On How to Learn, Do and Write Theory

Spring 2026
Brown University

D Ellis Hershkowitz

How Theory is (Often) Taught

How Theory is (Often) Taught

1. Here is **problem X**.
2. Here is **method A**.
3. Therefore **solution**

How to Solve Theory Problems (?)

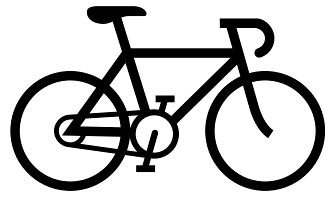
1. Write down the **problem X**.
2. Think *real* hard.
3. Write down the **solution**.

≈Murray Gell-Mann

How Theory is Done

Simplification

Active



How Theory Problems are Solved

1. Isolate a toy **model case x** of major **problem X**.
2. Solve **model case x** using **method A**.
3. Try using **method A** to solve the full **problem X**.
4. This does not succeed but **method A** can be extended to **model cases x′ and x′′**.
5. Eventually, it is realized that **method A** relies crucially on a **property P** being true which holds for **model cases x, x′ and x′′**.
6. Conjecture that **property P** is true for all instances of **problem X**.
7. Discover a family f **counterexamples y, y′, y′′**,… to this conjecture.
8. Take the simplest **counterexample y** in this family, and try to solve **problem X** for this special case. Meanwhile, try to see whether **method A** can work without **property P**
9. Discover several counterexamples in which **method A** fails, in which the cause of failure can be definitely traced back to **property P**. Abandon efforts to modify **method A**.
10. Realize that **counterexample y** is related to a **problem Z** in another field.

•••

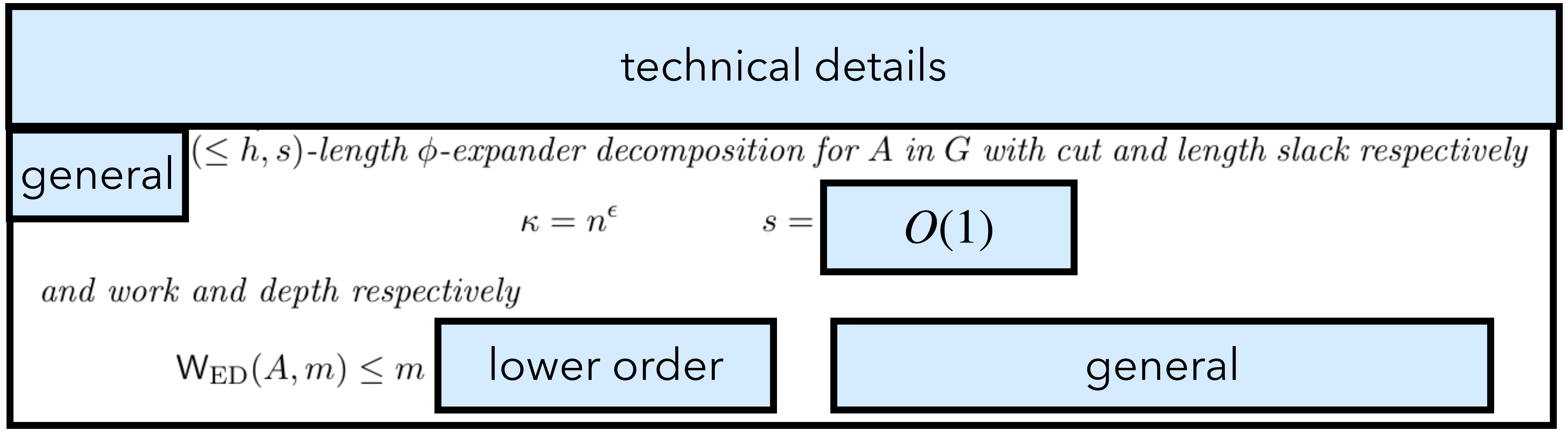
22. **Method Z** is rapidly developed and extended to get the **solution** to **problem X**.

≈Terry Tao

How to Learn Theory

How to Learn Theory

Simplification



ignore lower order parameters / technical details

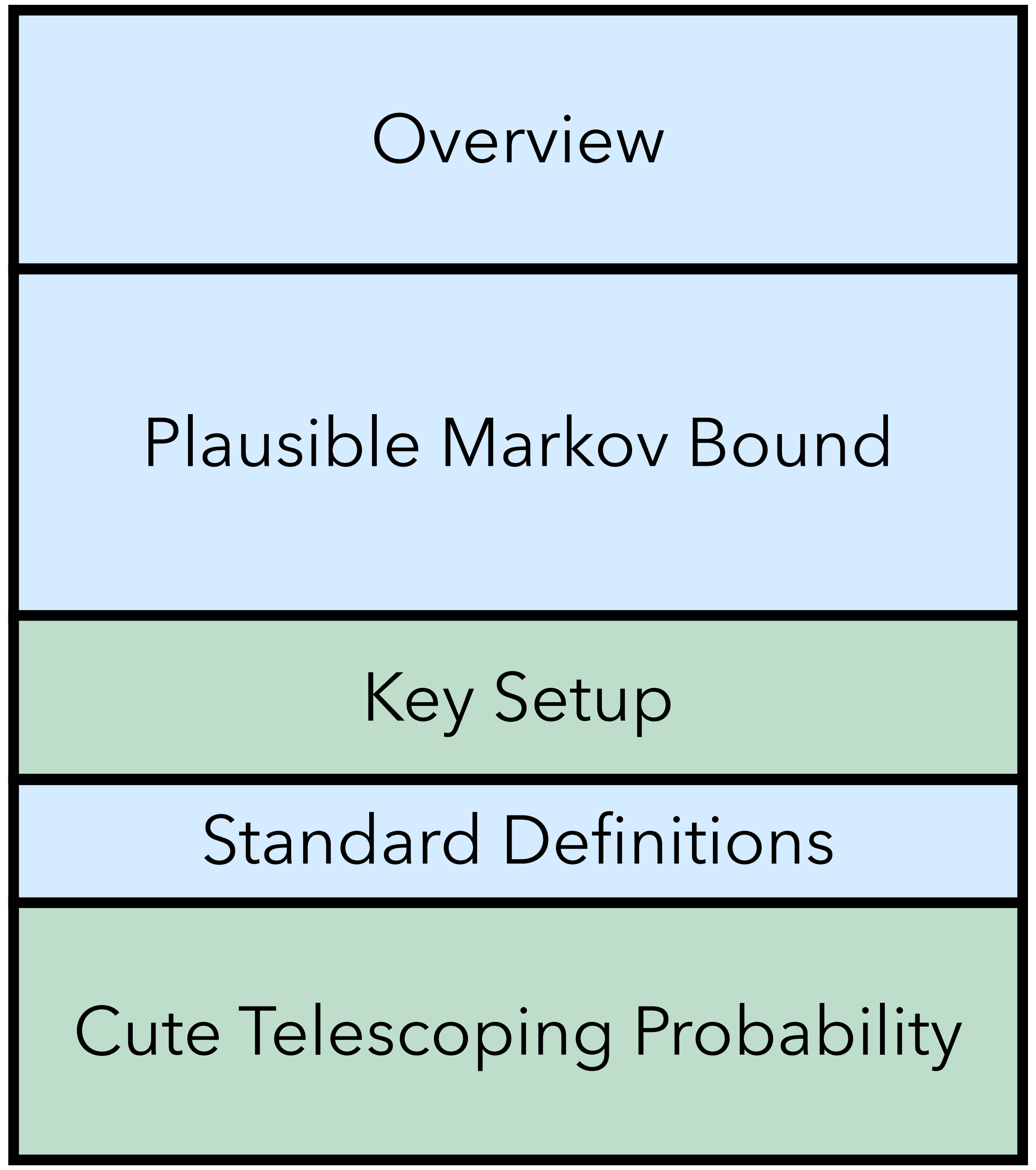
fix parameters

apply theorem to special cases

Simplify theorems

How to Learn Theory

Simplification

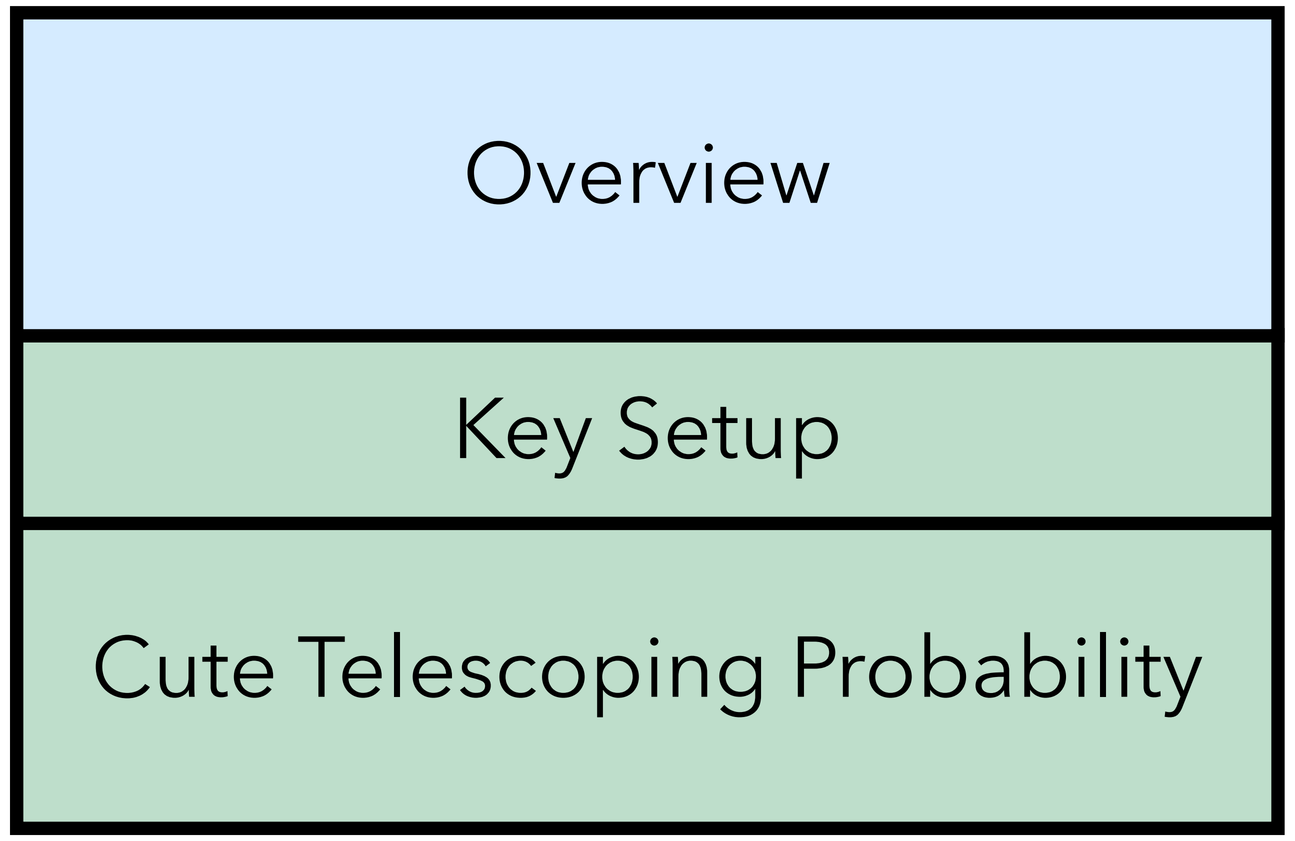


Proof on Arbitrary Graphs

*skip standard +
plausible details*

note tricks

*do proof on
special cases*

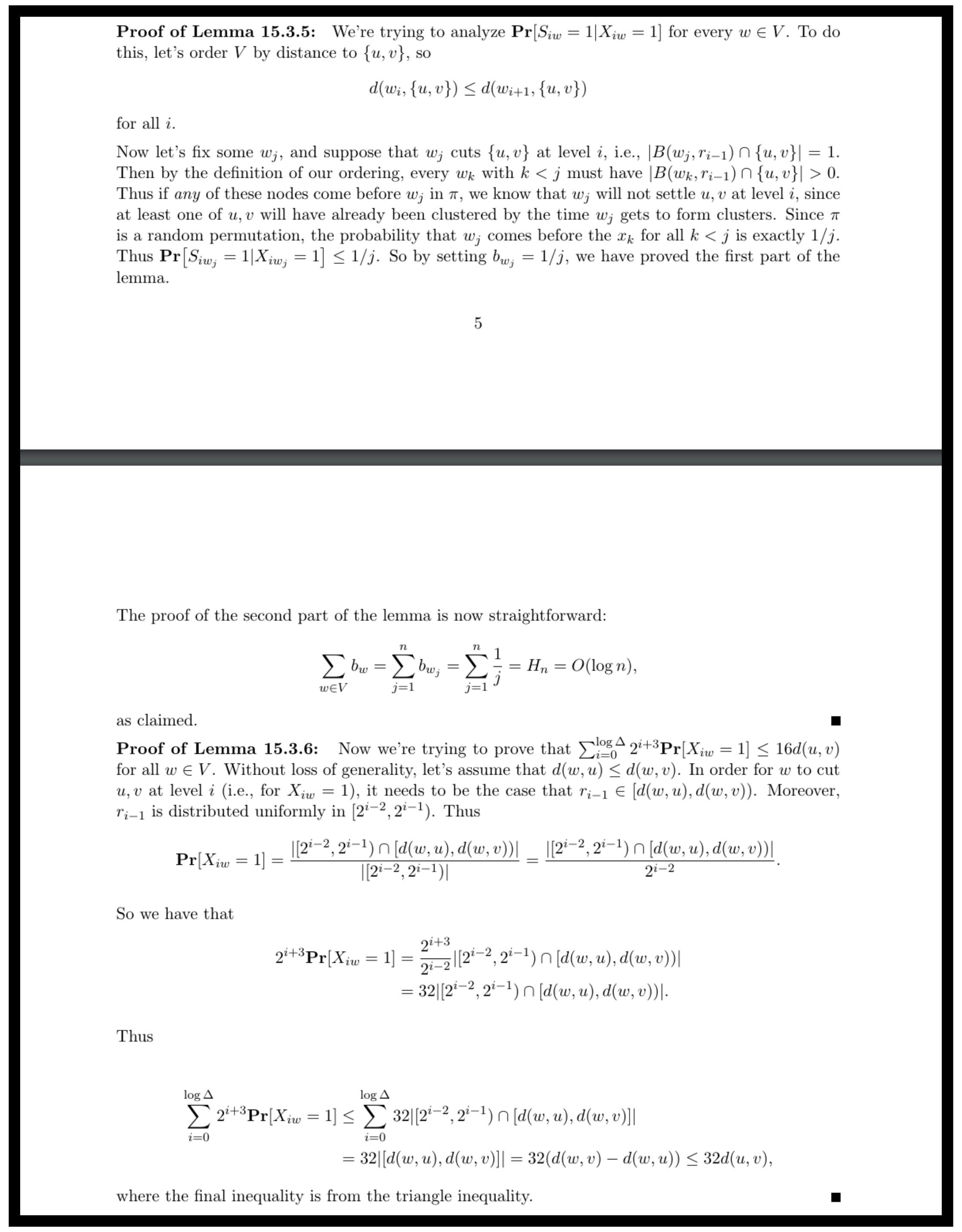


Proof on Regular Graphs

Simplify proofs

How to Learn Theory

Active Engagement



Proof of Lemma 15.3.5: We're trying to analyze $\mathbf{Pr}[S_{iw} = 1 | X_{iw} = 1]$ for every $w \in V$. To do this, let's order V by distance to $\{u, v\}$, so

$$d(w_i, \{u, v\}) \leq d(w_{i+1}, \{u, v\})$$

for all i.

Now let's fix some w_j, and suppose that w_j cuts $\{u, v\}$ at level i, i.e., $|B(w_j, r_{i-1}) \cap \{u, v\}| = 1$. Then by the definition of our ordering, every w_k with $k < j$ must have $|B(w_k, r_{i-1}) \cap \{u, v\}| > 0$. Thus if *any* of these nodes come before w_j in π, we know that w_j will not settle u, v at level i, since at least one of u, v will have already been clustered by the time w_j gets to form clusters. Since π is a random permutation, the probability that w_j comes before the x_k for all $k < j$ is exactly $1/j$. Thus $\mathbf{Pr}\left[S_{iw_j} = 1 | X_{iw_j} = 1\right] \leq 1/j$. So by setting $b_{w_j} = 1/j$, we have proved the first part of the lemma.

