Algorithms for Learning Probabilistic and Causal Models with Possible Imperfect Advice

Doctoral Seminar

9 September 2024

Davin Choo

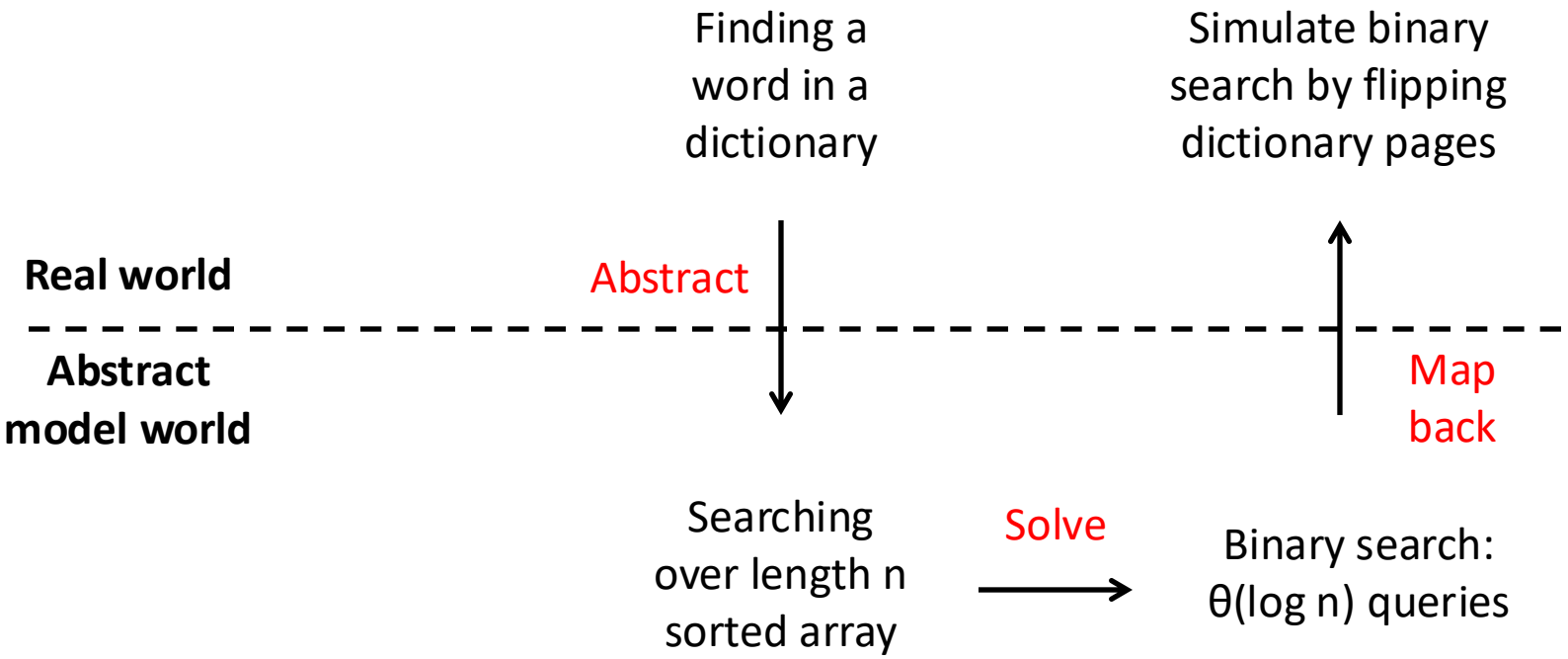
National University of Singapore

The world is complex

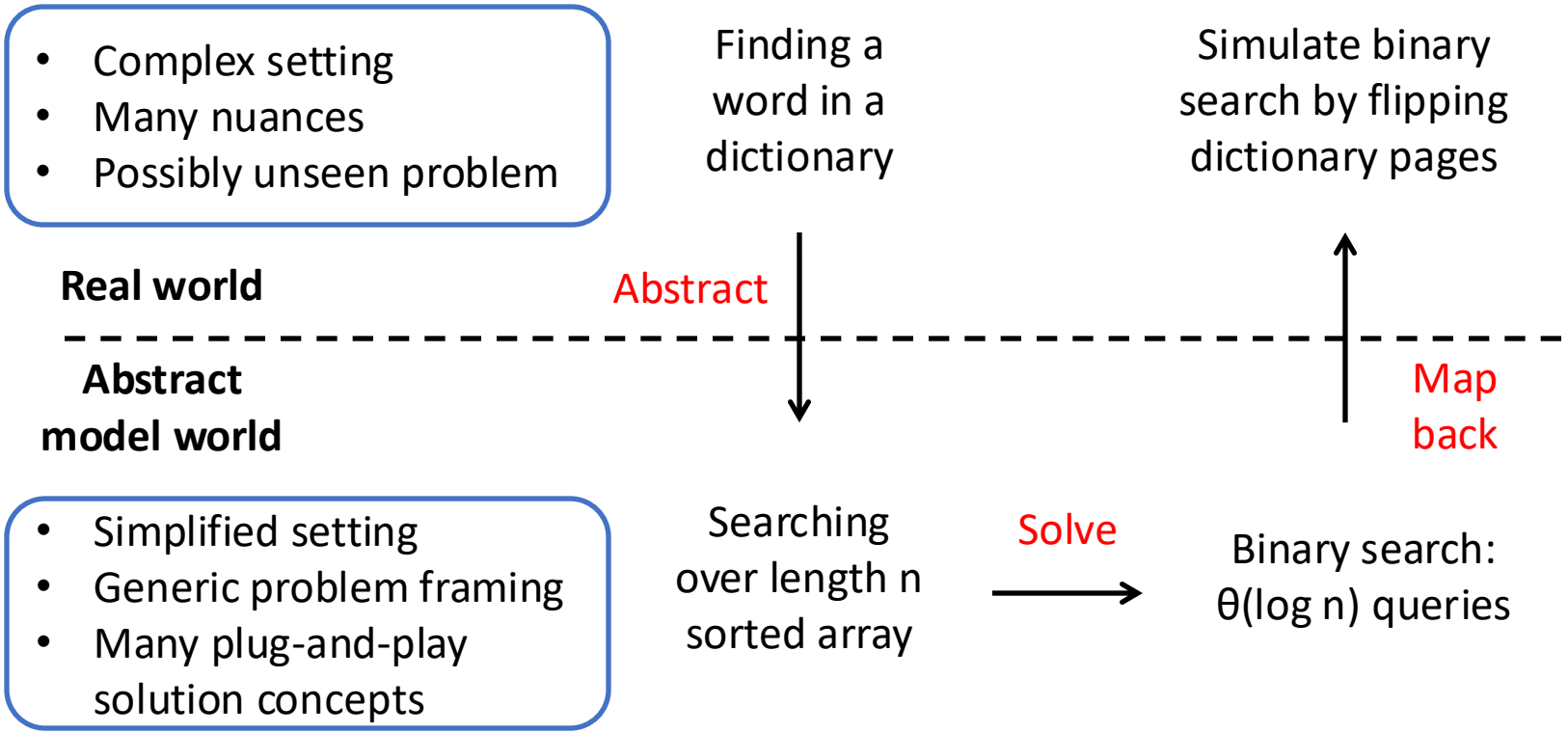
- **Learning a useful representation** of the world from data is a cornerstone of scientific discovery and the driving force behind successful modern machine learning methods
- This process often involves learning **probabilistic models for predictive tasks** and **causal models to understand interventional effects on systems**, which is crucial for informed downstream decision-making

https://huggingface.co/spaces/dalle-mini/dalle-mini on “A thoughtful person stands at the edge of a futuristic cityscape, filled with intricate technology, complex networks, and interconnected systems. They gaze intently at a holographic globe that displays dynamic data, charts, and code swirling around it, their expression a mix of curiosity and determination as they ponder how to interact and take action in this complex world”

