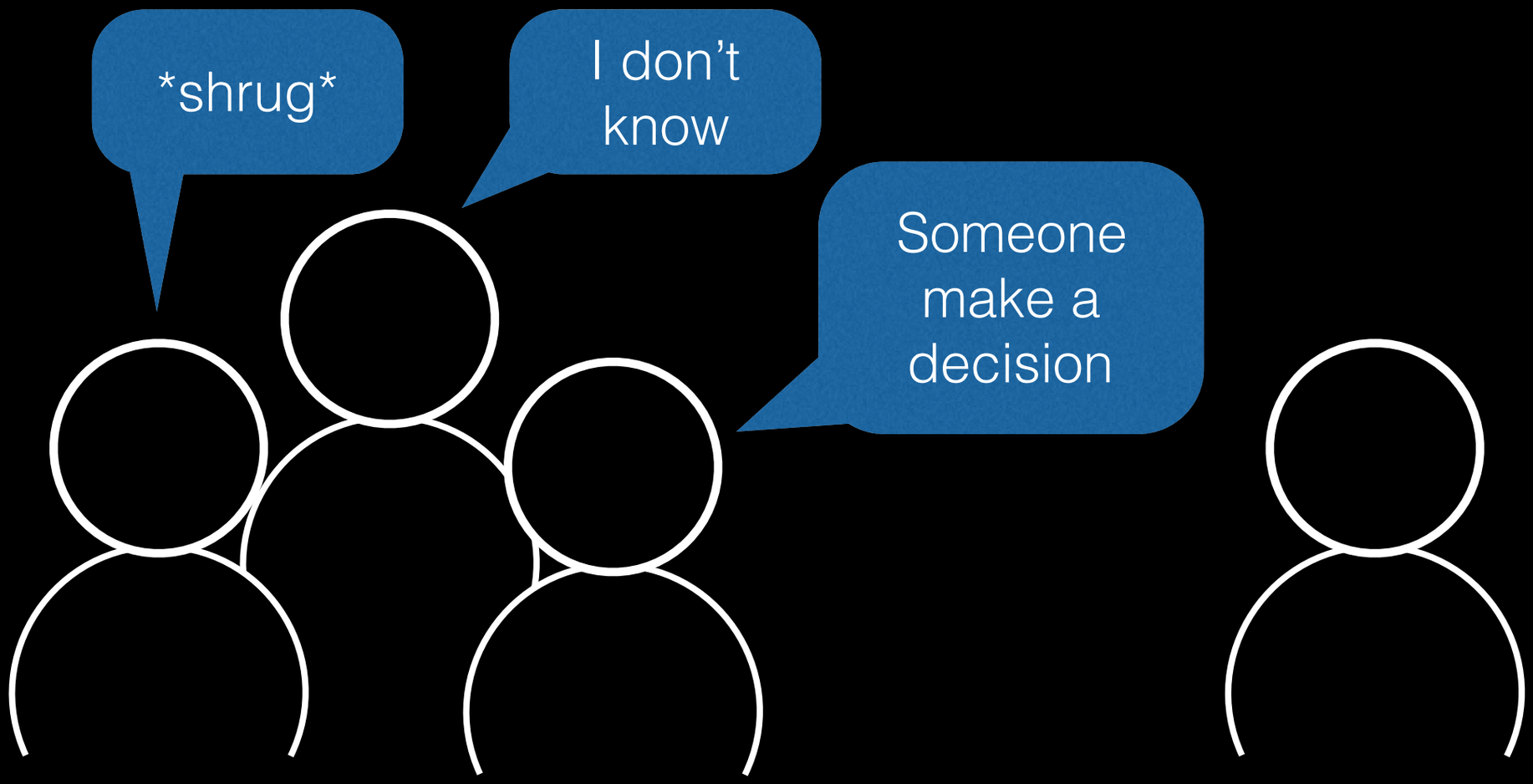
Example problem: The dining PhD student problem