5

The proof of the second part of the lemma is now straightforward:

$$\sum_{w \in V} b_w = \sum_{j=1}^{n} b_{w_j} = \sum_{j=1}^{n} \frac{1}{j} = H_n = O(\log n),$$

as claimed. ∎

Proof of Lemma 15.3.6: Now we're trying to prove that $\sum_{i=0}^{\log \Delta} 2^{i+3} \mathbf{Pr}[X_{iw} = 1] \leq 16 d(u, v)$ for all $w \in V$. Without loss of generality, let's assume that $d(w, u) \leq d(w, v)$. In order for w to cut u, v at level i (i.e., for $X_{iw} = 1$), it needs to be the case that $r_{i-1} \in [d(w, u), d(w, v))$. Moreover, r_{i-1} is distributed uniformly in $[2^{i-2}, 2^{i-1})$. Thus

$$\mathbf{Pr}[X_{iw} = 1] = \frac{|[2^{i-2}, 2^{i-1}) \cap [d(w, u), d(w, v))|}{|[2^{i-2}, 2^{i-1})|} = \frac{|[2^{i-2}, 2^{i-1}) \cap [d(w, u), d(w, v))|}{2^{i-2}}.$$

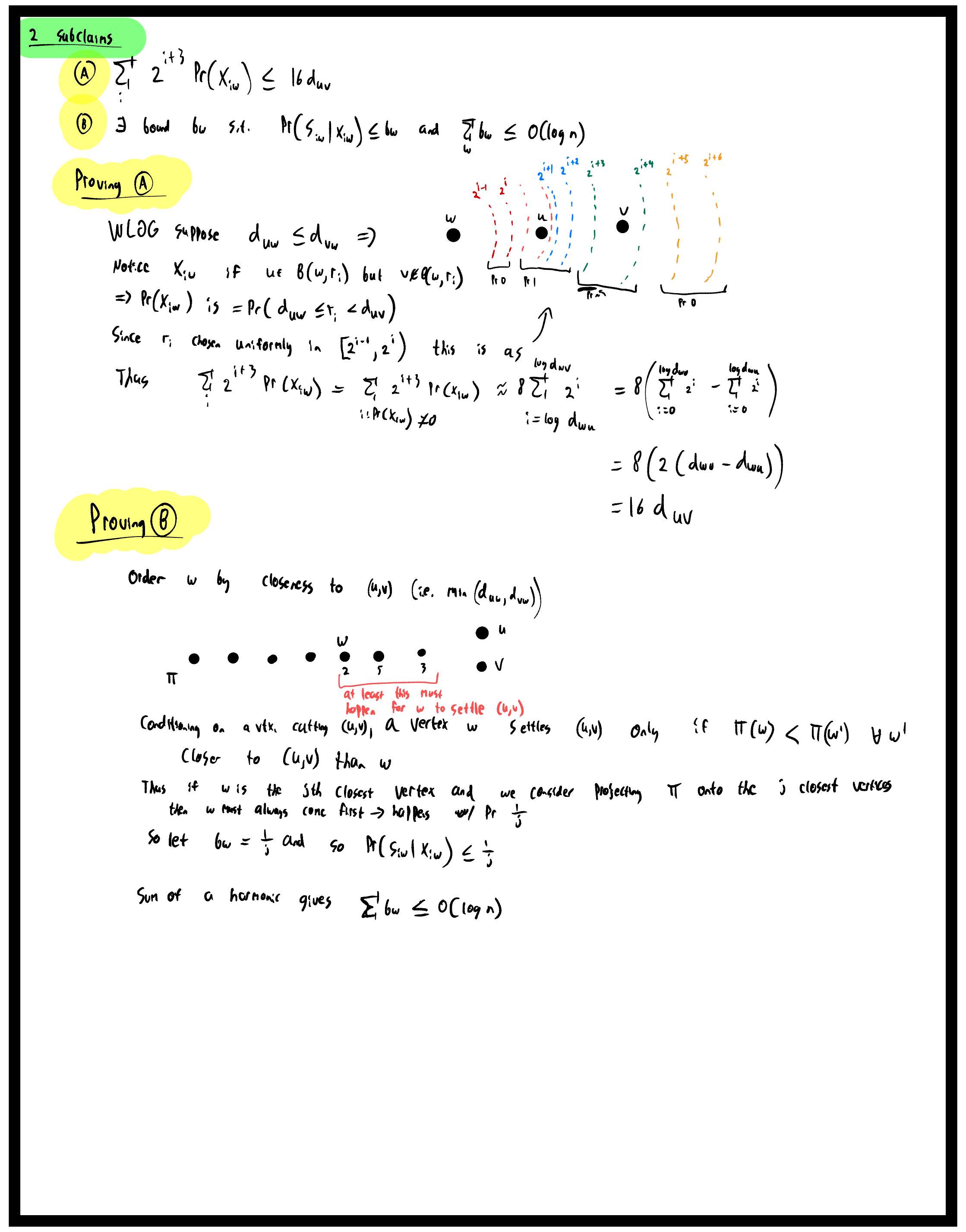
So we have that

$$\begin{aligned} 2^{i+3} \mathbf{Pr}[X_{iw} = 1] &= \frac{2^{i+3}}{2^{i-2}} |[2^{i-2}, 2^{i-1}) \cap [d(w, u), d(w, v))| \\ &= 32 |[2^{i-2}, 2^{i-1}) \cap [d(w, u), d(w, v))|. \end{aligned}$$

Thus

$$\begin{aligned} \sum_{i=0}^{\log \Delta} 2^{i+3} \mathbf{Pr}[X_{iw} = 1] &\leq \sum_{i=0}^{\log \Delta} 32 |[2^{i-2}, 2^{i-1}) \cap [d(w, u), d(w, v)]| \\ &= 32 |[d(w, u), d(w, v)]| = 32(d(w, v) - d(w, u)) \leq 32 d(u, v), \end{aligned}$$

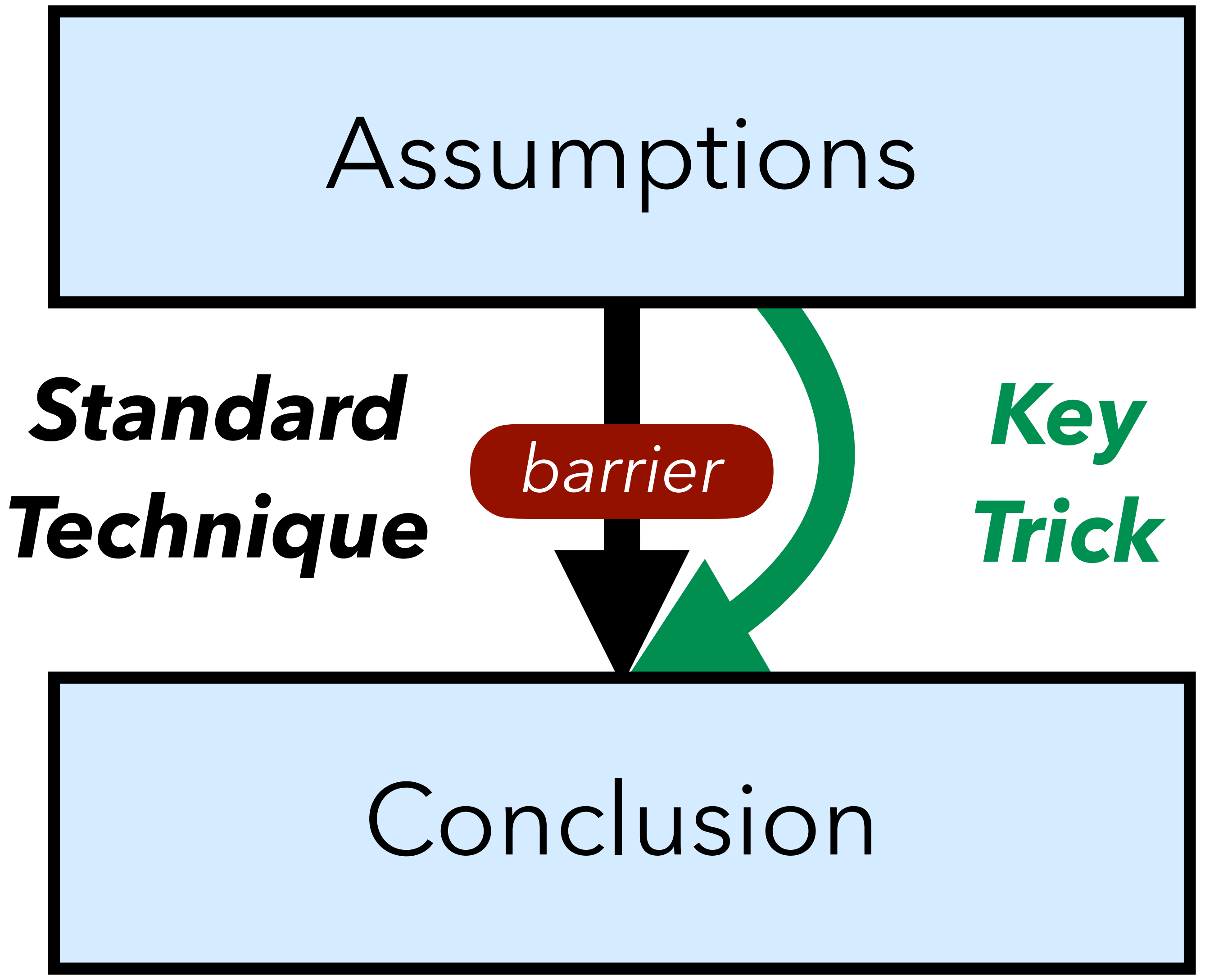
where the final inequality is from the triangle inequality. ∎

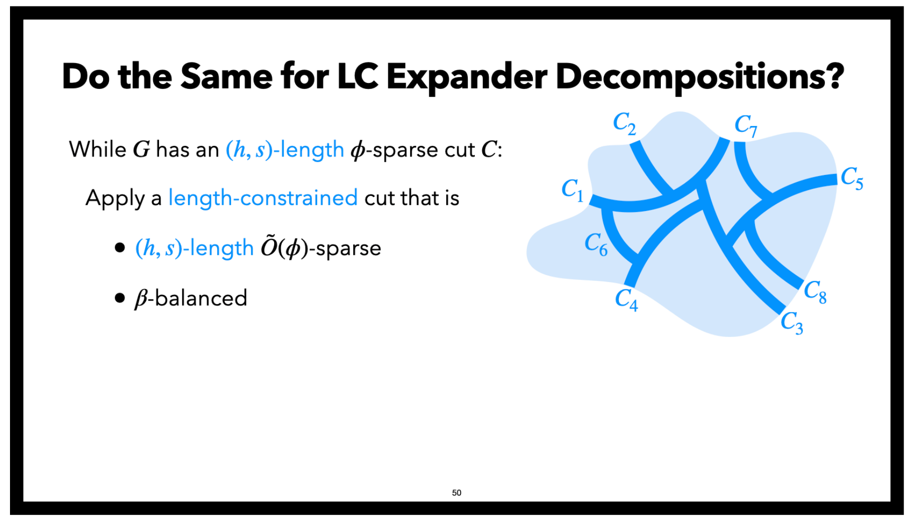


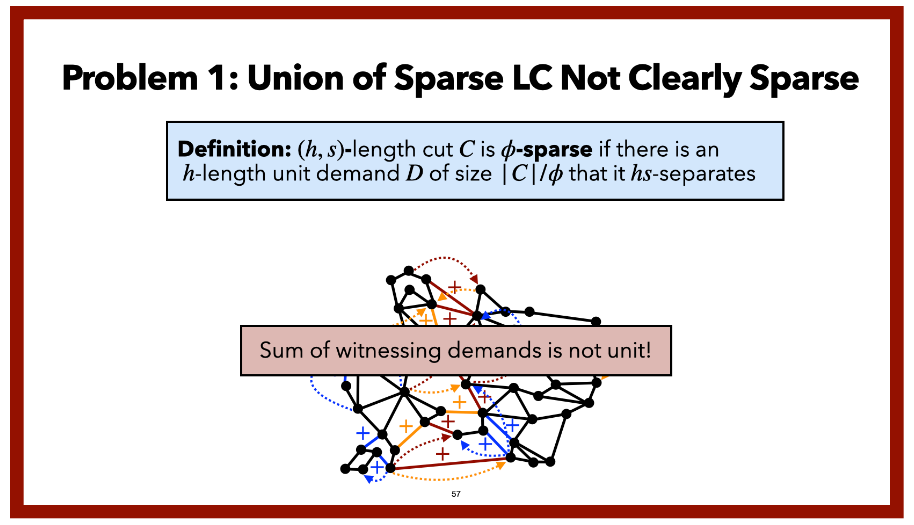
Recreate Proofs after you learn them; see where you get stuck

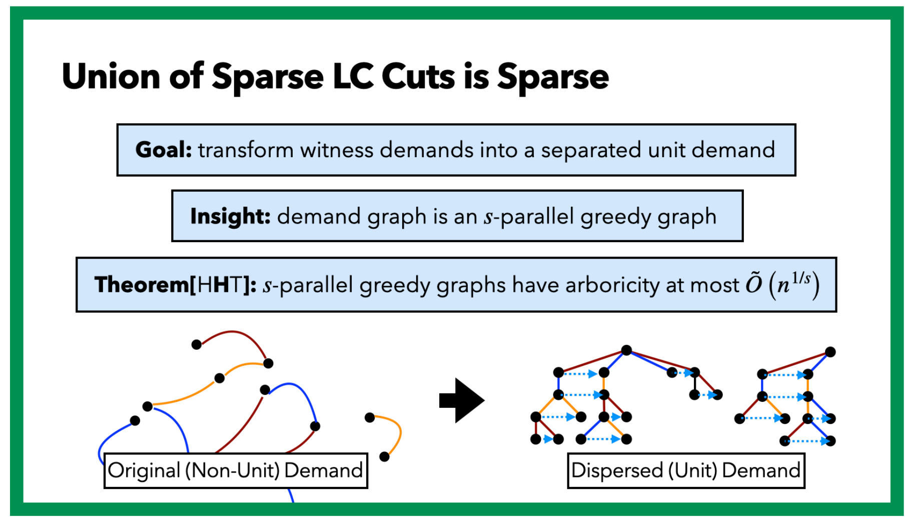
How to Learn Theory

Active Engagement







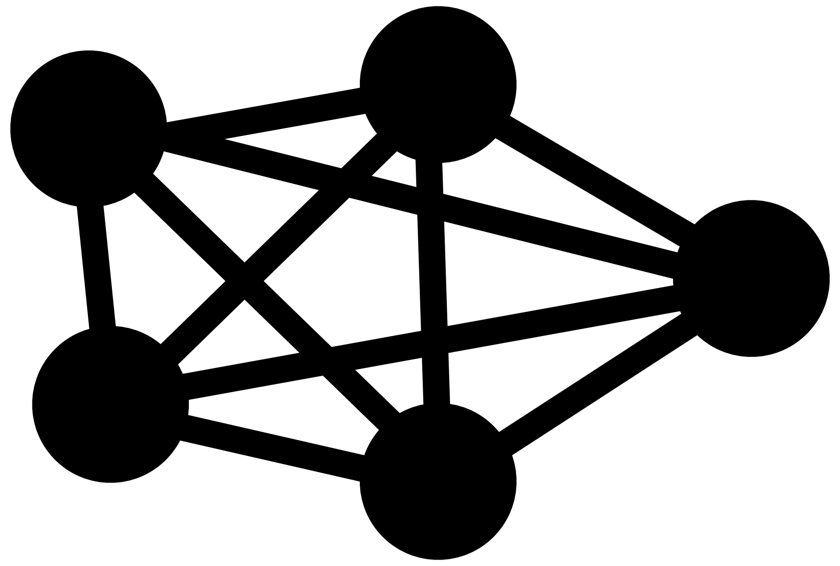


Invent Stories that you like / will remember

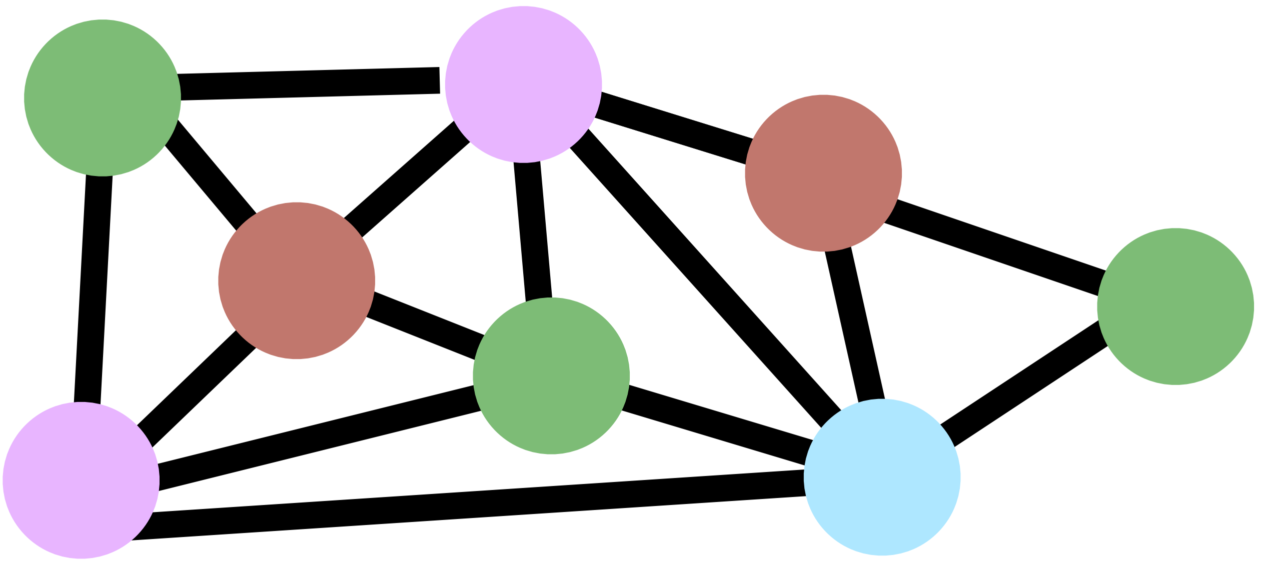
How to Learn Theory

Active Engagement

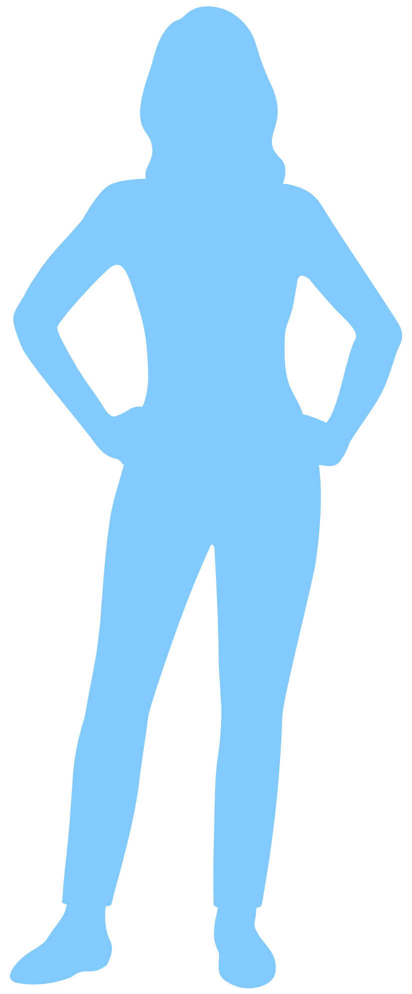
Theorem: Every planar graph is 4-colorable



why assumptions needed?