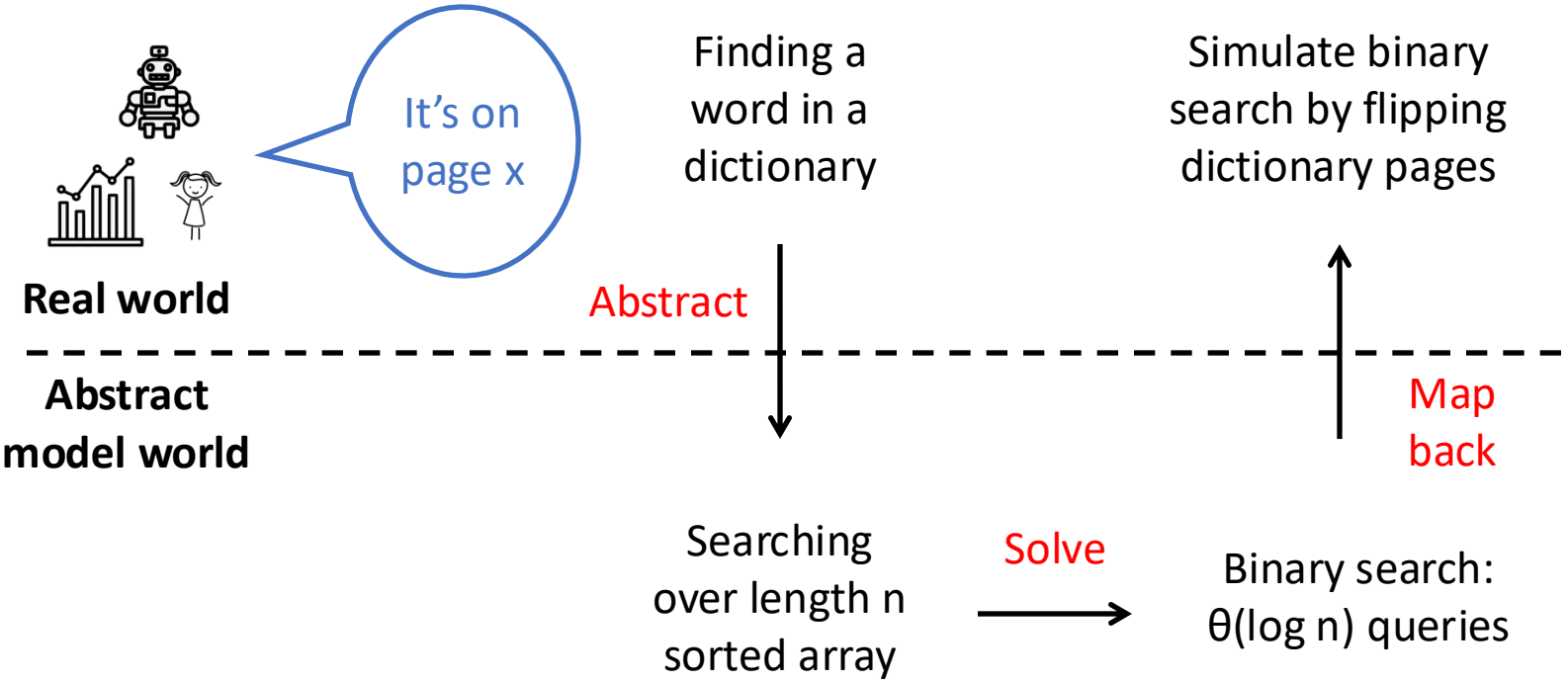
A general problem-solving framework



A general problem-solving framework

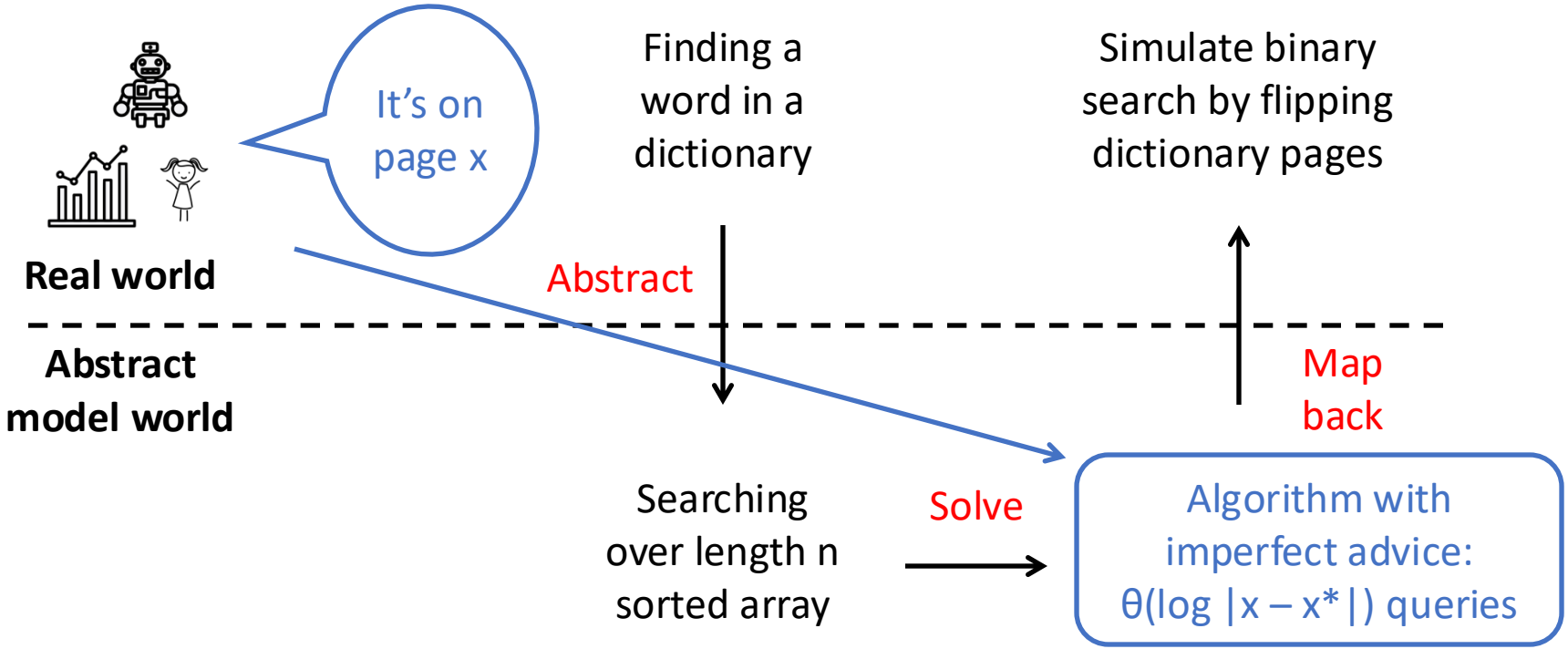


Side-information about problem instances

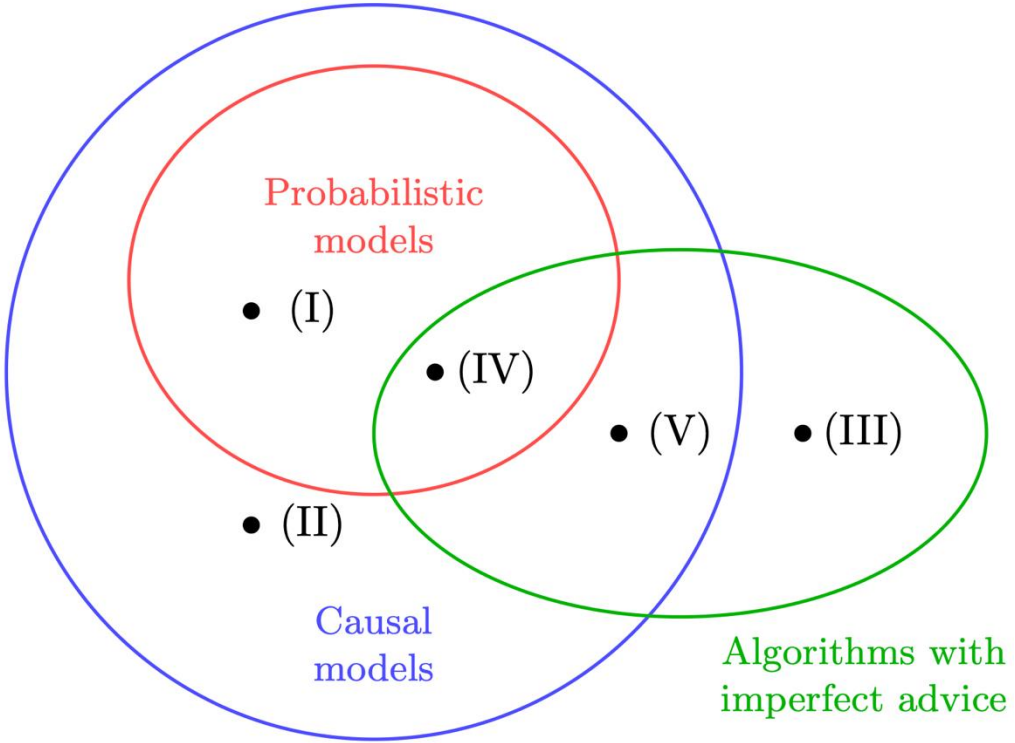


https://thenounproject.com/icon/statistics-7090732/
https://thenounproject.com/icon/girl-1257314/
https://thenounproject.com/icon/robot-7098785/