Example problem: The dining PhD student problem

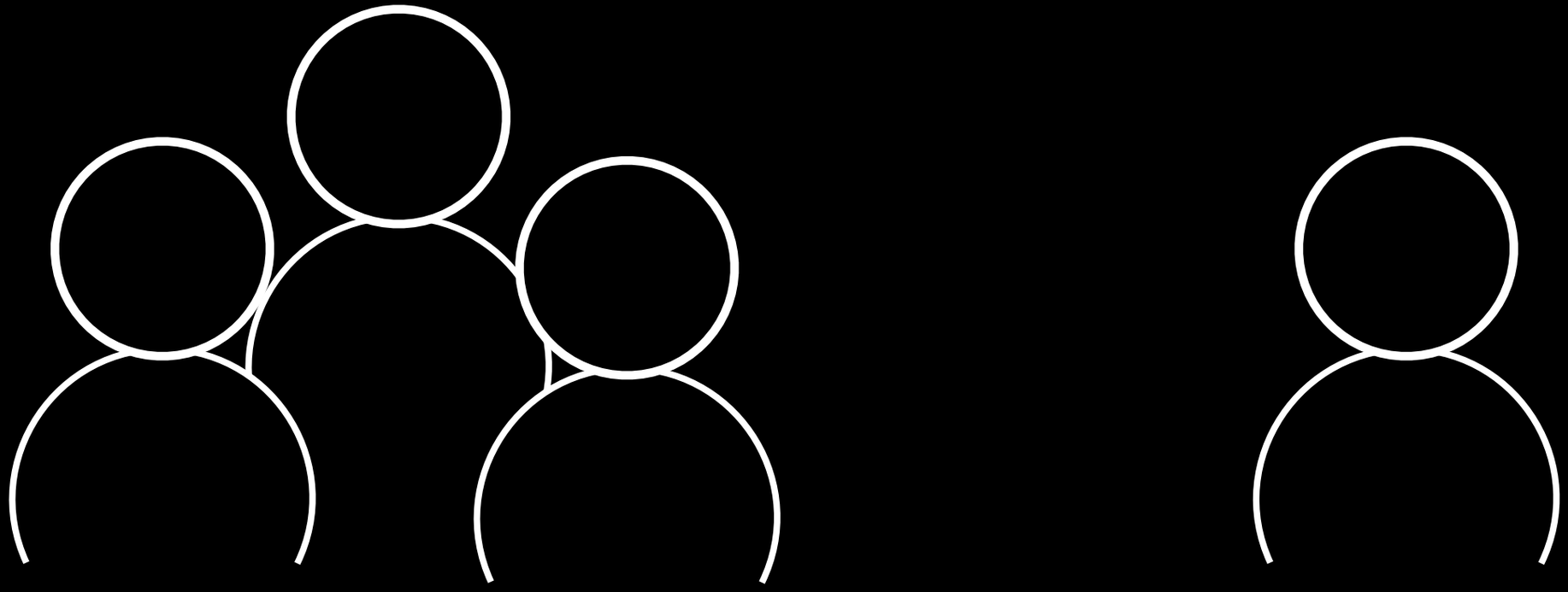


Example problem: The dining PhD student problem



Example problem: The dining PhD student problem

...