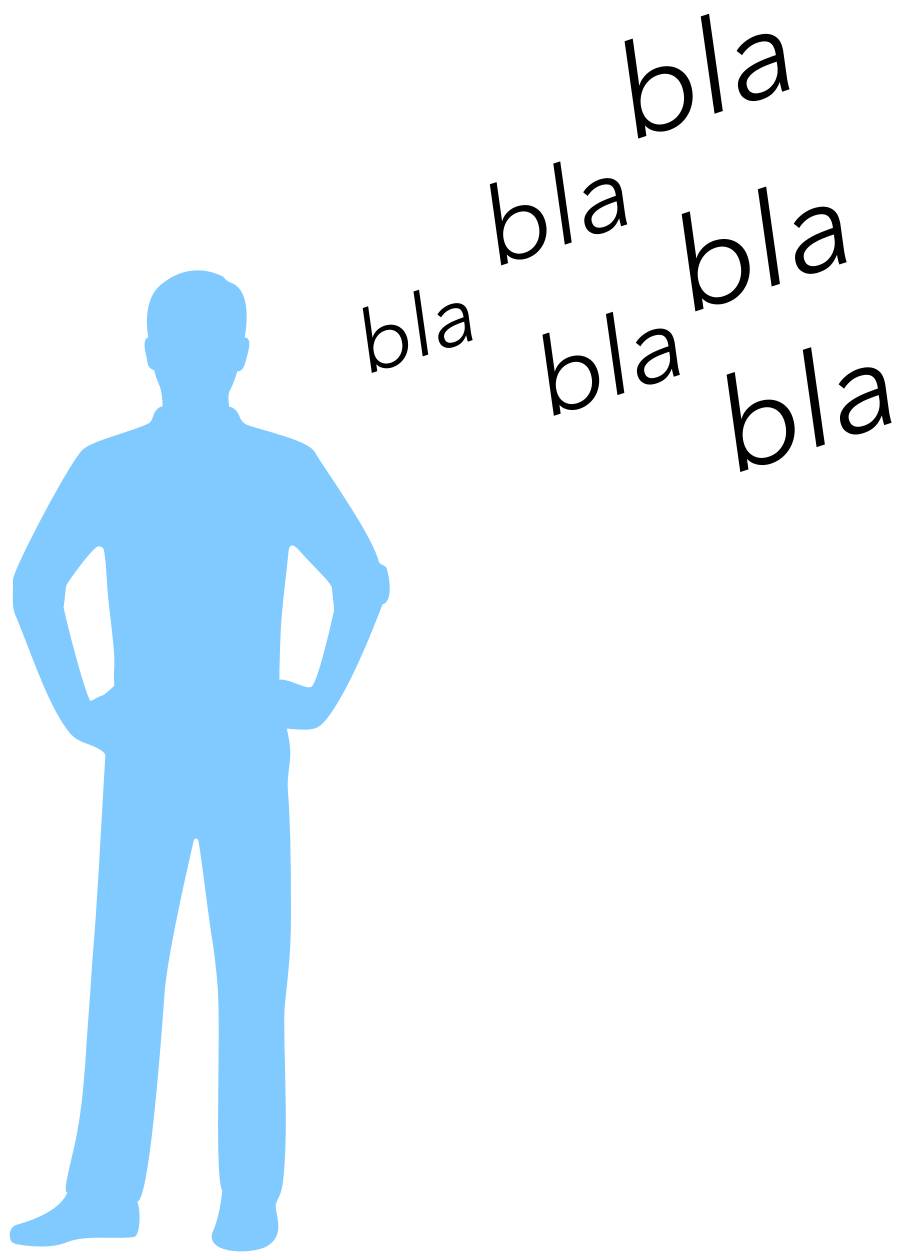
what does this give on e.g.s?