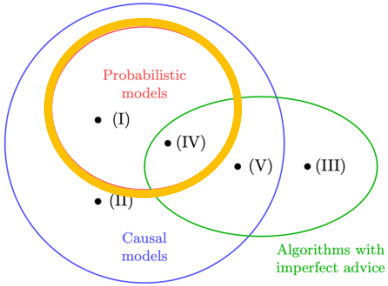
Side-information about problem instances



Main themes explored in my PhD

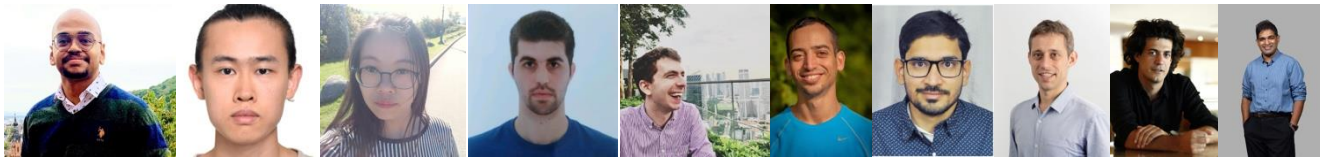


(I): Probabilistic models



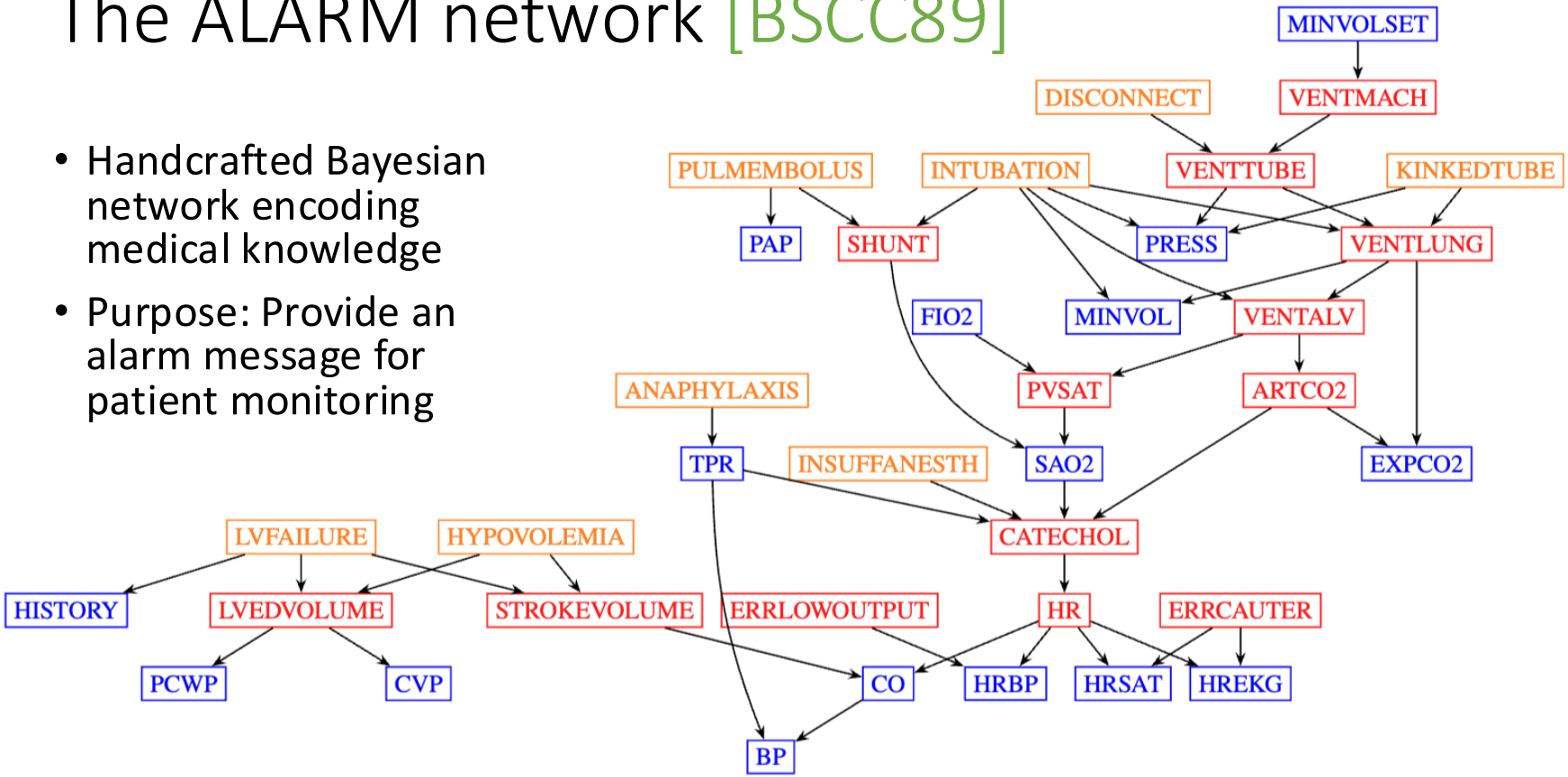
- Classic results in statistics show asymptotic convergence of estimators in the "infinite data" regime
- Probably Approximately Correct (PAC) learning model [Val84]
  - Given sample access to some underlying distribution $\mathcal{P}$, produce $\hat{\mathcal{P}}$ such that $\mathrm{TV}(\mathcal{P}, \hat{\mathcal{P}}) \leq \varepsilon$ with probability $\geq 1 - \delta$
- Bayesian networks [Pea88]
  - Probabilistic graphical model commonly used to model beliefs
  - $\approx 2^{n^2}$ candidate directed acyclic graphs (DAGs), one of which is $\mathcal{G}^*$

[Val84] Leslie G Valiant. *A theory of the learnable*. Communications of the ACM, 1984.
[Pea88] Judea Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann, 1988.



The ALARM network [BSCC89]

- Handcrafted Bayesian network encoding medical knowledge
- Purpose: Provide an alarm message for patient monitoring



[BSCC89] Ingo A Beinlich, Henri Jacques Suermondt, R Martin Chavez, and Gregory F Cooper. *The alarm monitoring system: A case study with two probabilistic inference techniques for belief network*. Second European Conference on Artificial Intelligence in Medicine (AIME), 1989

The ALARM network [BSCC89]

A sample consultation

ALARM is a data-driven system. Simulating an anesthesia monitor, ALARM accepts a set of physiologic measurements. An example would be as follows: blood pressure 120/80 mmHg, heart rate 80/min, inspired oxygen concentration 50%, tidal volume 500 ml, respiratory rate 10/min, breathing pressure 50 mbar, and measured minute ventilation 1.2 l/min. These measurements are categorized into 'low', 'normal', 'high', etc. and text messages are generated when measurements are outside of their normal range. These messages will then appear in the *Warning* and *Caution* fields of the monitor depending on their importance *(Fig. 3)*. In the given example, the high breathing pressure of 50 mbar imposes a direct danger to the patient and a warning is issued. The low minute ventilation is less immediate and is displayed as a caution only.