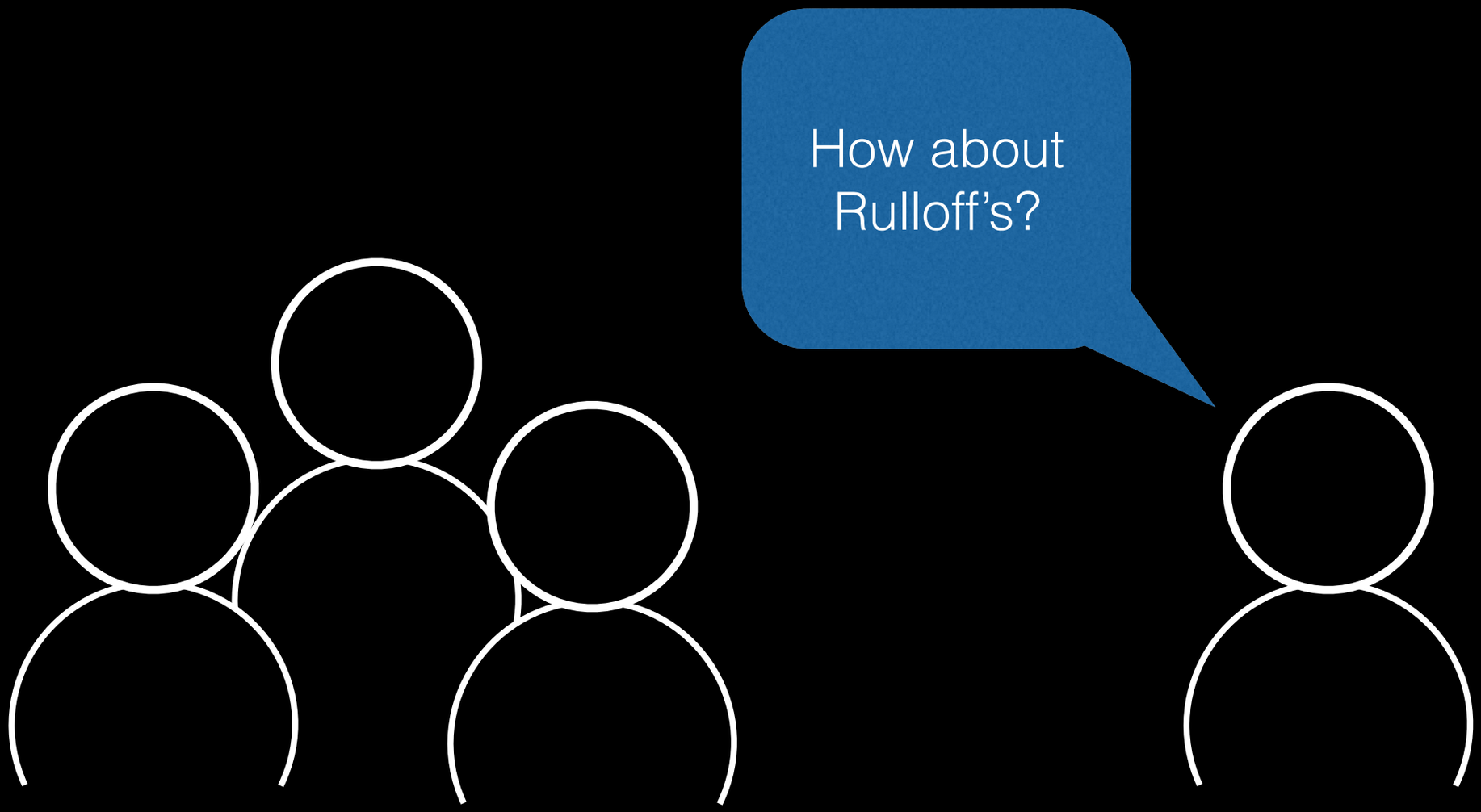


Example problem: The dining PhD student problem

Optimizing Subjective Functions

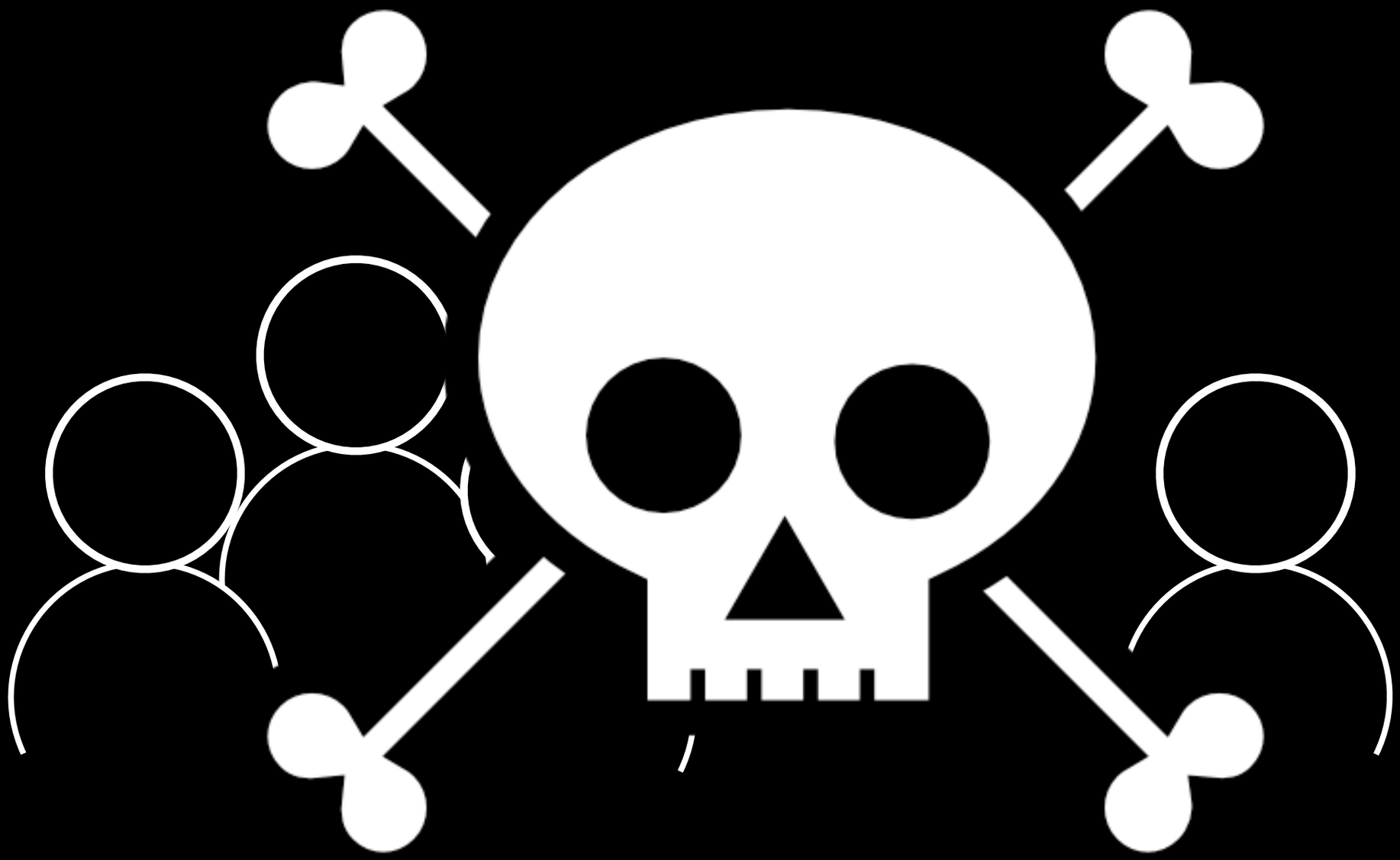
- Naive approach: just choose any \alpha
    - If everything's subjective, there's no way to tell optimality
    - Assert optimality and call it a day
- Problem: unsatisfactory in the presence of complaints