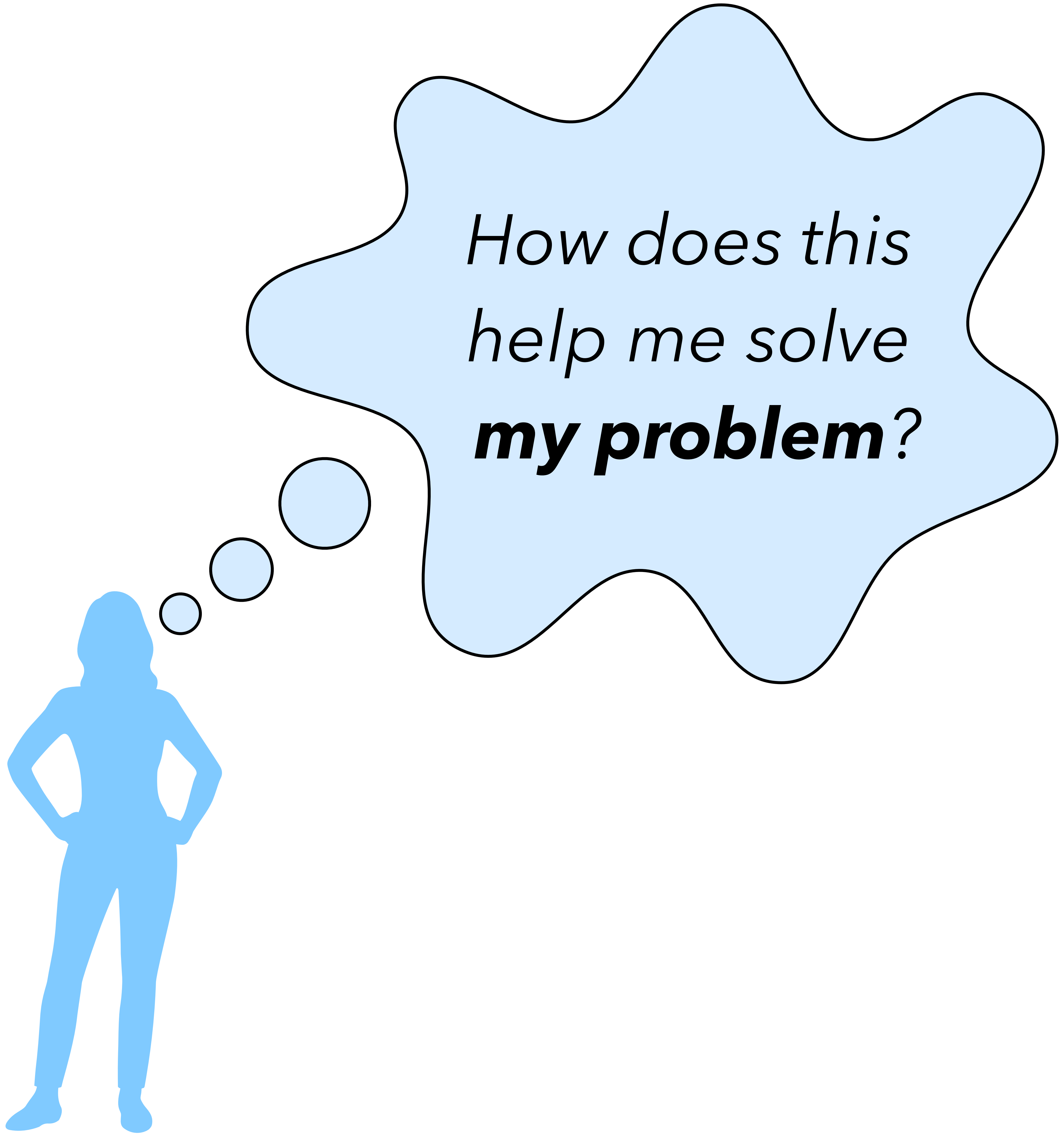


Ask Yourself Questions

How to Learn Theory

Active Engagement

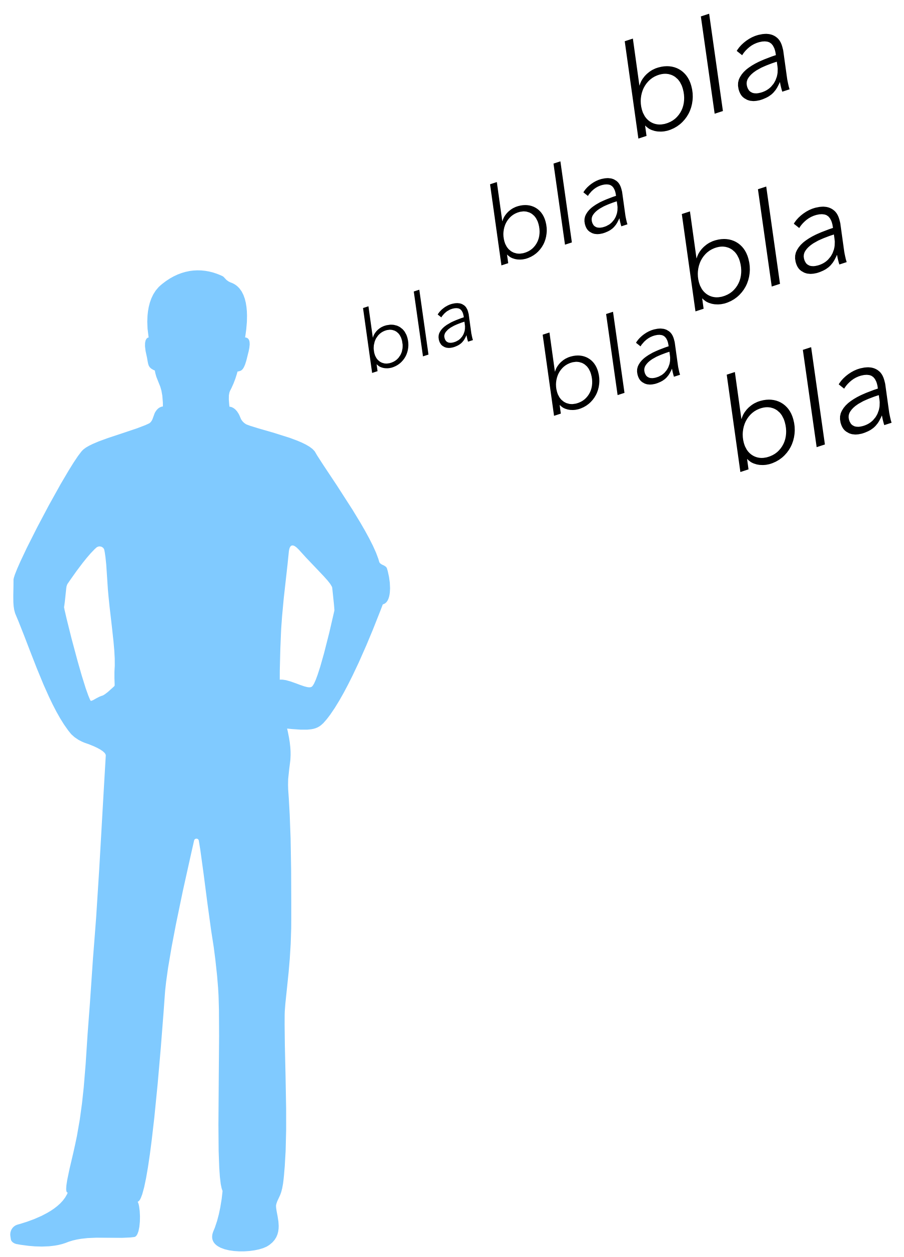


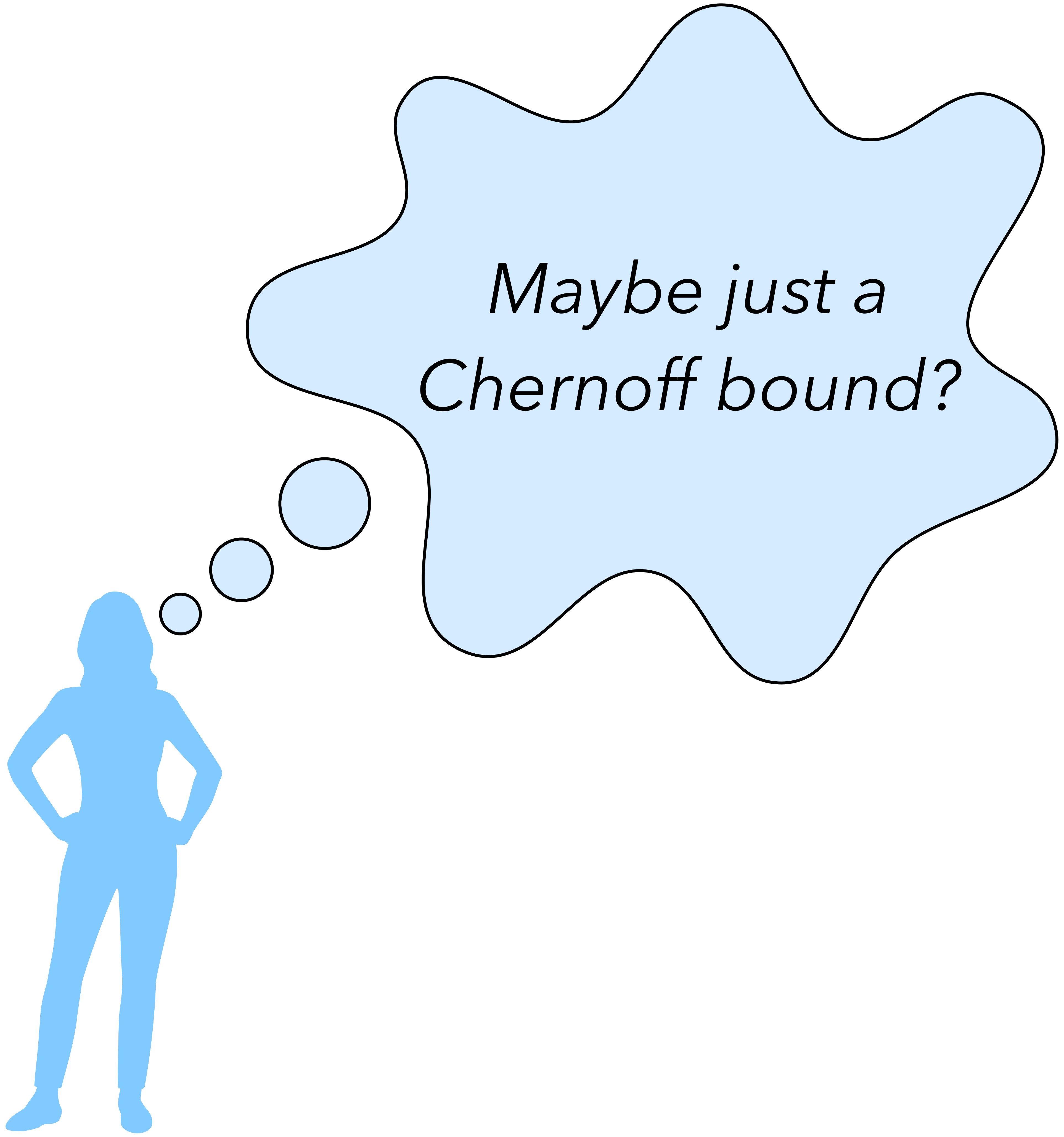


Anchor Your Learning with a problem you like

How to Learn Theory

Active Engagement

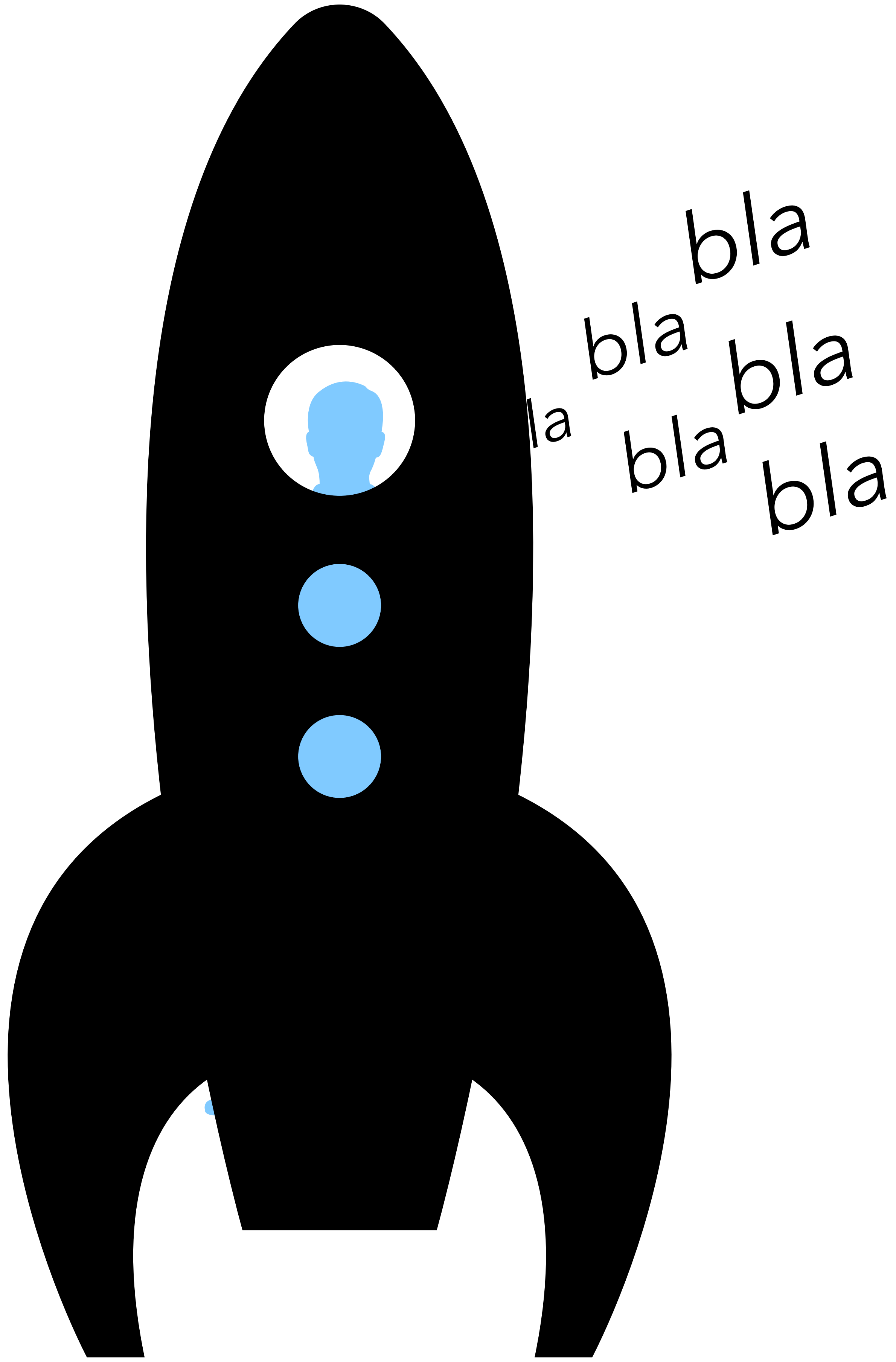




Guess what's coming next

How to Learn Theory

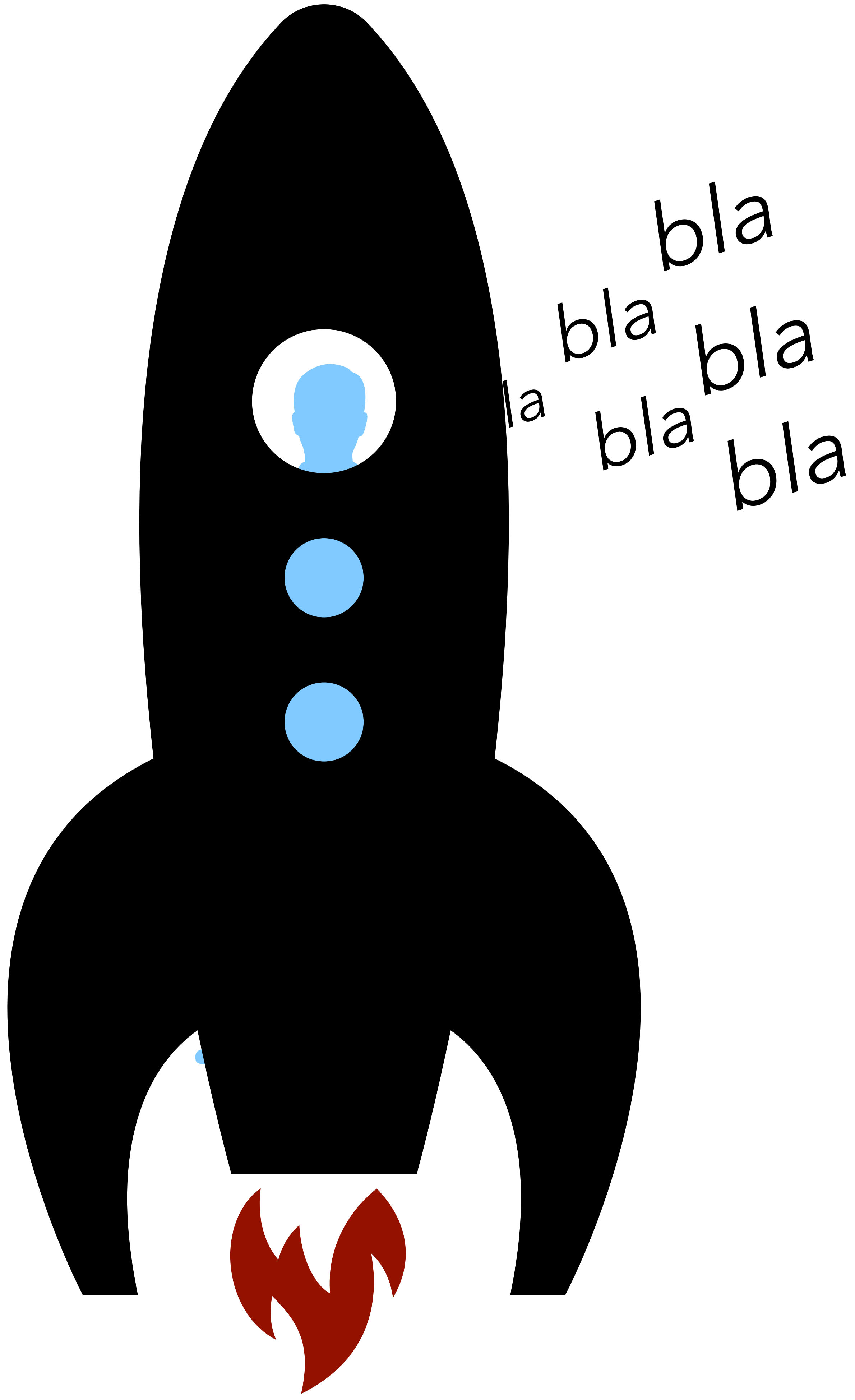
Active Engagement



Ask Questions if you're confused

How to Learn Theory

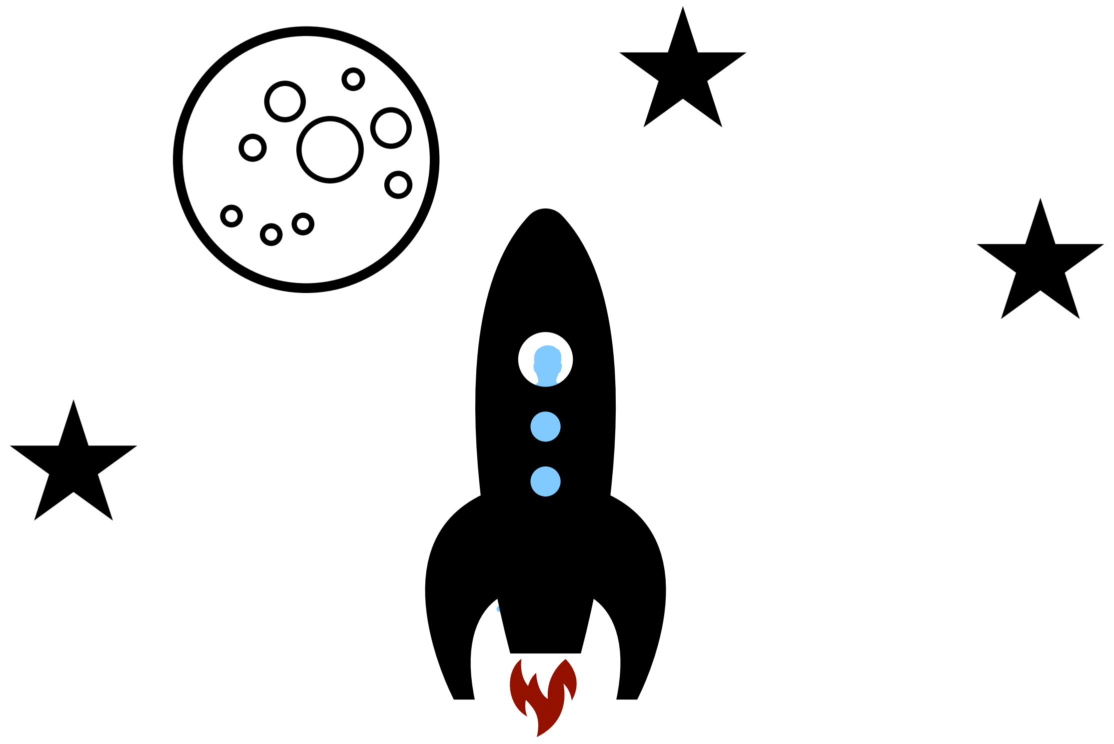
Active Engagement

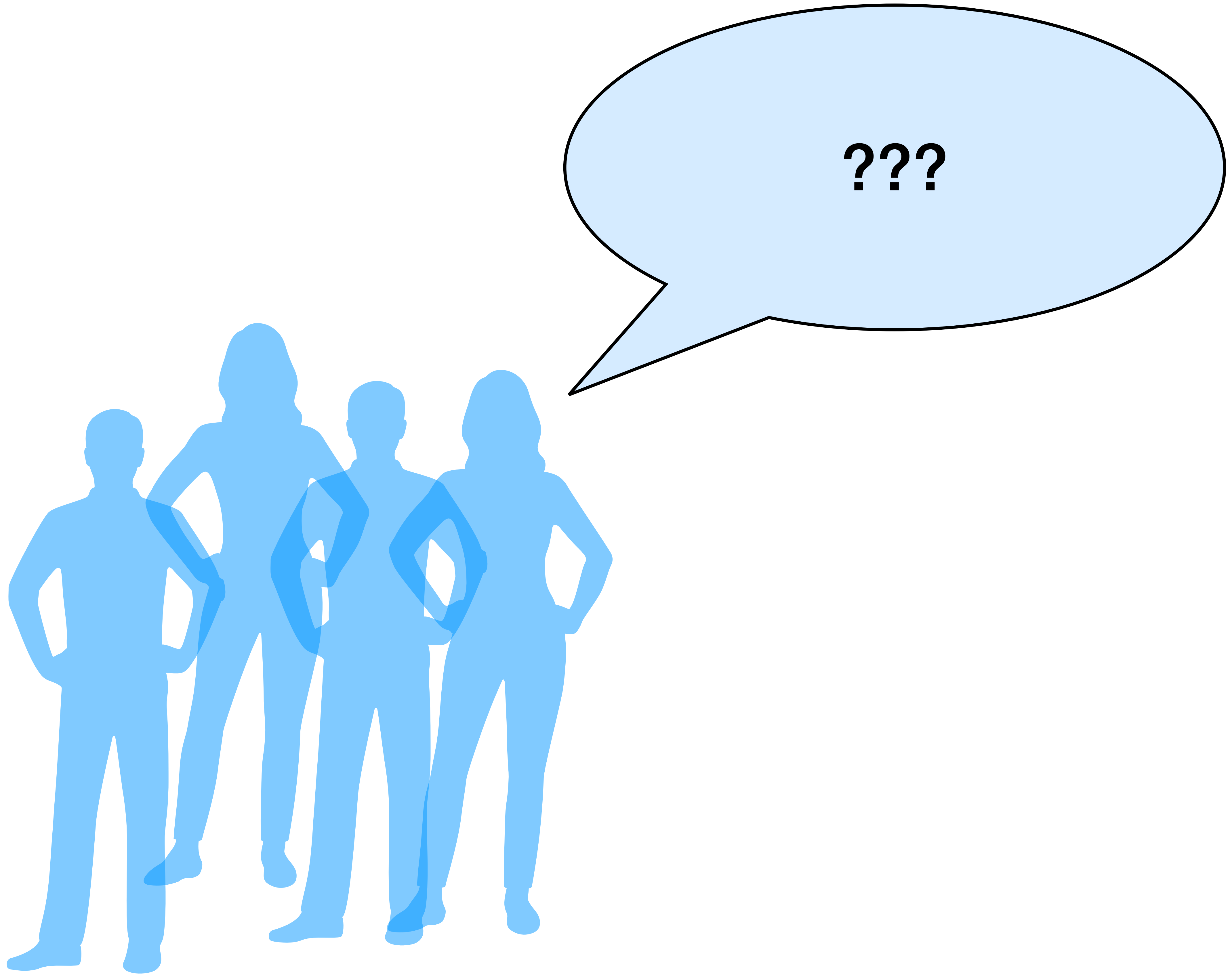


Ask Questions if you're confused

How to Learn Theory

Active Engagement

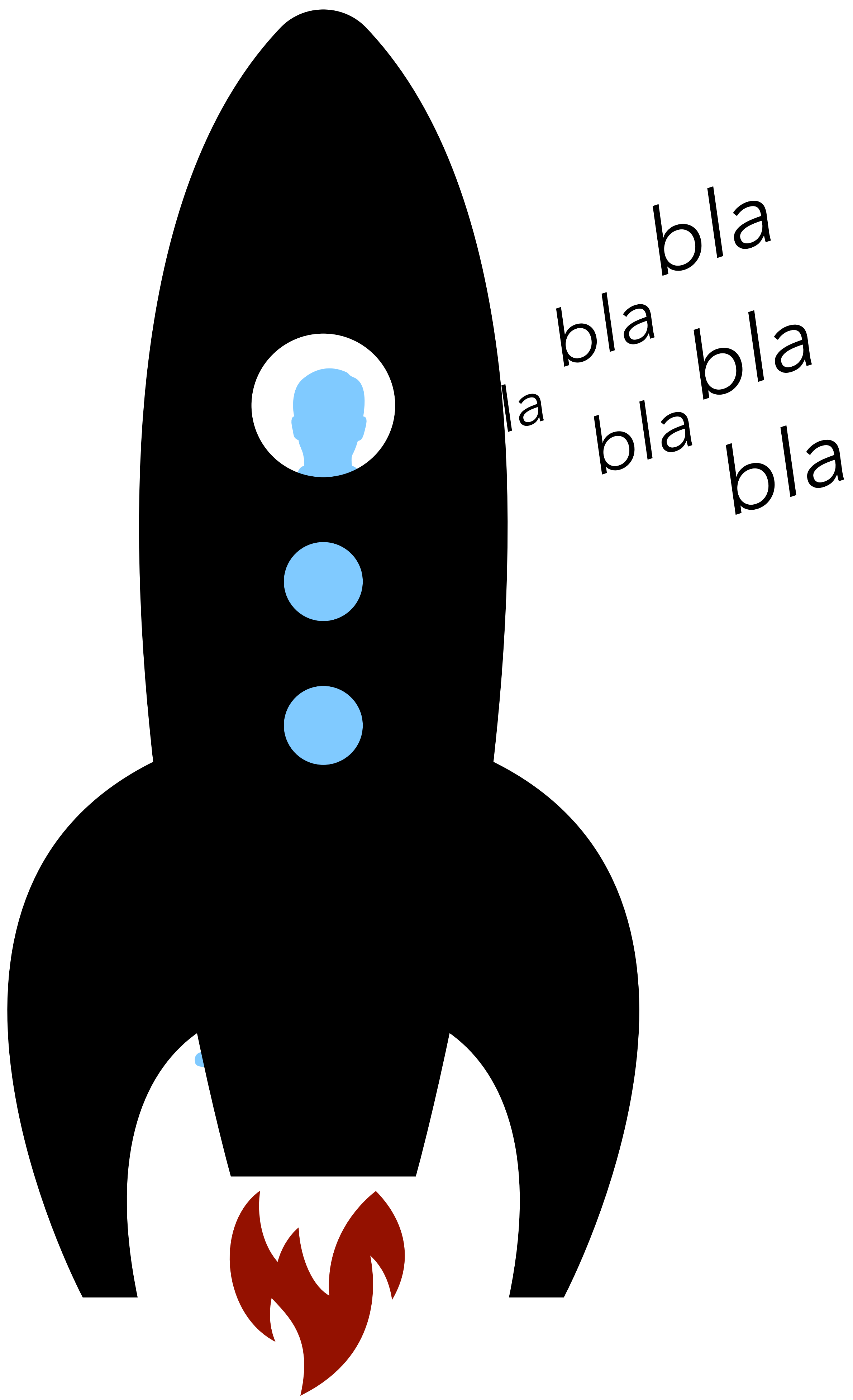


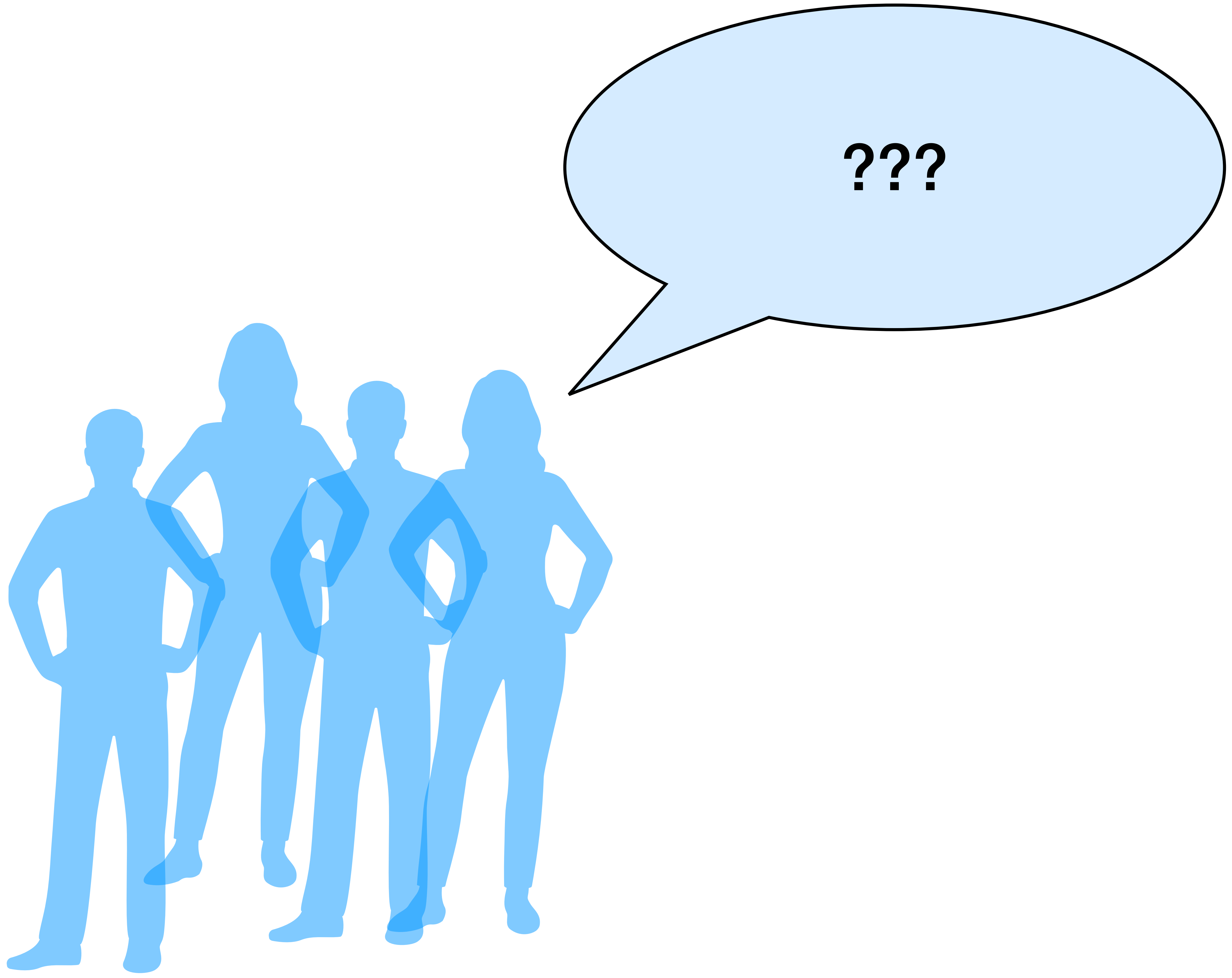


Ask Questions if you're confused

How to Learn Theory

Active Engagement

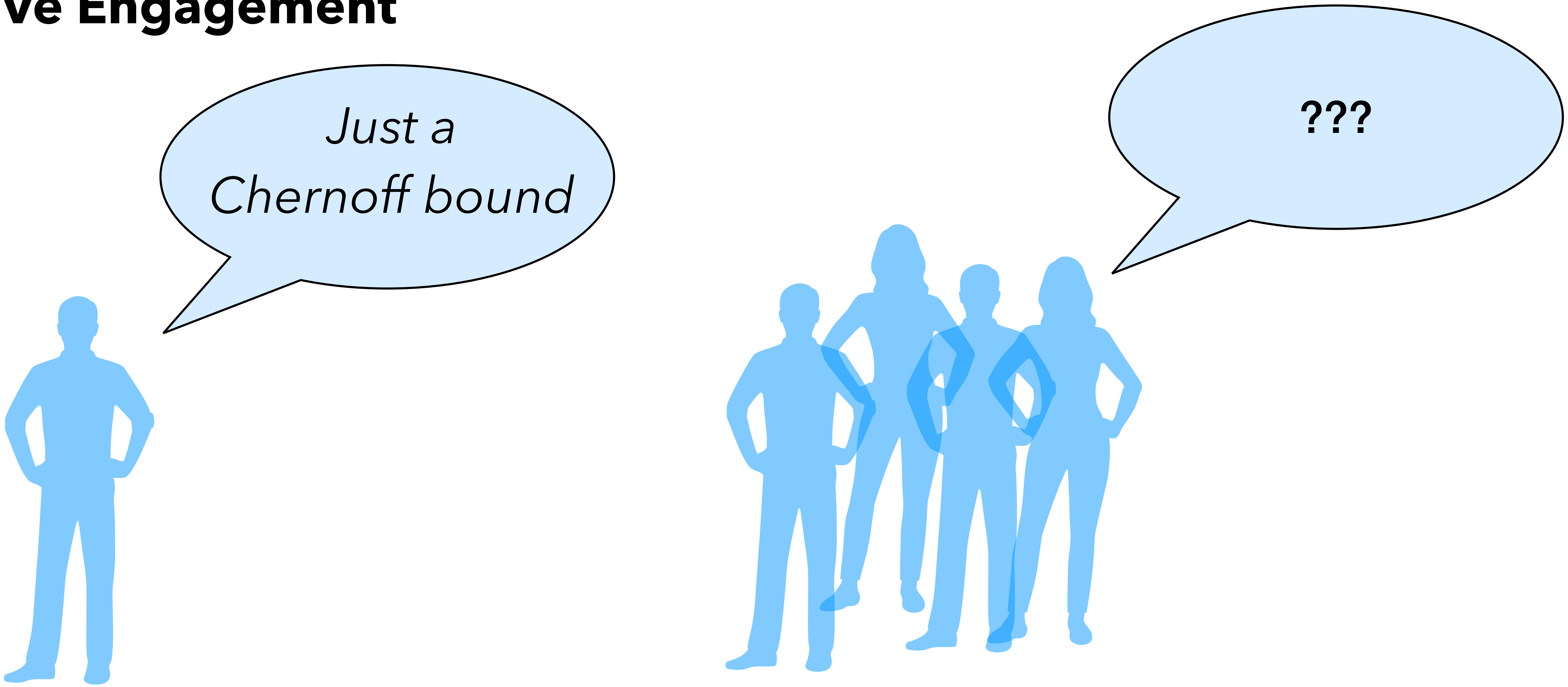