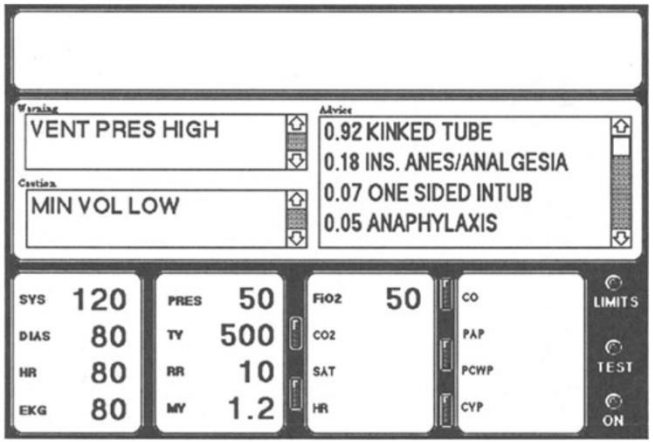
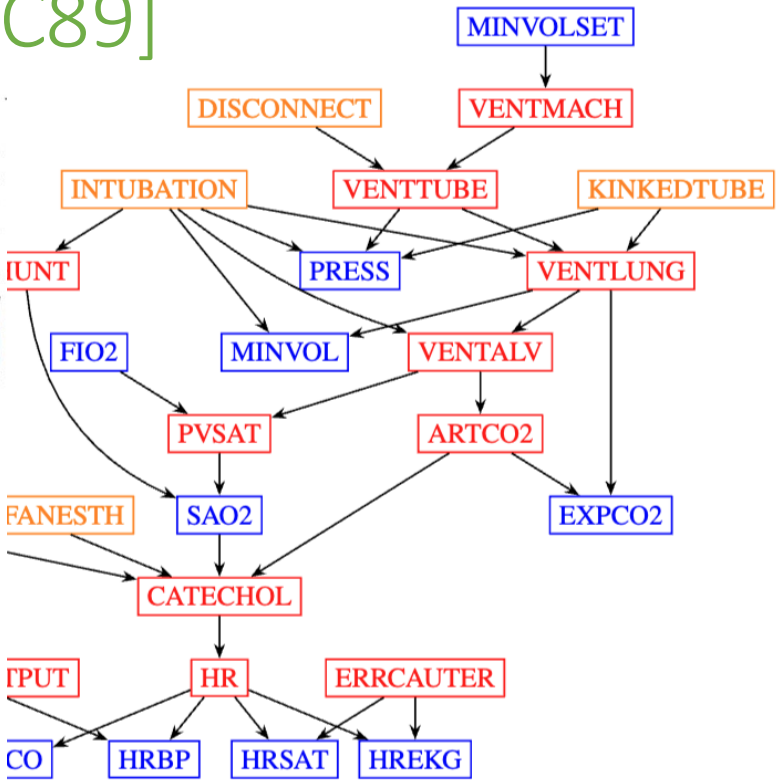


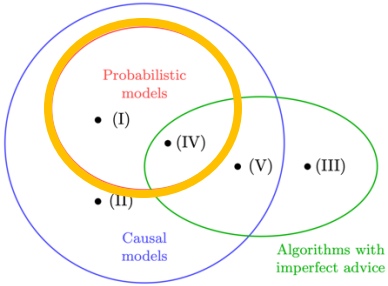
Fig. 3

ALARM simulates an anesthesia monitor. It takes patient measurements, displays warning and caution messages, and lists a differential diagnosis.



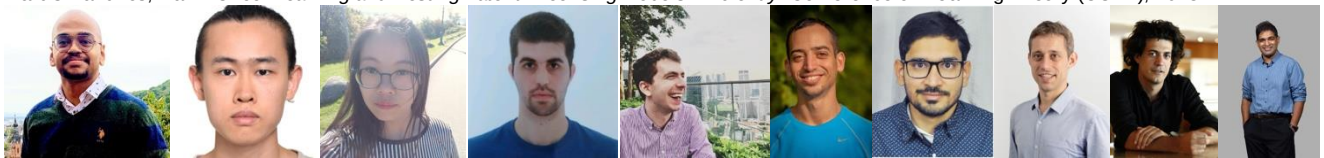
[BSCC89] Ingo A Beinlich, Henri Jacques Suermondt, R Martin Chavez, and Gregory F Cooper. *The alarm monitoring system: A case study with two probabilistic inference techniques for belief network*. Second European Conference on Artificial Intelligence in Medicine (AIME), 1989

(I): Probabilistic models

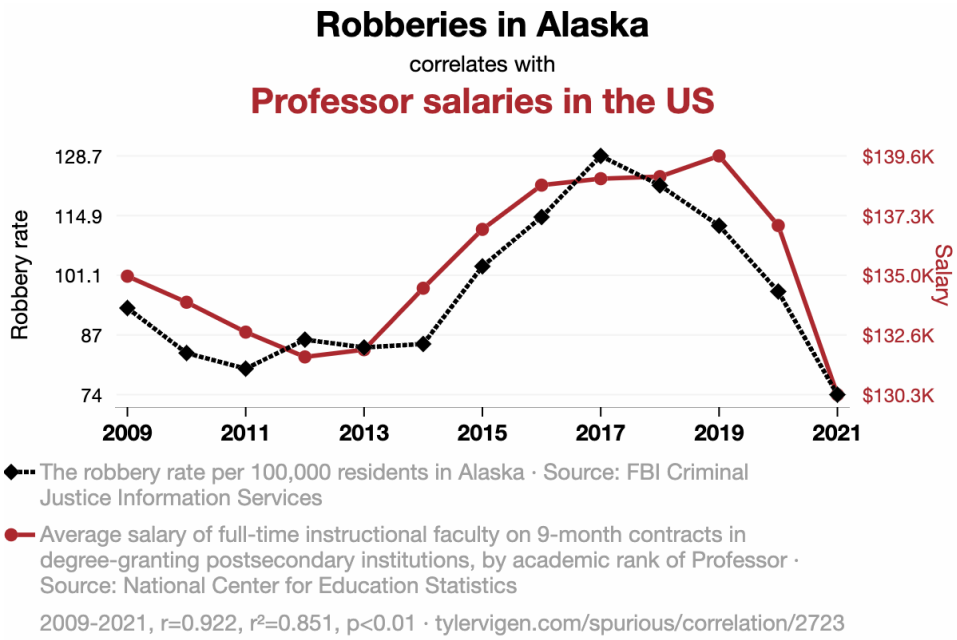


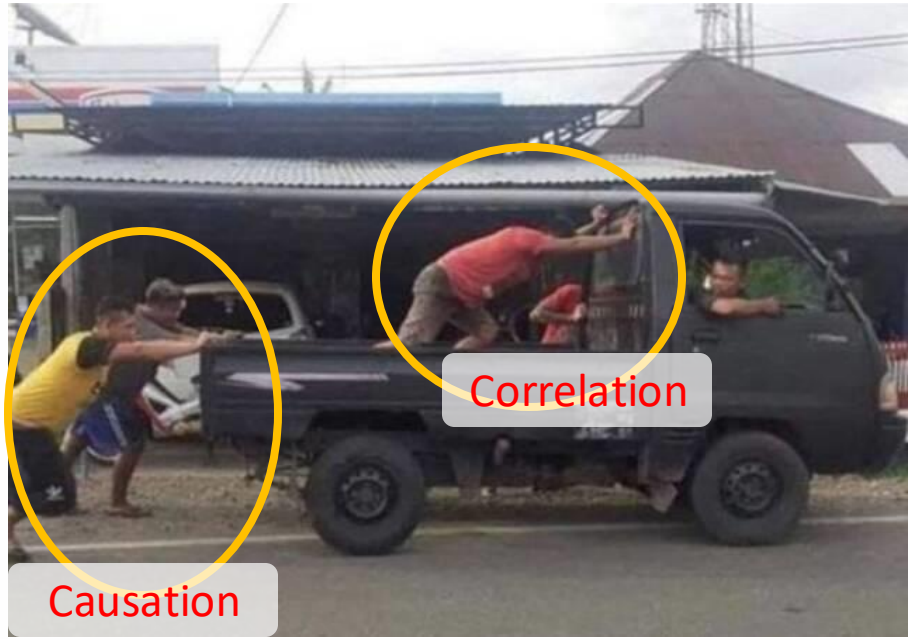
- Suppose data distribution $\mathcal{P}$ is described by Bayesian network
  - NP-hard to find "score maximizing" DAG from data [Chi96] and to decide whether $\mathcal{P}$ can be described by a DAG with p parameters [CHM04]
  - Even under the promise that $\mathcal{P}$ can be described by a DAG with p parameters, it is NP-hard to find such a parameter-bounded DAG [BCGM24]
  - We also have some PAC-style finite sample results in learning the structure and parameters of Bayesian network for $\mathcal{P}$ [BCG+22, DDKC23, CYBC24]
  - Insight: If network's in-degree is bounded, we can use less samples