Example problem: The dining PhD student problem



Example problem: The dining PhD student problem

Example problem: The dining PhD student problem



Complaints

- Like constraints, but not expressible mathematically
- Typically extremely vague and unhelpful
- Example complaints:
  - “I don’t like it, but I don’t know why”
  - “Needs more…pizzazz!”
  - “I’ll know the right answer when I see it”

A Slightly Improved Algorithm

```
for \alpha in \Alpha {
  try F(\alpha)
  if no complaints encountered {
    return \alpha
  }
}
ragequit
```

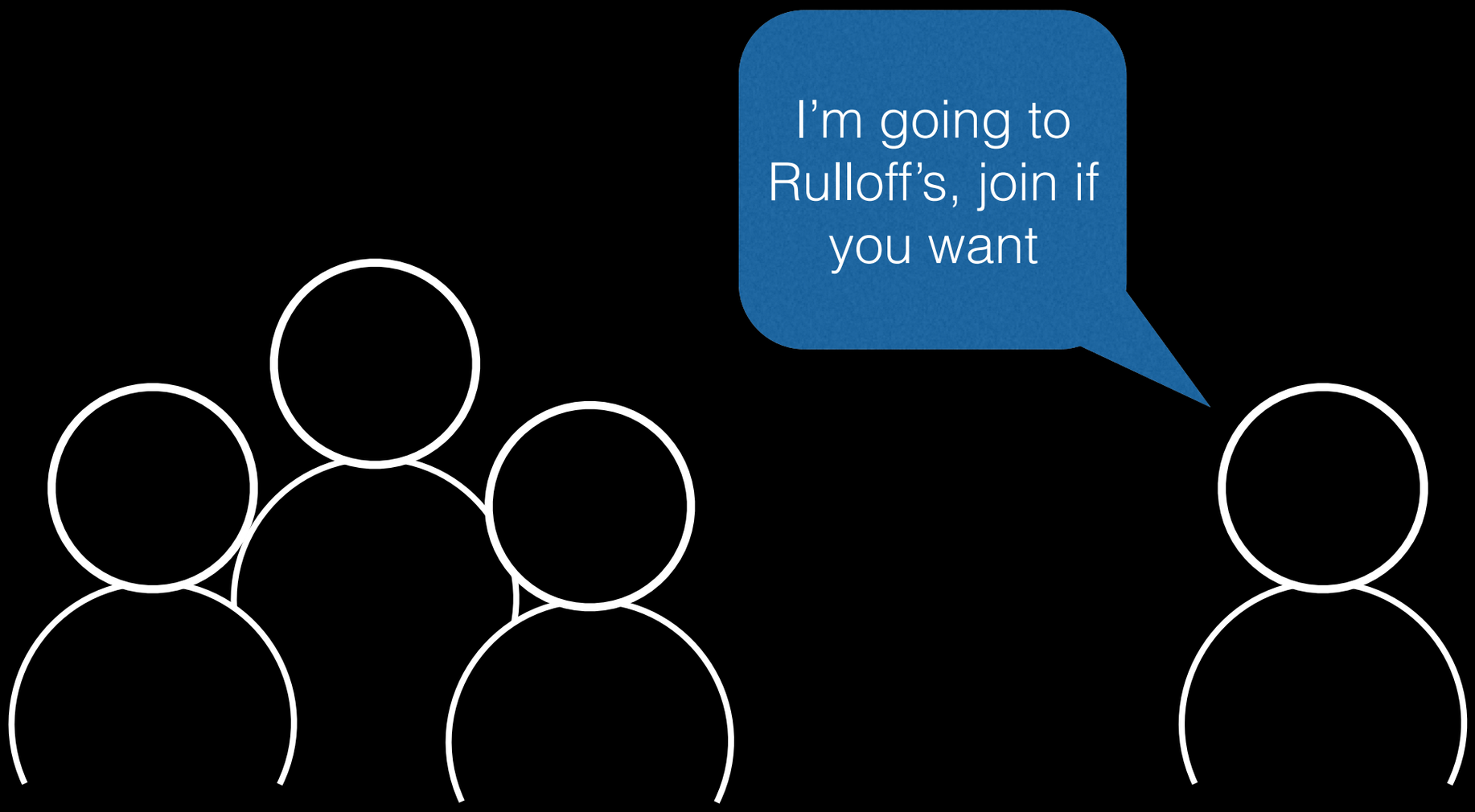
A Slightly More Improved Algorithm

```
for \alpha in \Alpha {
  try F(\alpha)
  if no complaints encountered {
    return \alpha
  }
}
return random \alpha in \Alpha
```