Ask Questions if you're confused

How to Learn Theory

Active Engagement



Ask Questions if you're confused

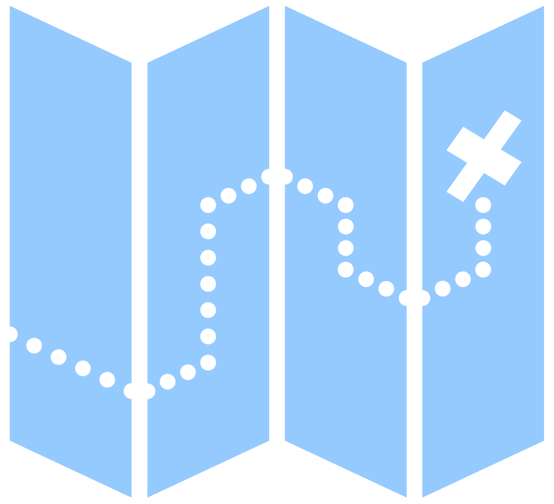
How to Learn Theory

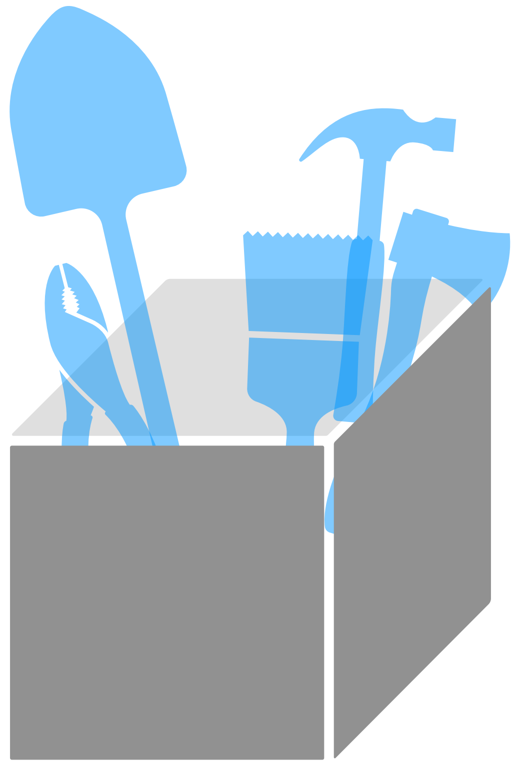
What to Aim For

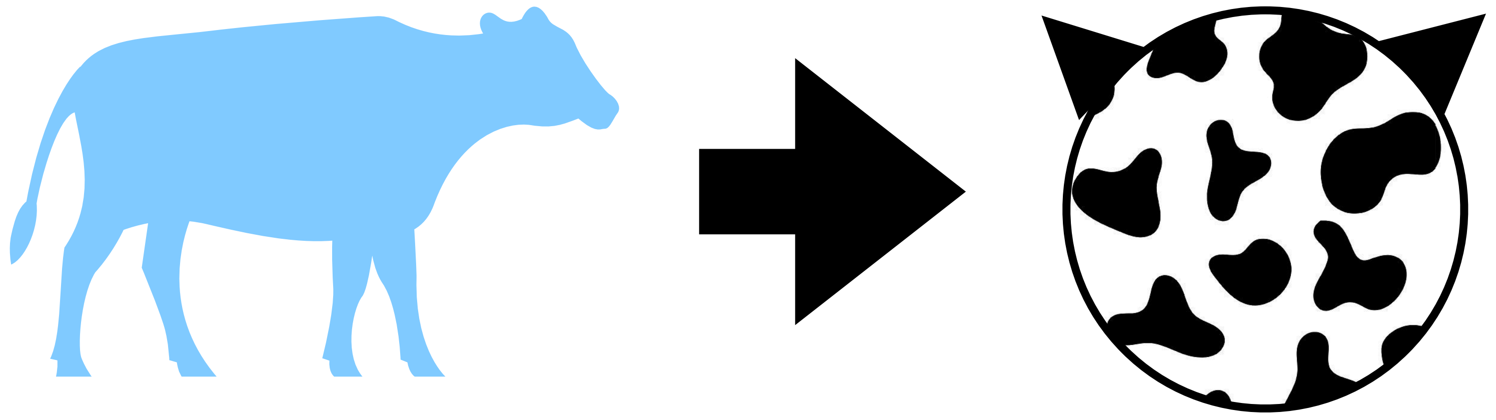
Roadmaps of proof

Tools and stories you'll remember

Intuition of how to think about complexity







a cow

a cow
(up to constants)

How to Do Theory

How to Do Theory

Doing Theory is Hard

Can Succeed with a Wide Range of Aptitudes:

- Mathematical **quick**ness
- **Good memory**
- Good **intuition**
- Reliab
- Just really **curious**
- **Stick-To-Itiveness**
- **Impatient** / only interested in elegant solutions
- …

Work to Your Strengths

How to Do Theory

You Will Get Stuck

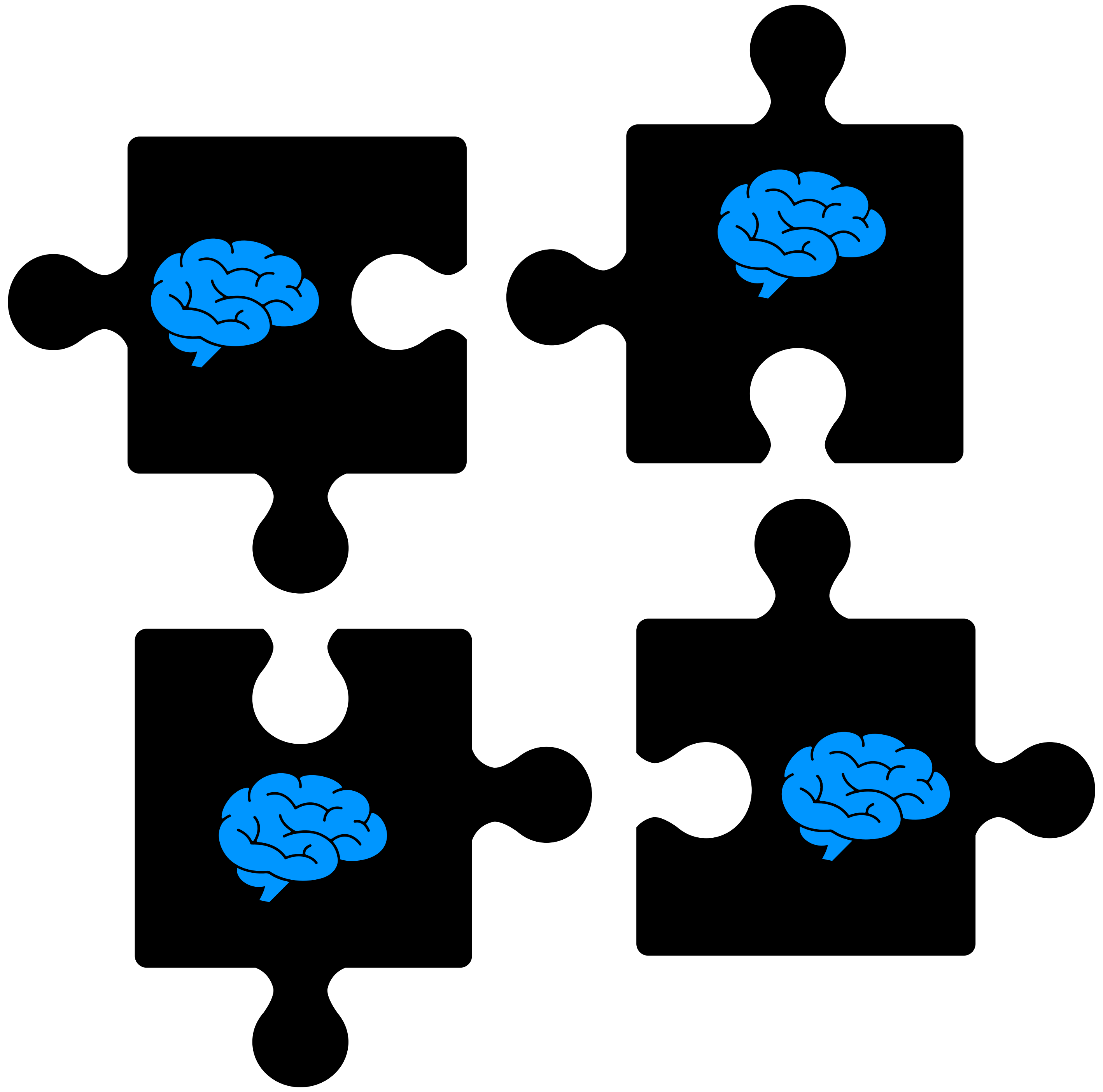
Theorem: Every planar graph is 4-colorable 😓