[Chi96] David Maxwell Chickering. *Learning Bayesian networks is NP-complete*. Lecture Notes in Statistics, vol 112, 1996
[CHM04] Max Chickering, David Heckerman, and Chris Meek. *Large-sample learning of Bayesian networks is NP-hard*. Journal of Machine Learning Research (JMLR), 2004
[BCGM24] Arnab Bhattacharyya, Davin Choo, Sutanu Gayen, Dimitrios Myrisiotis. *Learnability of Parameter-Bounded Bayes Nets*. Structured Probabilistic Inference & Generative Modeling (ICML Workshop), 2024
[BCG+22] Arnab Bhattacharyya, Davin Choo, Rishikesh Gajjala, Sutanu Gayen, Yuhao Wang. *Learning Sparse Fixed-Structure Gaussian Bayesian Networks*. International Conference on Artificial Intelligence and Statistics (AISTATS), 2024
[DDKC23] Yuval Dagan, Constantinos Daskalakis, Anthimos-Vardis Kandiros, Davin Choo. *Learning and Testing Latent-Tree Ising Models Efficiently*. Conference on Learning Theory (COLT), 2023
[CYBC24] Davin Choo, Joy Qiping Yang, Arnab Bhattacharyya, Clément L. Canonne. *Learning bounded degree polytrees with samples*. International Conference on Algorithmic Learning Theory (ALT), 2024



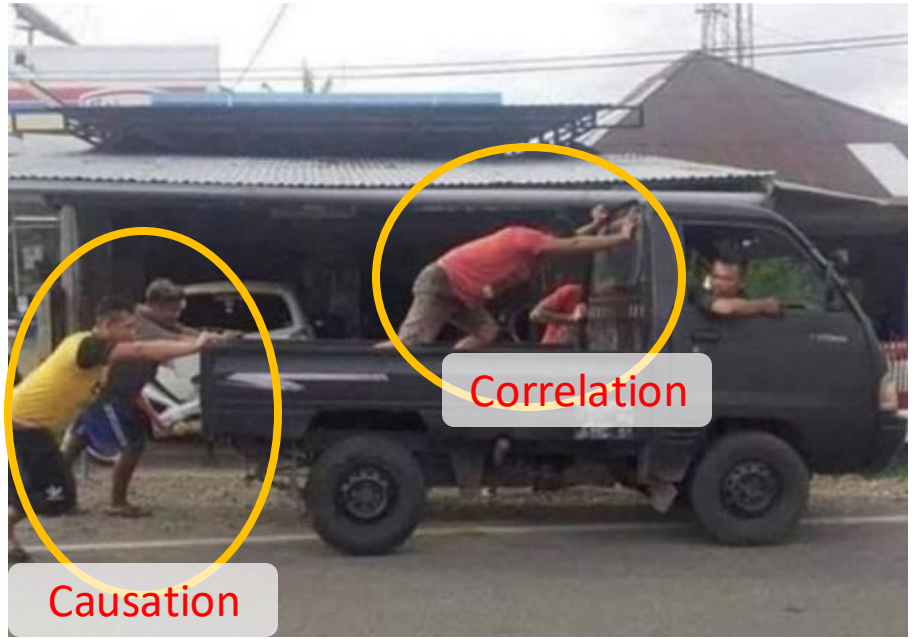
Correlation does not imply causation



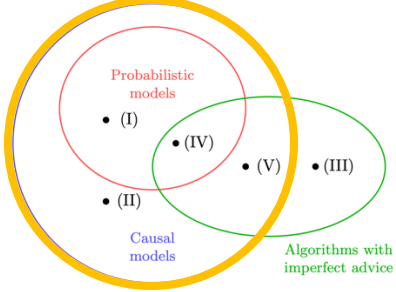


https://tylervigen.com/spurious/correlation/2723_robberies-in-alaska_correlates-with_professor-salaries-in-the-us
https://imgflip.com/memetemplate/327010423/Truck-teamwork

Correlation does not imply causation

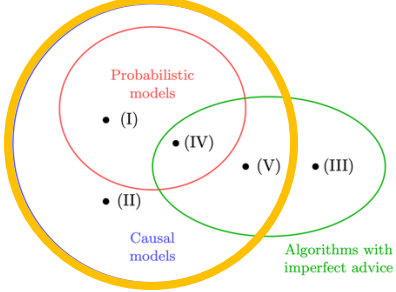


(II): Causal models



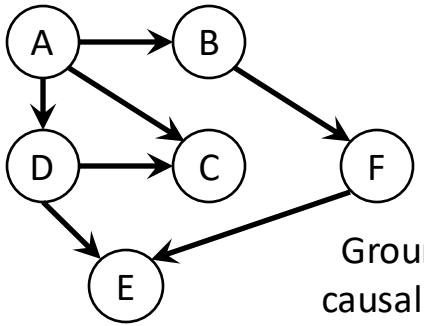
- Two fundamental problems in causal inference
  - Causal graph discovery: Recover true causal graph $\mathcal{G}^*$
  - Causal effect estimation: Estimate $\mathcal{P}(Y = y \mid do(X = x))$

(II): Causal models

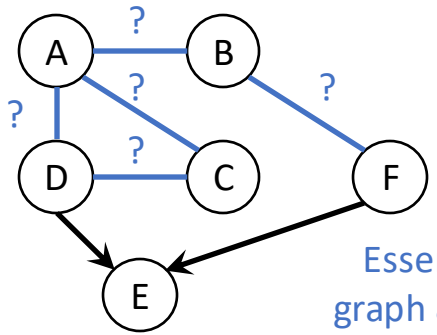


- Two fundamental problems in causal inference
  - Causal graph discovery: Recover true causal graph $\mathcal{G}^*$
    - Even with infinite observational data, can only determine causal graph up to some equivalence class where all conditional independence relations agree
    - Make distributional/structural assumptions or perform interventions/experiments!
  - Causal effect estimation: Estimate $\mathcal{P}(Y = y \mid do(X = x))$

Causal discovery via interventions



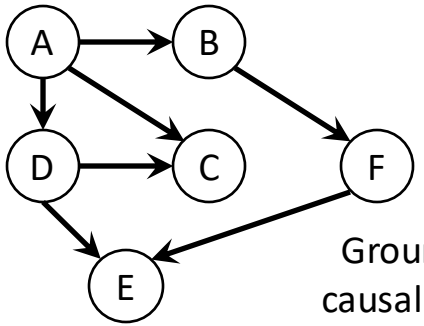
Ground truth causal graph $\mathcal{G}^*$



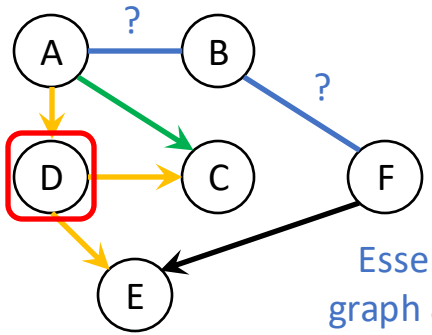
Essential graph $\mathcal{E}(\mathcal{G}^*)$

- Want: Recover $\mathcal{G}^*$ starting from partially oriented $\mathcal{E}(\mathcal{G}^*)$ from observational data