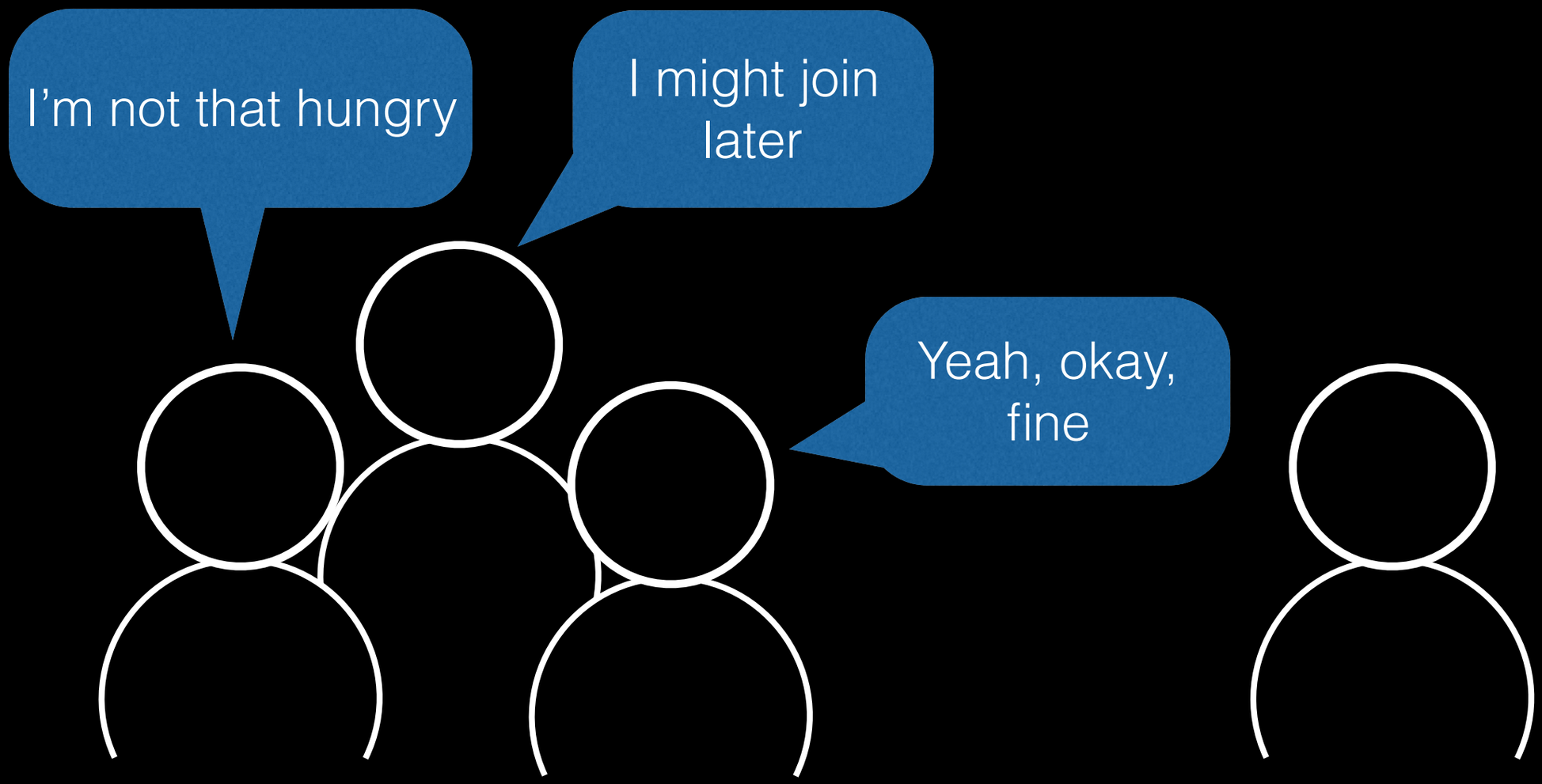
A Slightly Even More Improved Algorithm

```
for \alpha in \Alpha {
  try F(\alpha)
  if no complaints encountered {
    return \alpha
  } else if frustration > \beta {
    return \alpha, dammit
  }
}
return random \alpha in \Alpha
```