Theorem: Every tree is 4-colorable 😀

Simplify your problem

How to Do Theory

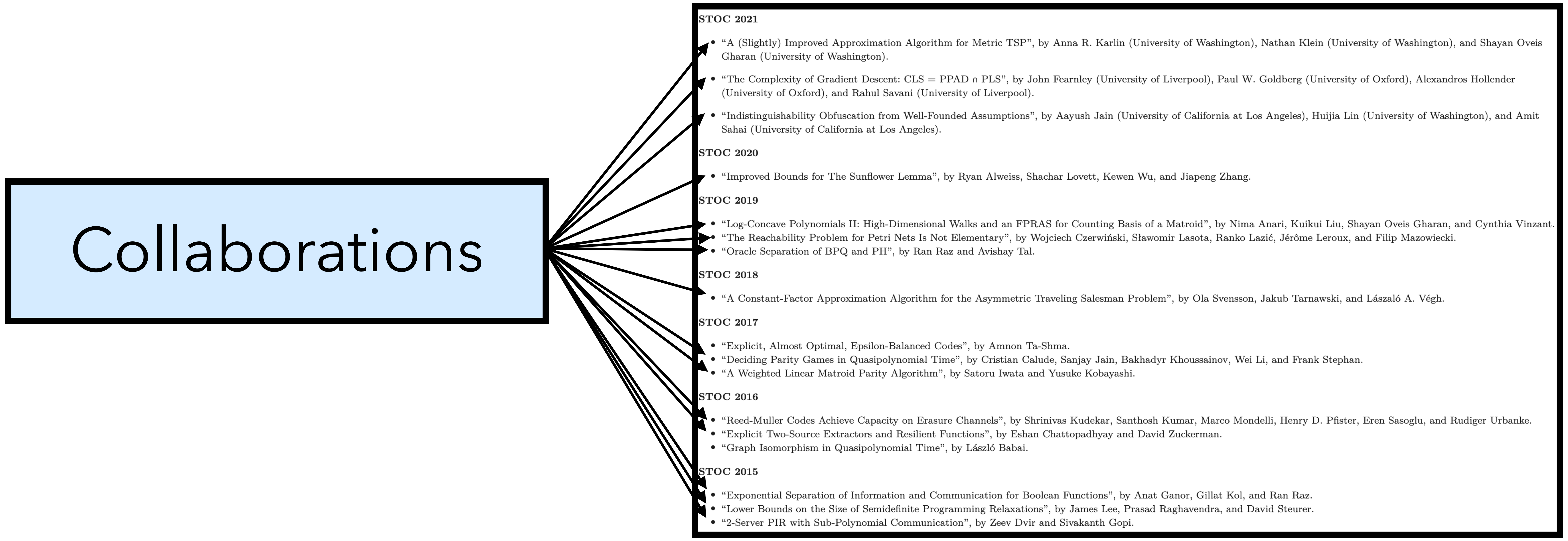
You Will Get Stuck



Collaborate

How to Do Theory

You Will Get Stuck



STOC 2021

- “A (Slightly) Improved Approximation Algorithm for Metric TSP”, by Anna R. Karlin (University of Washington), Nathan Klein (University of Washington), and Shayan Oveis Gharan (University of Washington).
- “The Complexity of Gradient Descent: CLS = PPAD ∩ PLS”, by John Fearnley (University of Liverpool), Paul W. Goldberg (University of Oxford), Alexandros Hollender (University of Oxford), and Rahul Savani (University of Liverpool).
- “Indistinguishability Obfuscation from Well-Founded Assumptions”, by Aayush Jain (University of California at Los Angeles), Huijia Lin (University of Washington), and Amit Sahai (University of California at Los Angeles).

STOC 2020

- “Improved Bounds for The Sunflower Lemma”, by Ryan Alweiss, Shachar Lovett, Kewen Wu, and Jiapeng Zhang.

STOC 2019

- “Log-Concave Polynomials II: High-Dimensional Walks and an FPRAS for Counting Basis of a Matroid”, by Nima Anari, Kuikui Liu, Shayan Oveis Gharan, and Cynthia Vinzant.
- “The Reachability Problem for Petri Nets Is Not Elementary”, by Wojciech Czerwiński, Sławomir Lasota, Ranko Lazić, Jérôme Leroux, and Filip Mazowiecki.
- “Oracle Separation of BPQ and PH”, by Ran Raz and Avishay Tal.

STOC 2018

- “A Constant-Factor Approximation Algorithm for the Asymmetric Traveling Salesman Problem”, by Ola Svensson, Jakub Tarnawski, and László A. Végh.

STOC 2017

- “Explicit, Almost Optimal, Epsilon-Balanced Codes”, by Amnon Ta-Shma.
- “Deciding Parity Games in Quasipolynomial Time”, by Cristian Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan.
- “A Weighted Linear Matroid Parity Algorithm”, by Satoru Iwata and Yusuke Kobayashi.

STOC 2016

- “Reed-Muller Codes Achieve Capacity on Erasure Channels”, by Shrinivas Kudekar, Santhosh Kumar, Marco Mondelli, Henry D. Pfister, Eren Sasoglu, and Rudiger Urbanke.
- “Explicit Two-Source Extractors and Resilient Functions”, by Eshan Chattopadhyay and David Zuckerman.
- “Graph Isomorphism in Quasipolynomial Time”, by László Babai.

STOC 2015

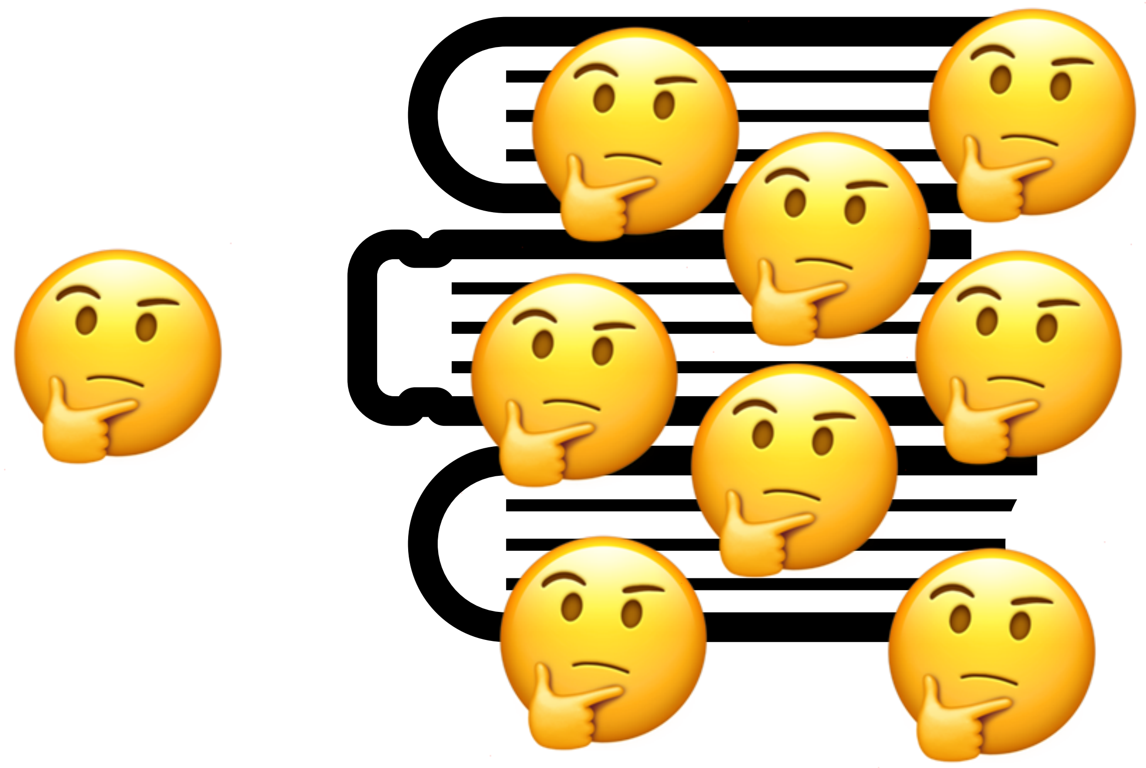
- “Exponential Separation of Information and Communication for Boolean Functions”, by Anat Ganor, Gillat Kol, and Ran Raz.
- “Lower Bounds on the Size of Semidefinite Programming Relaxations”, by James Lee, Prasad Raghavendra, and David Steurer.
- “2-Server PIR with Sub-Polynomial Communication”, by Zeev Dvir and Sivakanth Gopi.

STOC Best Papers

Collaborate

How to Do Theory

You Will Get Stuck



Read Related Work

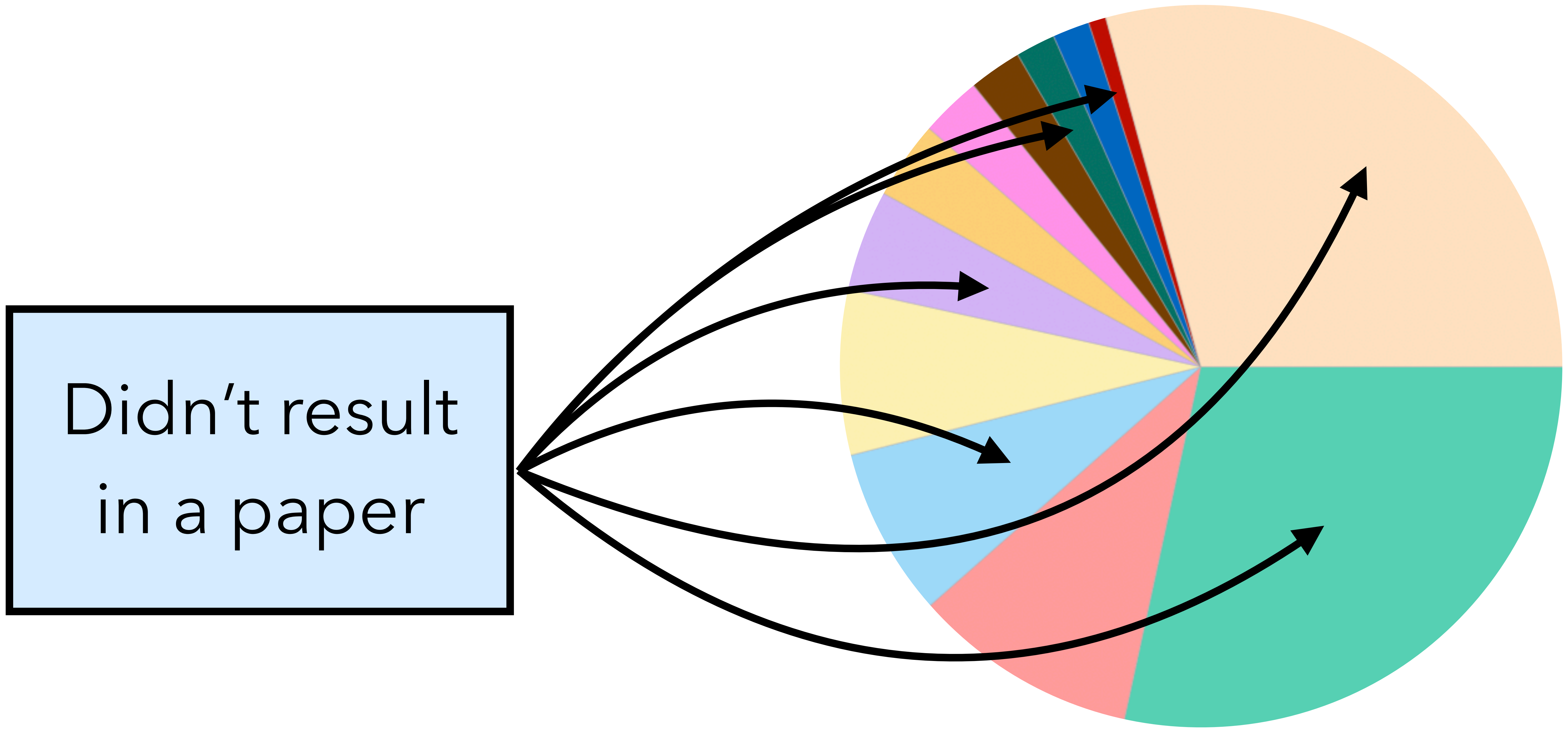
How to Do Theory

You Will Get Stuck

Cut Yourself Slack

How to Do Theory

A Few Mantras



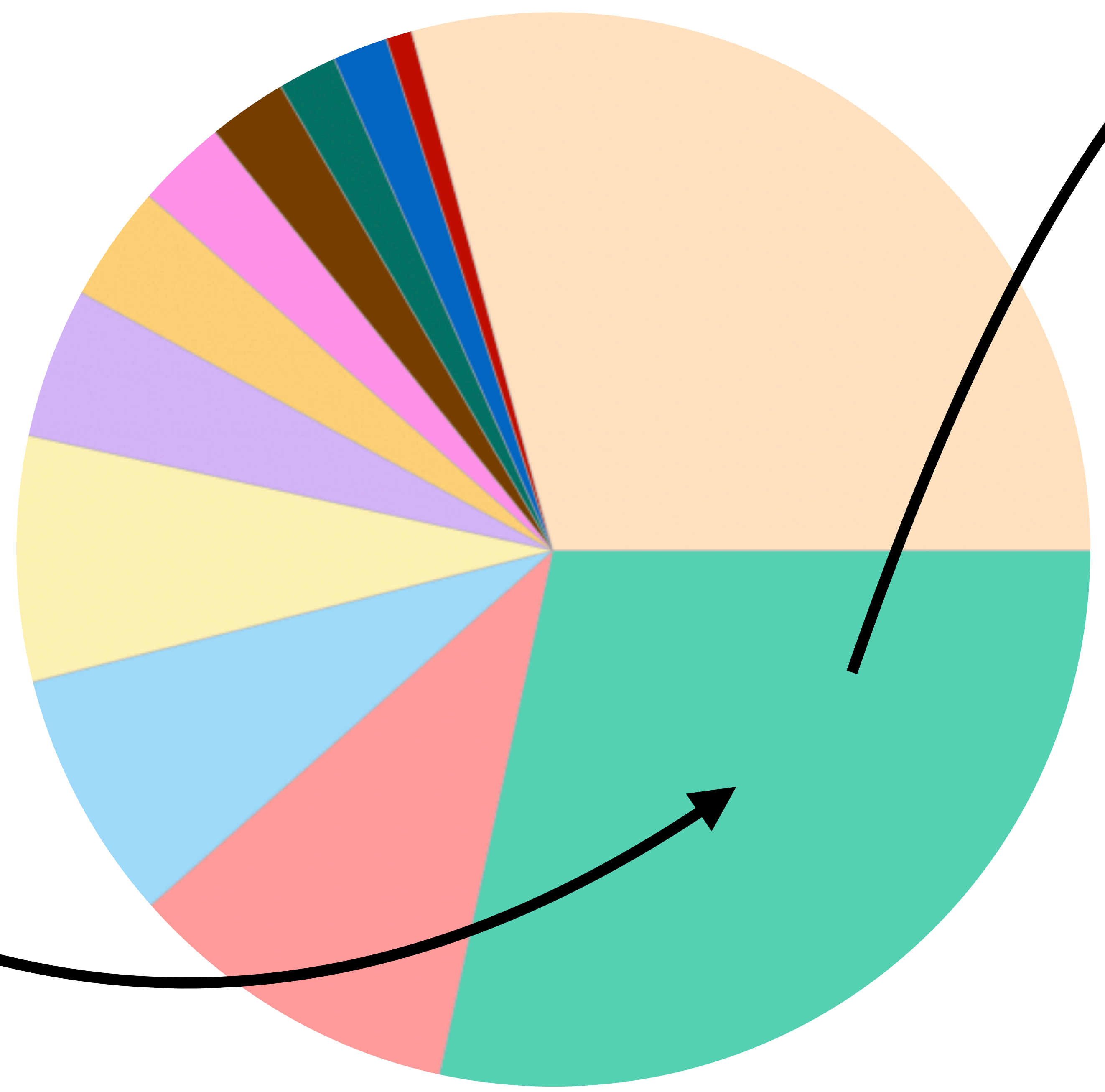
My Time in Grad School

Failure is Common

How to Do Theory

A Few Mantras

Didn't result in a paper



My Time in Grad School

But resulted in **knowledge**

which resulted in another **problem**

which resulted in a **collaboration** several years later

which resulted in a **paper**

Learning is Progress

How to Do Theory

A Few Mantras

How Theory Problems are Solved

1. Isolate a toy **model case x** of major **problem X**.
2. Solve **model case x** using **method A**.
3. Try using **method A** to solve the full **problem X**.
4. This does not succeed but **method A** can be extended to **model cases x' and x''**.
5. Eventually, it is realized that **method A** relies crucially on a **property P** being true which holds for **model cases x, x' and x''**.
6. Conjecture that **property P** is true for all instances of **problem X**.
7. Discover a family f **counterexamples y, y', y''**,… to this conjecture.
8. Take the simplest **counterexample y** in this family, and try to solve **problem X** for this special case. Meanwhile, try to see whether **method A** can work without **property P**
9. Discover several counterexamples in which **method A** fails, in which the cause of failure can be definitely traced back to **property P**. Abandon efforts to modify **method A**.
10. Realize that **counterexample y** is related to a **problem Z** in another field.