Causal discovery via interventions



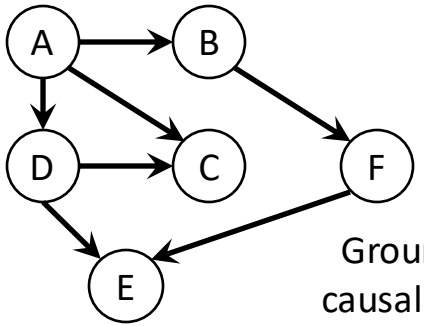
Ground truth causal graph $\mathcal{G}^*$



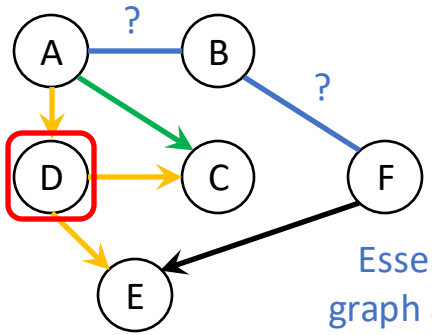
Essential graph $\mathcal{E}(\mathcal{G}^*)$

- Want: Recover $\mathcal{G}^*$ starting from partially oriented $\mathcal{E}(\mathcal{G}^*)$ from observational data
- Interventions reveal arc orientations (incident arcs + Meek rules)
- **Goal: Recover $\mathcal{G}^*$ using as few interventions as possible**

Causal discovery via interventions



Ground truth causal graph $\mathcal{G}^*$

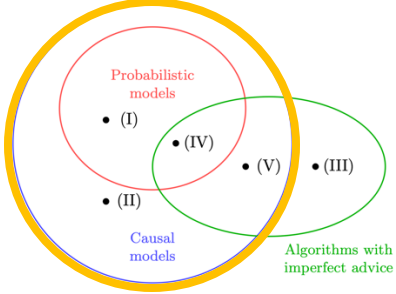


Essential graph $\mathcal{E}(\mathcal{G}^*)$

- Want: Recover $\mathcal{G}^*$ starting from partially oriented $\mathcal{E}(\mathcal{G}^*)$ from observational data
- Interventions reveal arc orientations (incident arcs + Meek rules)
- **Goal: Recover $\mathcal{G}^*$ using as few interventions as possible**
- We have some results regarding how to design algorithms to perform optimal adaptive interventions under various scenarios [CSB22, CS23a, CGB23, CS23b, CS23c, CSU24]
- Insight: Can abstract and treat this as a graph problem with specialized causal operations

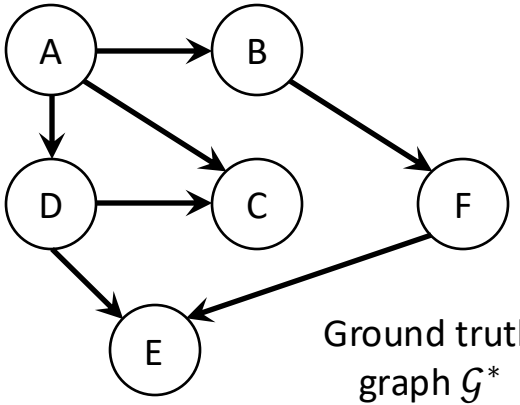
[CSB22] Davin Choo, Kirankumar Shiragur, Arnab Bhattacharyya. *Verification and search algorithms for causal DAGs*. Conference on Neural Information Processing Systems (NeurIPS), 2022.
[CS23a] Davin Choo, Kirankumar Shiragur. *Subset verification and search algorithms for causal DAGs*. International Conference on Artificial Intelligence and Statistics (AISTATS), 2023.
[CGB23] Davin Choo, Themistoklis Gouleakis, Arnab Bhattacharyya. *Active causal structure learning with advice*. International Conference on Machine Learning (ICML), 2023.
[CS23b] Davin Choo, Kirankumar Shiragur. *New metrics and search algorithms for weighted causal DAGs*. International Conference on Machine Learning (ICML), 2023.
[CS23c] Davin Choo, Kirankumar Shiragur. *Adaptivity Complexity for Causal Graph Discovery*. Conference on Uncertainty in Artificial Intelligence (UAI), 2023.
[CSU24] Davin Choo, Kirankumar Shiragur, Caroline Uhler. *Causal discovery under off-target interventions*. International Conference on Artificial Intelligence and Statistics (AISTATS), 2024.

(II): Causal models

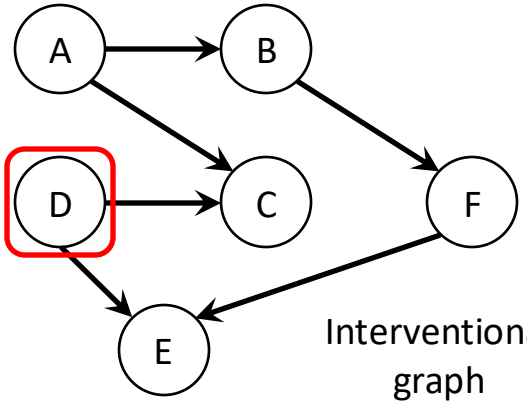


- Two fundamental problems in causal inference
  - Causal graph discovery: Recover true causal graph $\mathcal{G}^*$
    - Even with infinite observational data, can only determine causal graph up to some equivalence class where all conditional independence relations agree
    - Make distributional/structural assumptions or perform interventions/experiments!
  - Causal effect estimation: Estimate $\mathcal{P}(Y = y \mid do(X = x))$
    - Typically, a 2-stage process: learn $\mathcal{G}^*$, then apply closed-form formulas

Causal identification (the 2nd step)



Ground truth graph $\mathcal{G}^*$



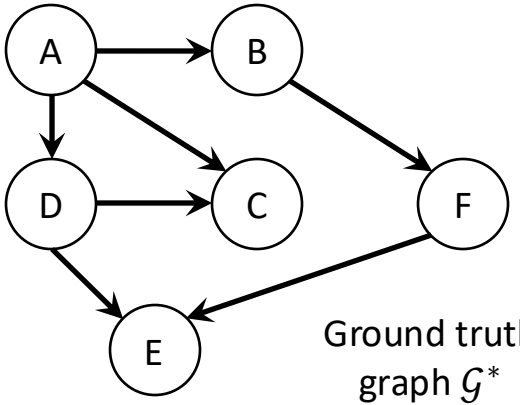
Interventional graph

$$\mathcal{P}(E = e \mid do(D = d^*))$$

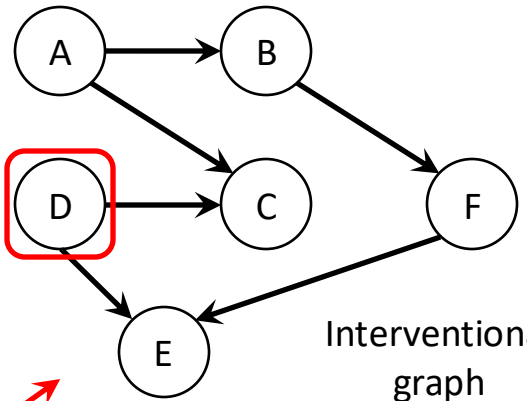
Interventional query

What is probability of $E = e$ when we fix $D = d^*$?

Causal identification (the 2nd step)



Ground truth graph $\mathcal{G}^*$



Interventional graph

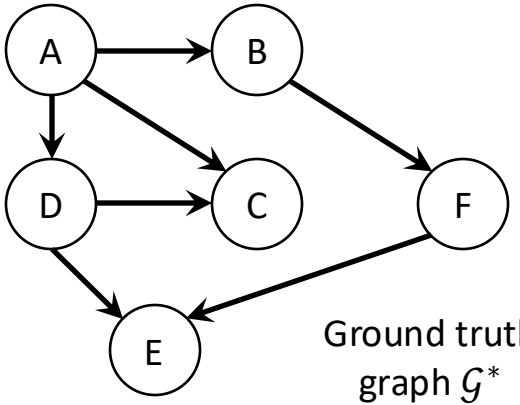
$$\mathcal{P}(E = e \mid do(D = d^*)) = \mathcal{P}(e \mid do(d^*)) \neq \mathcal{P}(e \mid d^*) \text{ in general}$$

Interventional query

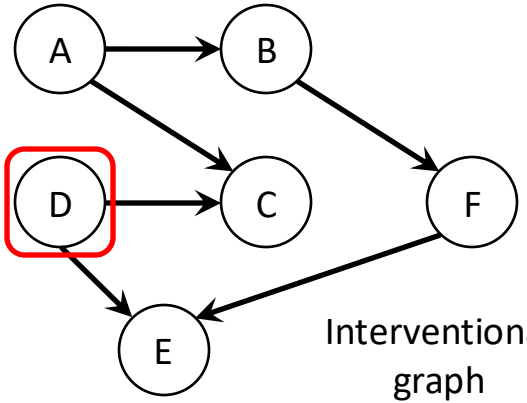
What is probability of $E = e$ when we fix $D = d^*$?