Example problem: The dining PhD student problem



Example problem: The dining PhD student problem

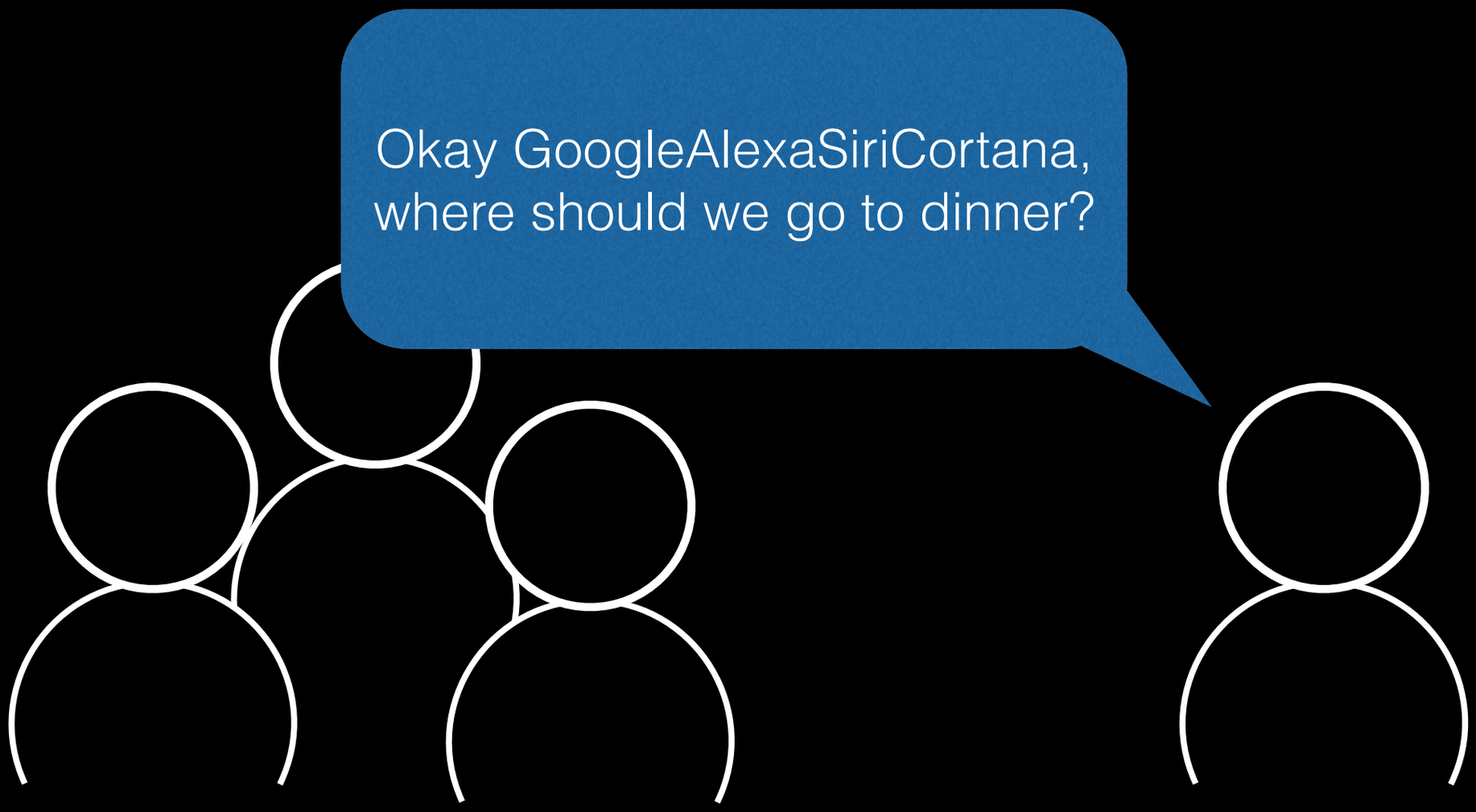


Example problem: The dining PhD student problem

Our Method

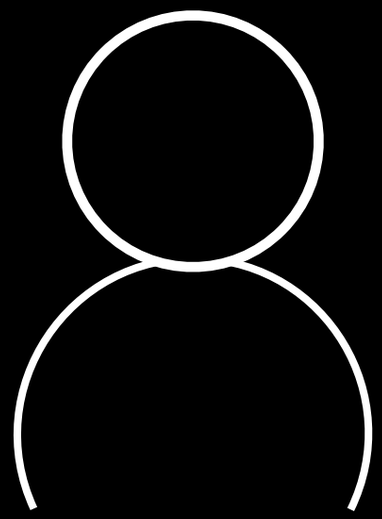
- Ask an Oracle
- Obvious in hindsight
- Simple, hassle free
- Avoids complaints altogether