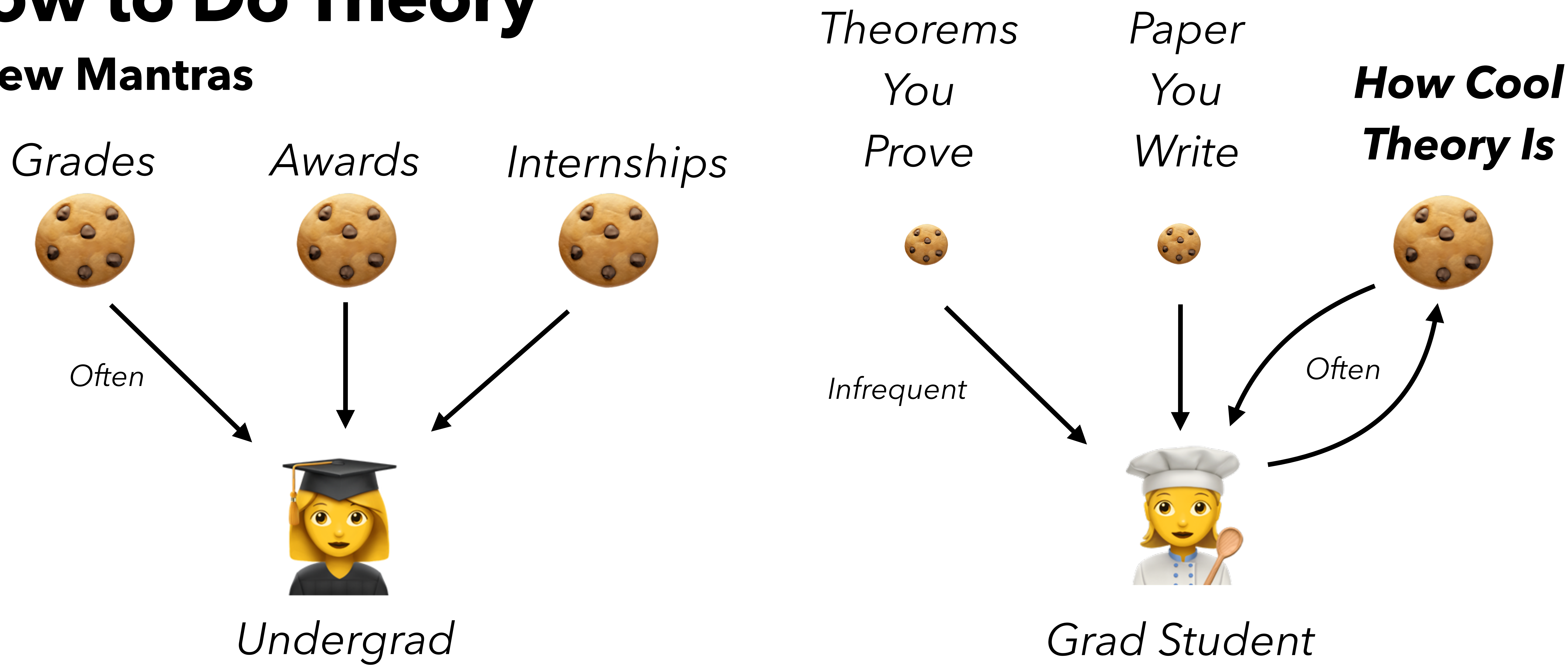
…

22. **Method Z** is rapidly developed and extended to get the **solution** to **problem X**.

Any New Insight is Progress

How to Do Theory

A Few Mantras



Learn to Love the Process Not the Outcome

How to Write Theory

Writing Dos and Don'ts

Bad References

Theorem 1: 1+1+1=3

Proof:
First we show 1+1=2...
Next, we show 2+1=3...

Theorem 2: 2+1+1=4

Proof:
First we show 1+1=2...
Next, we show 2+2=4...

Lemma: 1+1=2

Theorem 1: 1+1+1=3

Proof:
By Lemma 1+1=2
Next, we show 2+1=3...

Theorem 2: 2+1+1=4

Proof:
By Lemma 1+1=2
Next, we show 2+2=4...

Abstract out reused arguments into lemmas

Writing Dos and Don'ts

Bad References

Theorem 1: 1+1+1=3

Proof:
First we show 1+1=2**...**
Next, we show 2+1=3**...**

Theorem 2: 2+1+1=4

Proof:
By the argument in Theorem 1, 1+1=2
Next, we show 2+2=4**...**

Lemma: 1+1=2

Theorem 1: 1+1+1=3

Proof:
By Lemma 1+1=2**...**
Next, we show 2+1=3**...**

Theorem 2: 2+1+1=4

Proof:
By Lemma 1+1=2**...**
Next, we show 2+2=4**...**

Don't reference the insides of other proofs

Writing Dos and Don'ts

Bad References

Theorem 1: 1+1+1=3

Proof:
Hershkowitz et al. showed that 1+1=2**…**
Next, we show 2+1=3**…**

Lemma[Hershkowitz et al. **]:** 1+1=2

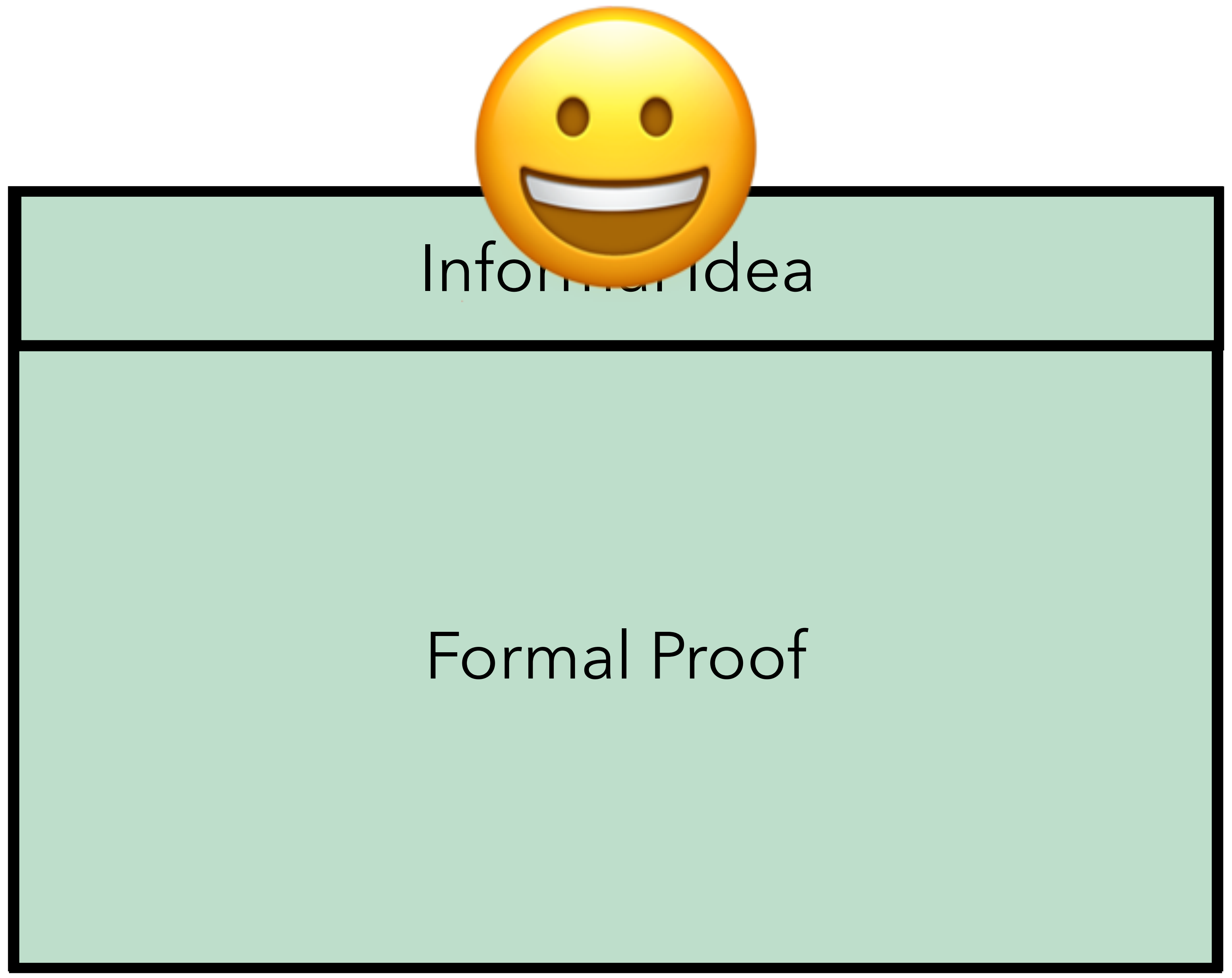
Theorem 1: 1+1+1=3

Proof:
By Lemma 1+1=2**…**
Next, we show 2+1=3**…**

Don't reference facts not stated as theorems/lemmas/etc.

Writing Dos and Don’ts

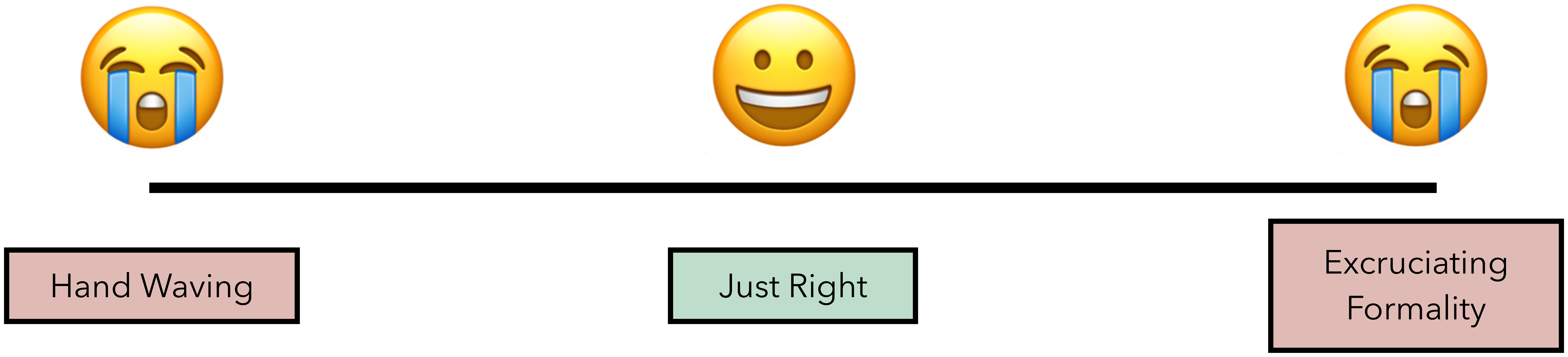
Intuition



Give intuition / an overview at the beginning of your proofs

Writing Dos and Don'ts

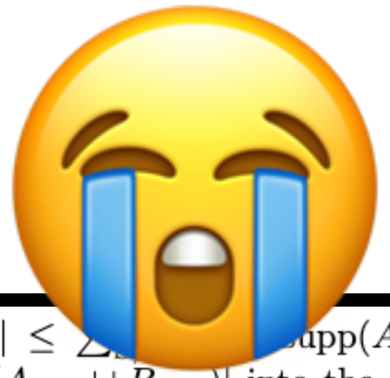
Intuition



Balance intuition and formality

Writing Dos and Don'ts

General Style



and so $\sum_{S,i}\sum_k |\mathrm{supp}(A_{Si,k} \cup B_{Si,k})| \le$ … $\mathrm{upp}(A_S^{(i)}) \le \tilde{O}(n^{1+O(\epsilon)} + N^{O(\epsilon)}L^2)$ Thus, plugging this bound on $\sum_{S,i,k} |\mathrm{supp}(A_{Si,k} \cup B_{Si,k})|$ into the guarantees of Theorem 10.4 and the fact that our pairs are $L \cdot N^{O(\epsilon)}$-batchable, we have that each time we compute a cutmatch in step 2, the total number of edges we add across all G_S for $S \in \mathcal{N}[h']$ for a fixed h' is at most $\tilde{O}(m + L \cdot N^{O(\epsilon)} + n^{1+O(\epsilon)} + N^{O(\epsilon)}L) = \tilde{O}(m + n^{1+O(\epsilon)} + L^2 \cdot N^{O(\epsilon)})$. Since we have $1/\epsilon$ iterations, it follows that the number of edges across all G_S for $S \in \mathcal{N}[h']$ is never more than $\frac{1}{\epsilon} \cdot \tilde{O}(m + n^{1+O(\epsilon)} + L^2 \cdot N^{O(\epsilon)})$. It follows that the work and depth to compute all cut strategies for all $S \in \mathcal{N}[h']$ for all $1/\epsilon$-many iterations and all $h' \le h \cdot \frac{1}{\epsilon} \cdot (s)^{O(1/\epsilon)}$ a power of 2 in step 2 are respectively $\frac{1}{\epsilon} \cdot \sum_i \mathsf{W}_{\text{cut-strat}}(A_i, m_i)$ and $\frac{1}{\epsilon} \cdot \max_i \mathsf{D}_{\text{cut-strat}}(A_i, m_i)$ where $|A_i| \le |A|/L$ for all i and $\sum_i m_i \le \tilde{O}(m + n^{1+O(\epsilon)} + L^2 \cdot N^{O(\epsilon)})$.

and so

$$\sum_{S,i}\sum_k |\mathrm{supp}(A_{Si,k} \cup B_{Si,k})| \le \sum_{S,i} N^{O\ldots} \ldots \mathrm{supp}(A_S^{(i)}) \le \tilde{O}(n^{1+O(\epsilon)} + N^{O(\epsilon)}L^2)$$

Thus, plugging this bound on $\sum_{S,i,k} |\mathrm{supp}(A_{Si,k} \cup B_{Si,k})|$ into the guarantees of Theorem 10.4 and the fact that our pairs are $L \cdot N^{O(\epsilon)}$-batchable, we have that each time we compute a cutmatch in step 2, the total number of edges we add across all G_S for $S \in \mathcal{N}[h']$ for a fixed h' is at most

$$\tilde{O}(m + L \cdot N^{O(\epsilon)} + n^{1+O(\epsilon)} + N^{O(\epsilon)}L) = \tilde{O}(m + n^{1+O(\epsilon)} + L^2 \cdot N^{O(\epsilon)}).$$

Since we have $1/\epsilon$ iterations, it follows that the number of edges across all G_S for $S \in \mathcal{N}[h']$ is never more than

$$\frac{1}{\epsilon} \cdot \tilde{O}(m + n^{1+O(\epsilon)} + L^2 \cdot N^{O(\epsilon)}).$$

It follows that the work and depth to compute all cut strategies for all $S \in \mathcal{N}[h']$ for all $1/\epsilon$-many iterations and all $h' \le h \cdot \frac{1}{\epsilon} \cdot (s)^{O(1/\epsilon)}$ a power of 2 in step 2 are respectively

$$\frac{1}{\epsilon} \cdot \sum_i \mathsf{W}_{\text{cut-strat}}(A_i, m_i) \quad (19)$$

and

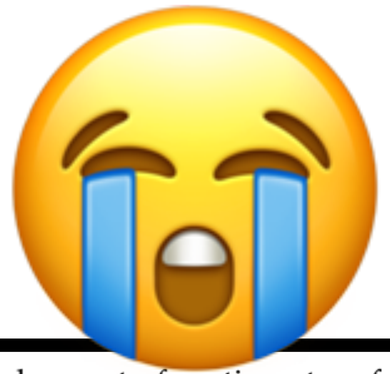
$$\frac{1}{\epsilon} \cdot \max_i \mathsf{D}_{\text{cut-strat}}(A_i, m_i) \quad (20)$$

where $|A_i| \le |A|/L$ for all i and $\sum_i m_i \le \tilde{O}(m + n^{1+O(\epsilon)} + L^2 \cdot N^{O(\epsilon)})$.