Need to draw samples from interventional graph, i.e., perform experiment and measure

Causal identification (the 2nd step)



Ground truth graph $\mathcal{G}^*$



Interventional graph

Because of structure of $\mathcal{G}^*$

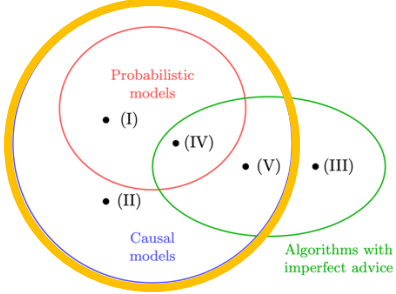
$$\mathcal{P}(E = e \mid do(D = d^*)) = \mathcal{P}(e \mid do(d^*)) = \int \mathcal{P}(e \mid d^*, a) \cdot \mathcal{P}(a)\, da$$

Interventional query

What is probability of $E = e$ when we fix $D = d^*$?

Just observational terms!

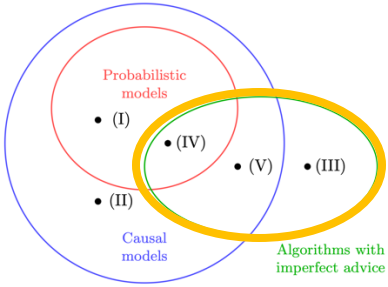
(II): Causal models



- Two fundamental problems in causal inference
  - Causal graph discovery: Recover true causal graph $\mathcal{G}^*$
    - Even with infinite observational data, can only determine causal graph up to some equivalence class where all conditional independence relations agree
    - Make distributional/structural assumptions or perform interventions/experiments!
  - Causal effect estimation: Estimate $\mathcal{P}(Y = y \mid do(X = x))$
    - Typically, a 2-stage process: learn $\mathcal{G}^*$, then apply closed-form formulas
    - [CSBS24] This is suboptimal as it may require strong assumptions and a lot of samples
    - Insight: "weak edges" shouldn't affect much for PAC-style results

[CSBS24] Davin Choo, Chandler Squires, Arnab Bhattacharyya, and David Sontag. *Causal effect estimation via covariate adjustments without knowing an equivalent causal graph or relying on faithfulness.* In preparation, 2024.

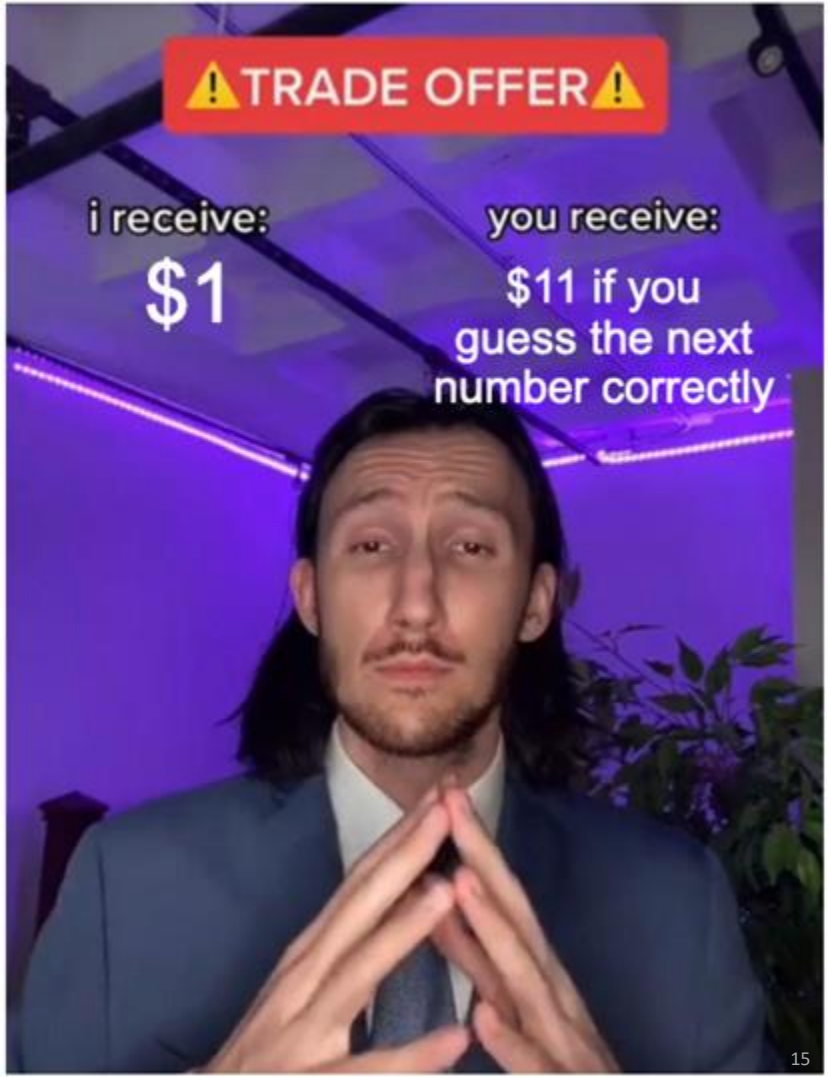
(III/IV/V): Algorithms with advice



- Two key performance measures
  - Consistency: If advice is “perfect”, how good are things?
  - Robustness: If advice is “garbage”, how bad are things?
- Challenge: We don’t know how good the given advice is a priori!

Detour: Let's make a deal

- There are 10 numbers in the universe U = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
- There is an underlying process $\mathcal{P}$ that generates IID samples from U
  - I.e., We can observe a sequence such as 1, 6, 3, 6, 2, 8, 0, 3, 9, 5, 4, …
- What property of $\mathcal{P}$ will make this deal profitable in expectation?



https://imgflip.com/memegenerator/309868304/Trade-Offer

Detour: Property testing land

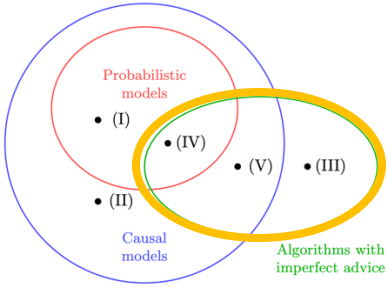
- How to test if $\mathcal{P}$ is the uniform distribution over U?
  - Say, we only care about constant success probability (can be amplified)
- Learning a ε-close $\hat{\mathcal{P}}$ then check: $\Theta\left(\frac{|U|}{\varepsilon^2}\right)$ IID samples from $\mathcal{P}$
- Uniformity testing requires $\Theta\left(\frac{\sqrt{|U|}}{\varepsilon^2}\right)$ IID samples from $\mathcal{P}$
  - If $\mathcal{P}$ is uniform, output YES w.p. $\geq \frac{2}{3}$
  - If $\mathcal{P}$ is ε-far from uniform, output NO w.p. $\geq \frac{2}{3}$
  - Many existing proofs for this bound. E.g., look at collisions in samples

→ Allowed to output arbitrarily if not uniform, yet not "far from uniform"

- See also [Can22] for an excellent property testing survey

[Can22] Clément L. Canonne. *Topics and Techniques in Distribution Testing: A Biased but Representative Sample*. Foundations and Trends® in Communications and Information Theory, 2022.