Example problem: The dining PhD student problem

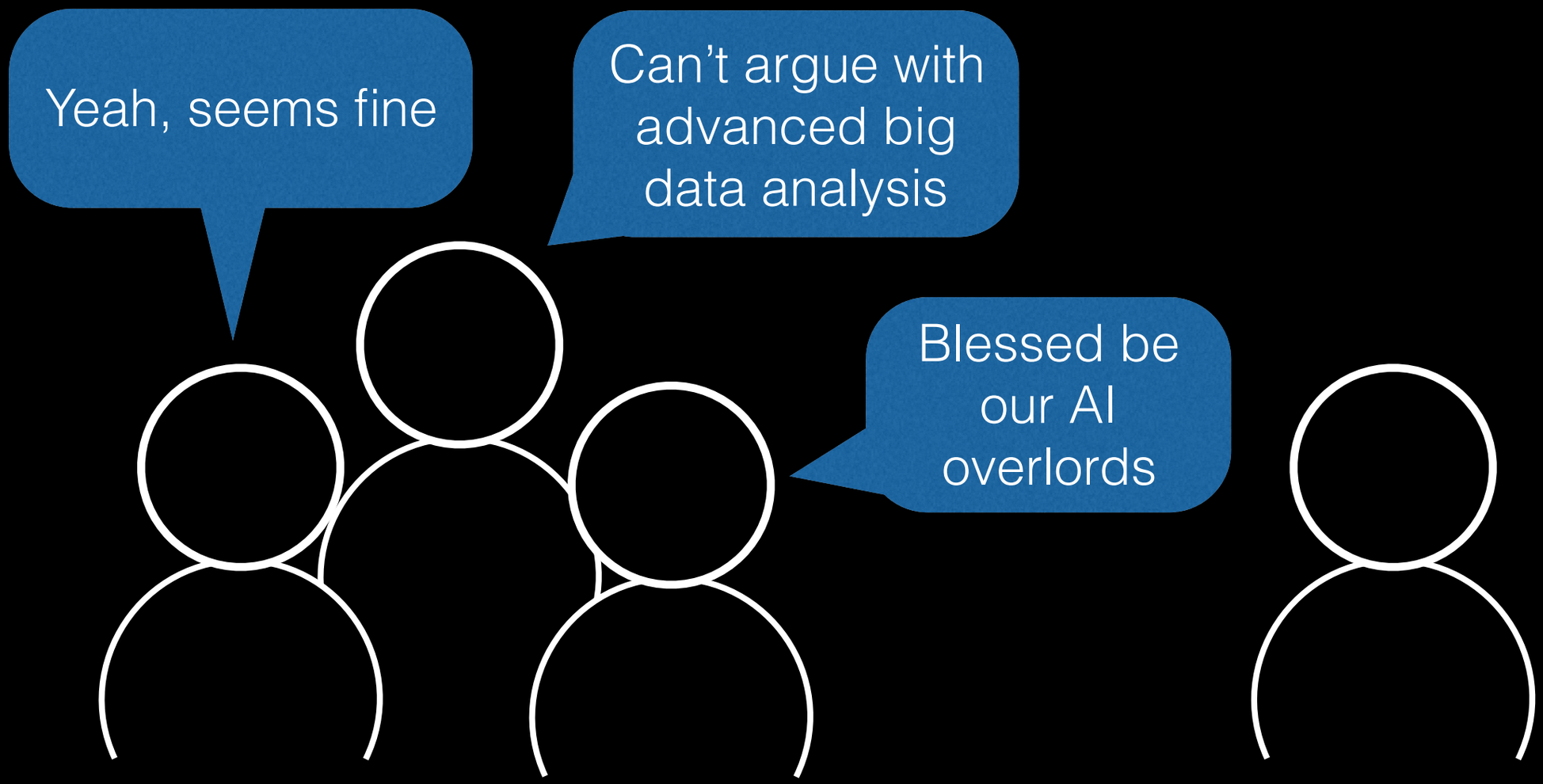


Example problem: The dining PhD student problem

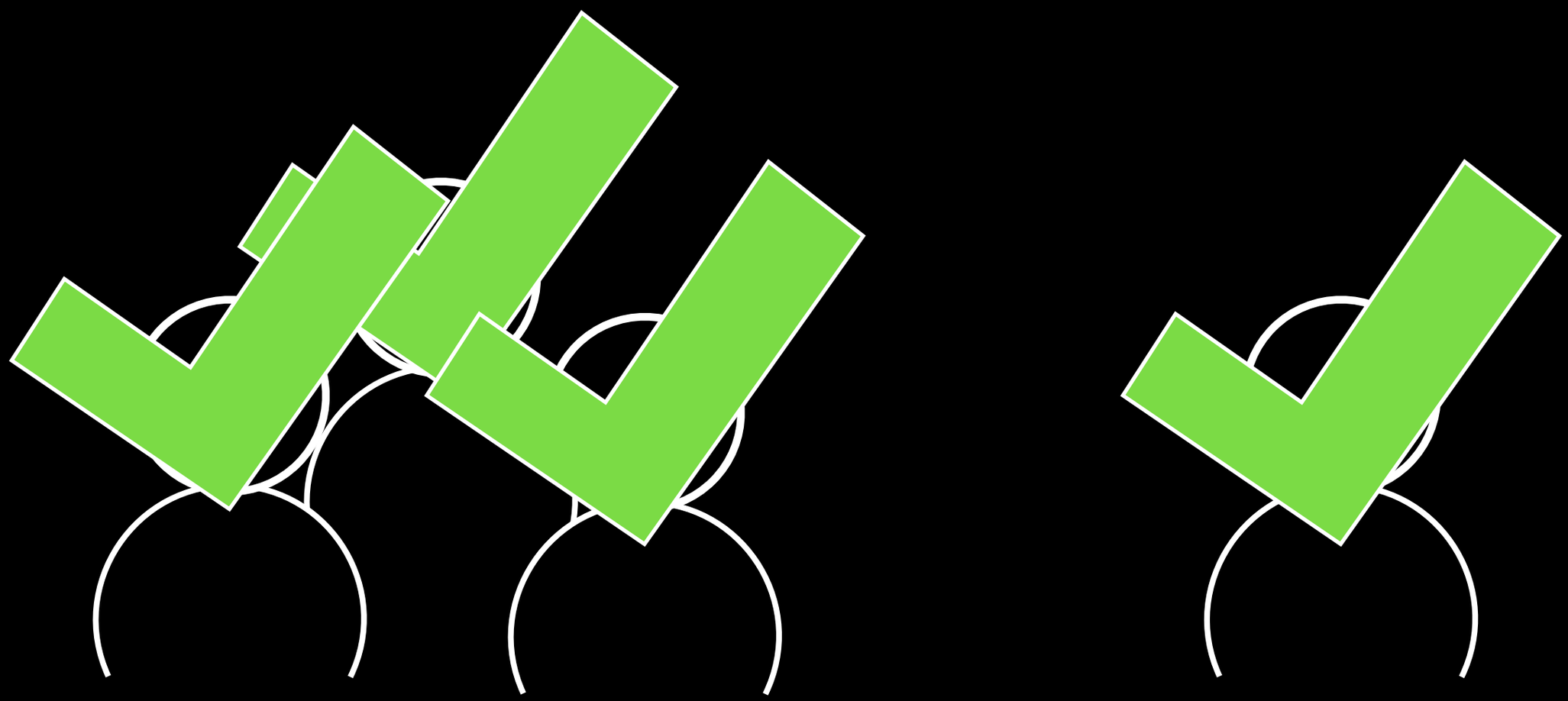
Based on my calculations, it is optimal to go to Rulloff's