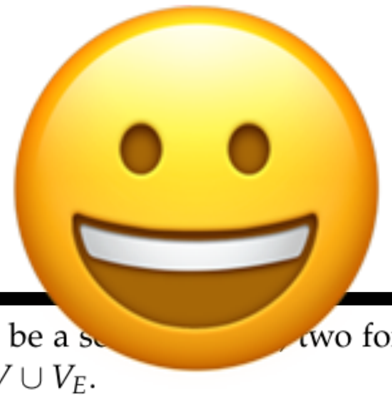
Use whitespace (align*s) generously

Writing Dos and Don'ts

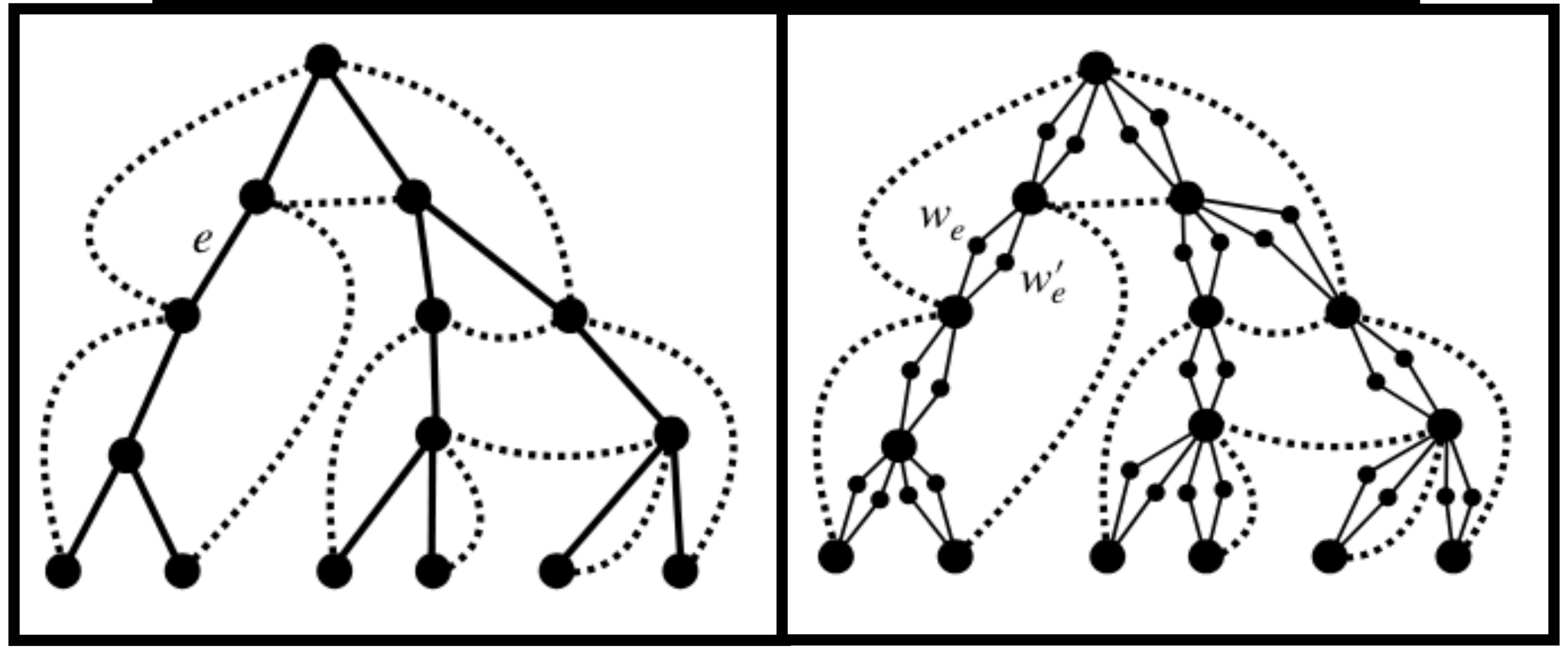
General Style



1. **Vertices:** Let $V_E := \{w_e, w'_e\}_{e \in E}$ be a set of vertices, two for each edge of E. The vertex set of our k-ECSM instance is $W := V \cup V_E$.
2. **Edges:** For each edge $e = \{u, v\} \in E$, we have 4 edges in our k-ECSM instance, namely $\{u, w_e\}$, $\{w_e, v\}$, $\{u, w'_e\}$, and $\{w'_e, v\}$. Let E_{Gadget} be all such edges. The edge set of our k-ECSM instance is $B := E_{\text{Gadget}} \cup L$.
3. **Costs:** The cost of each edge $b \in B$ in our k-ECSM instance is 1, i.e., $c_b = 1$.



1. **Vertices:** Let $V_E := \{w_e, w'_e\}_{e \in E}$ be a s… two for each edge of E. The vertex set of our k-ECSM instance is $W := V \cup V_E$.
2. **Edges:** For each edge $e = \{u, v\} \in E$, we have 4 edges in our k-ECSM instance, namely $\{u, w_e\}$, $\{w_e, v\}$, $\{u, w'_e\}$, and $\{w'_e, v\}$. Let E_{Gadget} be all such edges. The edge set of our k-ECSM instance is $B := E_{\text{Gadget}} \cup L$.
3. **Costs:** The cost of each edge $b \in B$ in our k-ECSM instance is 1, i.e., $c_b = 1$.



Use (a lot of) figures

Writing Dos and Don’ts

LaTeX Nits (many courtesy of Ryan O’Donnell)

```
Quoth the Raven "Nevermore".
```

Quoth the Raven ”Nevermore”.

```
Quoth the Raven ``Nevermore''.
```

Quoth the Raven “Nevermore”.

Writing Dos and Don'ts

LaTeX Nits (many courtesy of Ryan O'Donnell)

`Inner product $<x,y>$.`

Inner product $< x, y >$.

`Inner product $\langle x,y \rangle$.`

Inner product $\langle x, y \rangle$.

Writing Dos and Don'ts

LaTeX Nits (many courtesy of Ryan O'Donnell)

`$(\frac{x^2}{y}) \leq z$`

$$(\frac{x^2}{y}) \leq z$$

`$\left(\frac{x^2}{y}\right) \leq z$`

$$\left(\frac{x^2}{y}\right) \leq z$$

Writing Dos and Don'ts

LaTeX Nits (many courtesy of Ryan O'Donnell)

`$ALG(x) = log n$,`

$$ALG(x) = log n$$

`$\textsc{ALG}(x) = \log n$`

$$\textsc{ALG}(x) = \log n$$

Writing Dos and Don'ts

LaTeX Nits (many courtesy of Ryan O'Donnell)

`Let G be a k-connected graph.`

Let G be a k-connected graph.

`Let $G$ be a $k$-connected graph.`

Let G be a k-connected graph.

Writing Dos and Don'ts

LaTeX Nits (many courtesy of Ryan O'Donnell)

```
\begin{align}\label{eq}
  A & \leq B \\
  & \leq D
\end{align}
so $A \leq C$ by \ref{eq}.
```

We have

$$\begin{aligned} A &\leq B && (1) \\ &\leq C && (2) \end{aligned}$$

so $A \leq C$ by Equation 1.

```
\begin{align}\label{eq}
  A & \leq B \nonumber \\
  & \leq D
\end{align}
so $A \leq C$ by \ref{eq}.
```

We have

$$\begin{aligned} A &\leq B \\ &\leq C && (1) \end{aligned}$$

so $A \leq C$ by Equation 1.

Writing Dos and Don'ts

LaTeX Nits (many courtesy of Ryan O'Donnell)

😭

```
\begin{proof}
\begin{align}
  A & \leq B \\
  & \leq D
\end{align}
\end{proof}
```

Proof.

$$\begin{aligned} A &\leq B \\ &\leq D. \end{aligned}$$

□

😃

```
\begin{proof} We have
\begin{align}
  A & \leq B \\
  & \leq D \qedhere
\end{align}
\end{proof}
```

Proof. We have

$$\begin{aligned} A &\leq B \\ &\leq D. \end{aligned} \quad \square$$

Writing Dos and Don’ts

LaTeX Nits (many courtesy of Ryan O’Donnell)

```
Math is fun, e.g. algebra.
```

Math is fun, e.g. algebra.

```
Math is fun, e.g.\ algebra.
```

Math is fun, e.g. algebra.

Writing Dos and Don'ts

LaTeX Nits (many courtesy of Ryan O'Donnell)

😭

Math is fun, e.g. algebra.

😃

Math is fun, e.g.\ algebra.

Math is fun, e.g. algebra.

Math is fun, e.g. algebra.

🧐

Summary

Learning **Doing** **Writing**

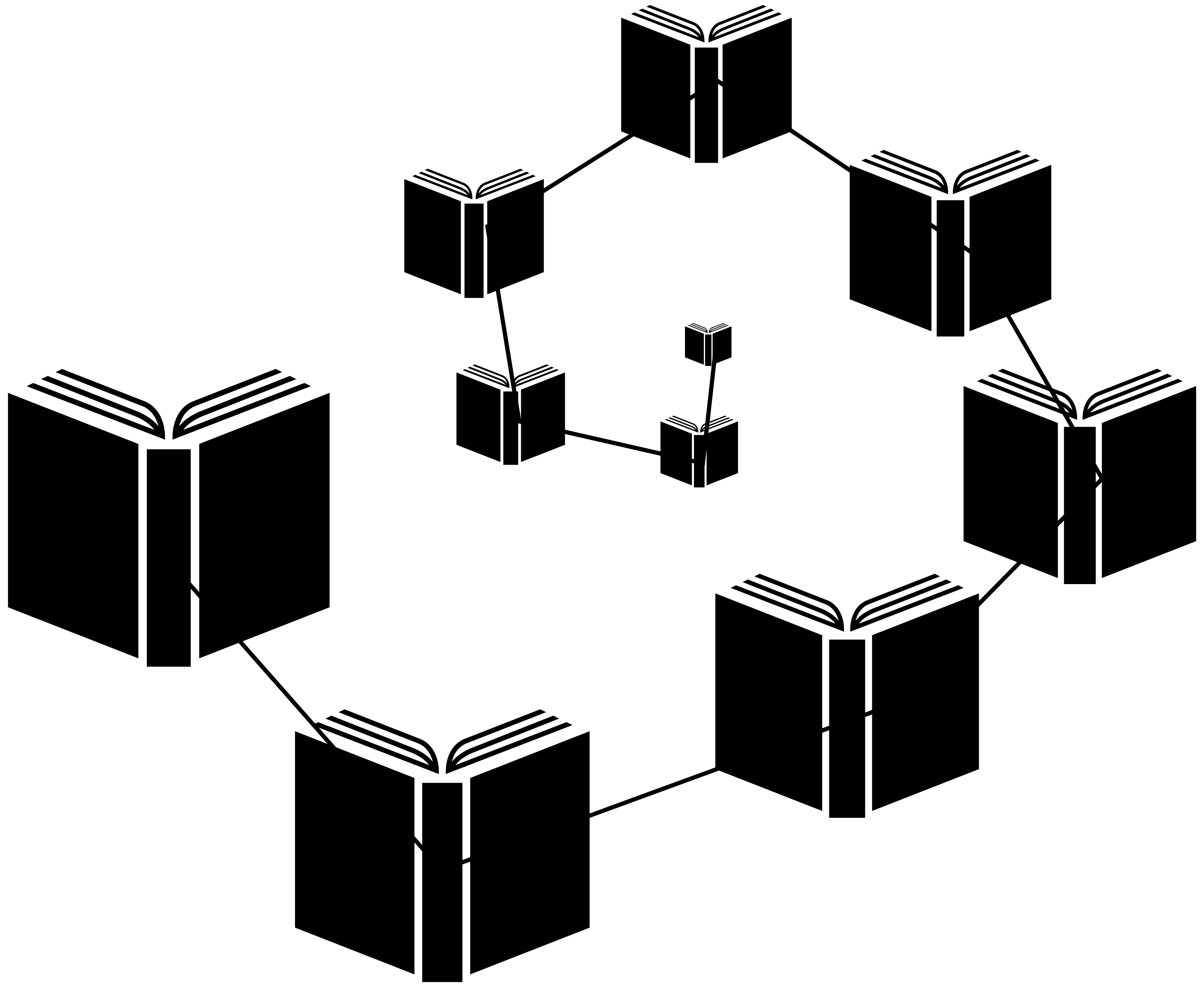
Simplification 🧠

Active 🚲

Why Do Theory

Why Do Theory

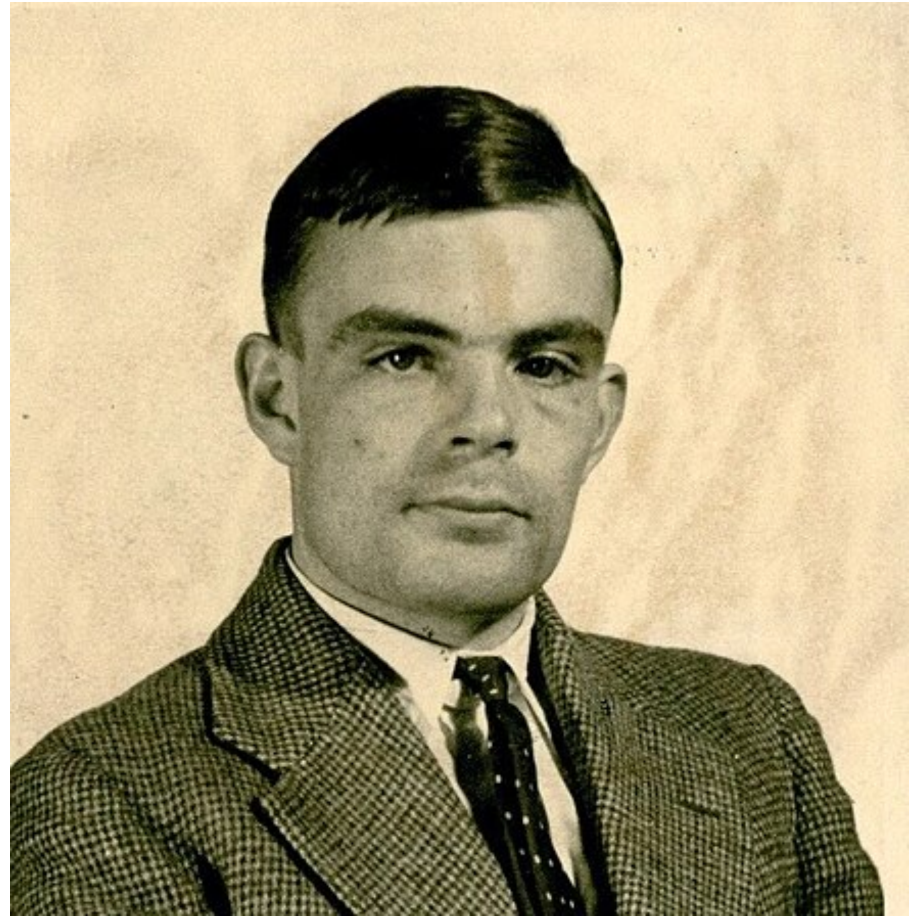
Infinite **learning opportunities** of beautiful facts



Why Do Theory

It's a **young** field (less to get up to speed with)

~2000 Years Ago

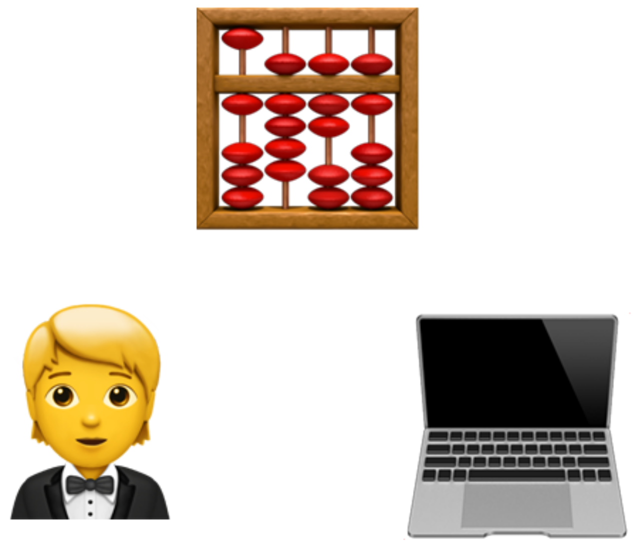


~100 Years Ago

Why Do Theory

Uniquely at the intersection of the **creative** and the **formal**

and (sometimes) the **practical**



Why Do Theory

Theory

+

" *"Music is the only **magic** left in this world."*

-Bob Dylan "

-My dad

1. guided by arcane laws
2. results often defy common sense
3. takes intense study to master