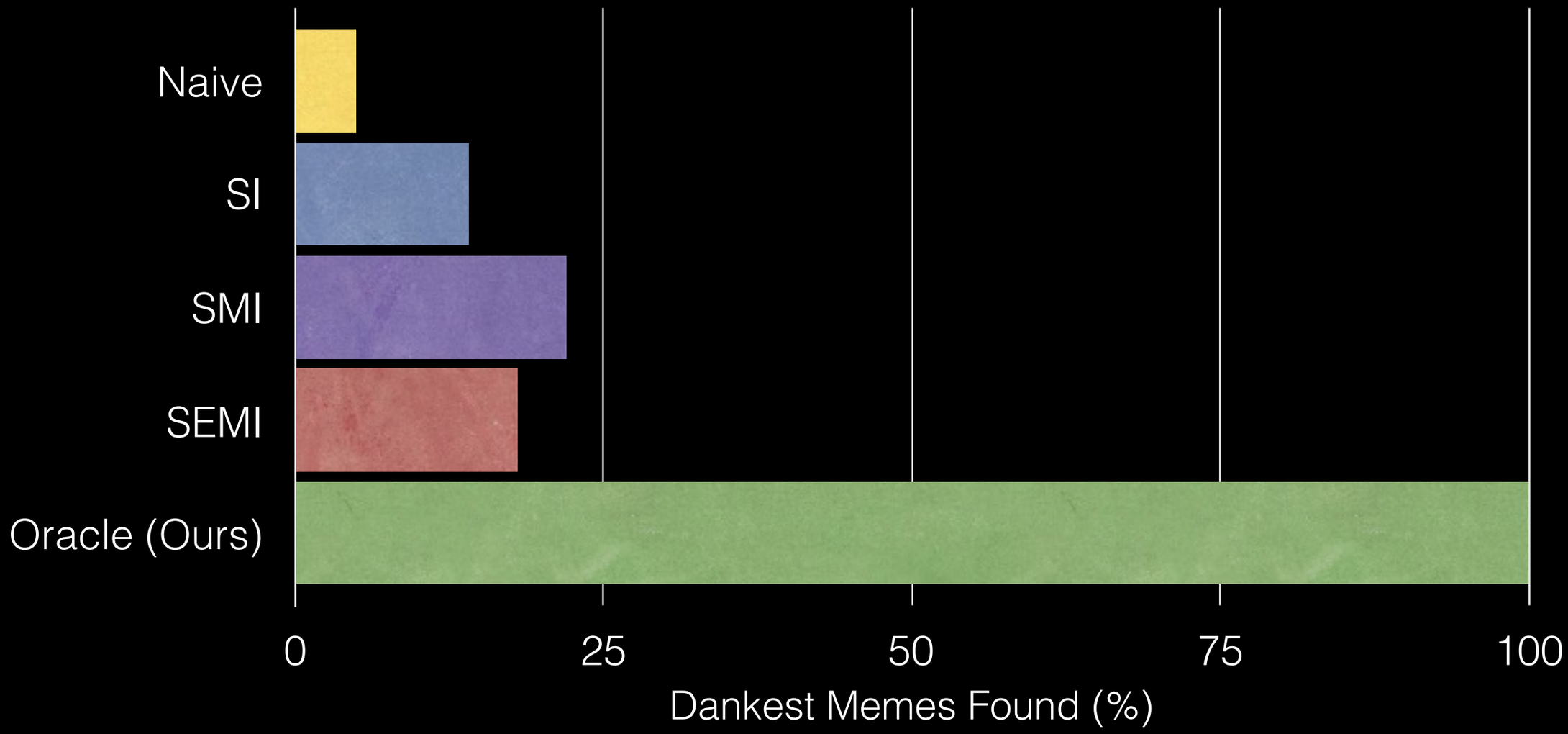
Example problem: The dining PhD student problem



Example problem: The dining PhD student problem



Results


Naive
SI
SMI
SEMI
Oracle (Ours)
0
25
50
75
100
Dankest Memes Found (%)

Conclusions

- Turns out this was super easy all along
- Not sure why others haven’t tried this approach yet
- Significant implications in complexity theory

Acknowledgements

- Google
- Apple
- Amazon
- Microsoft
- Facebook
- Netflix
- Oracle
- Our AI overlords (blessed be)
- The lizardpeople who secretly run the world
- OBEY
- Blockchain

Questions?

The Best Cat