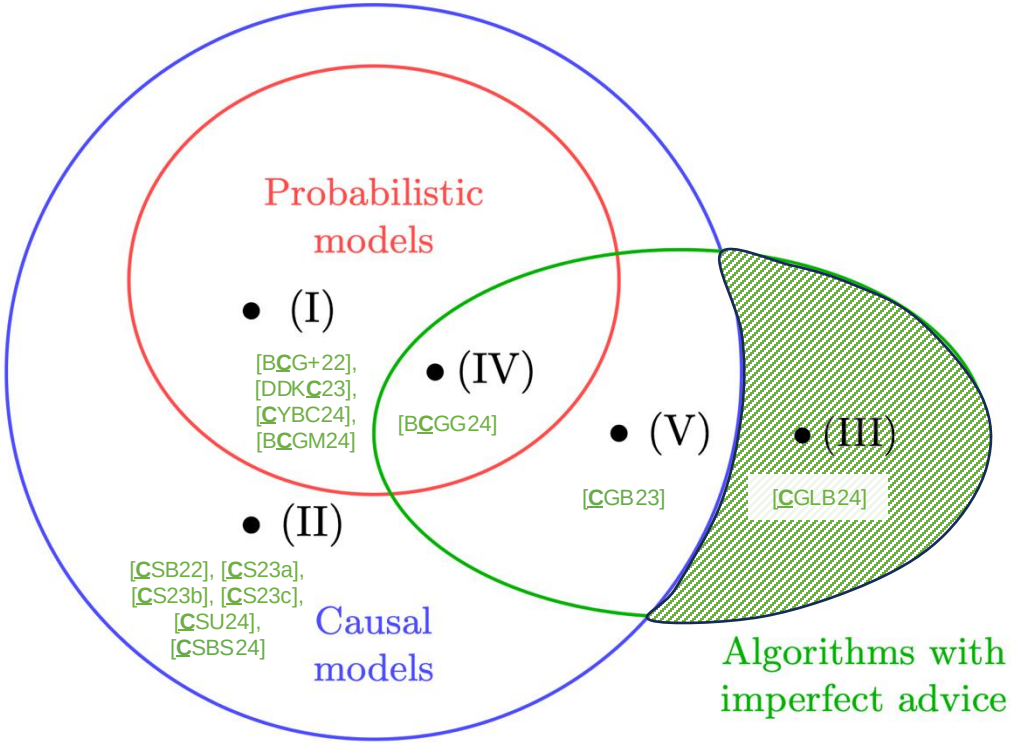
(III/IV/V): Algorithms with advice



- Two key performance measures
  - Consistency: If advice is “perfect”, how good are things?
  - Robustness: If advice is “garbage”, how bad are things?
- Challenge: We don’t know how good the given advice is a priori!
- Insight: “Testing can be cheaper than learning” → TestAndAct
  - [CGLB24] TestAndMatch: Improve competitive ratio of online bipartite matching (III)
  - [BCGG24] TestAndScheffe: Improve sample complexity of learning multivariate Gaussians (IV)
  - [CGB23] TestAndSubsetSearch: Reduce num of interventions required for causal graph discovery (V)

[CGLB24] Davin Choo, Themistoklis Gouleakis, Chun Kai Ling, and Arnab Bhattacharyya. *Online bipartite matching with imperfect advice*. International Conference on Machine Learning (ICML), 2024.
[BCGG24] Arnab Bhattacharyya, Davin Choo, Philips George John, and Themistoklis Gouleakis. *Learning multivariate Gaussians with imperfect advice*. Under submission to Innovations in Theoretical Computer Science (ITCS), 2024.
[CGB23] Davin Choo, Themistoklis Gouleakis, Arnab Bhattacharyya. *Active causal structure learning with advice*. International Conference on Machine Learning (ICML), 2023.

For the rest of this talk

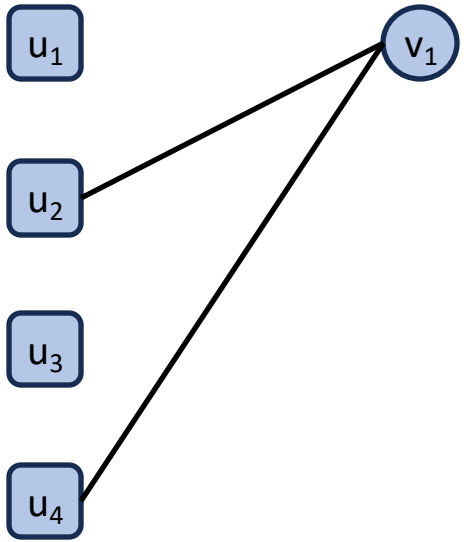


Online bipartite matching

- Offline set $U = \{u_1, \ldots, u_n\}$ fixed and known
- Online set $V = \{v_1, \ldots, v_n\}$ arrive one by one

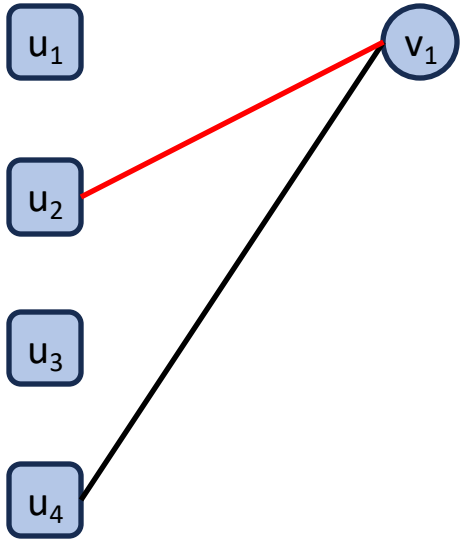
Online bipartite matching

- Offline set $U = \{u_1, \ldots, u_n\}$ fixed and known
- Online set $V = \{v_1, \ldots, v_n\}$ arrive one by one
- When an online vertex v_i arrives
  - Its neighbors $N(v_i)$ are revealed
  - We must make an irrevocable decision whether, and how, to match v_i to something in $N(v_i)$



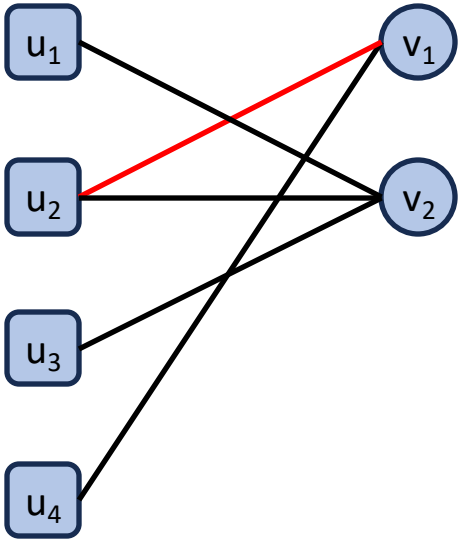
Online bipartite matching

- Offline set $U = \{u_1, \ldots, u_n\}$ fixed and known
- Online set $V = \{v_1, \ldots, v_n\}$ arrive one by one
- When an online vertex v_i arrives
  - Its neighbors $N(v_i)$ are revealed
  - We must make an irrevocable decision whether, and how, to match v_i to something in $N(v_i)$



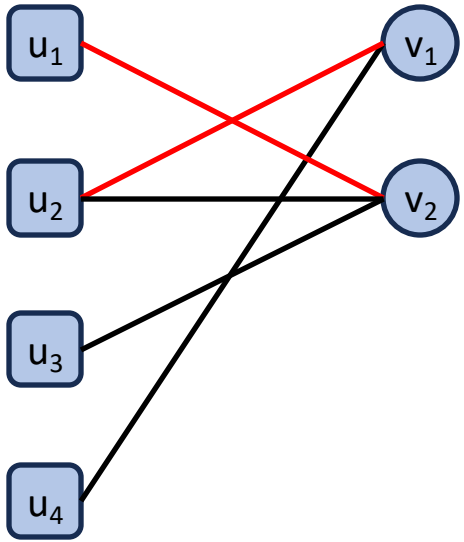
Online bipartite matching

- Offline set $U = \{u_1, \ldots, u_n\}$ fixed and known
- Online set $V = \{v_1, \ldots, v_n\}$ arrive one by one
- When an online vertex v_i arrives
  - Its neighbors $N(v_i)$ are revealed
  - We must make an irrevocable decision whether, and how, to match v_i to something in $N(v_i)$



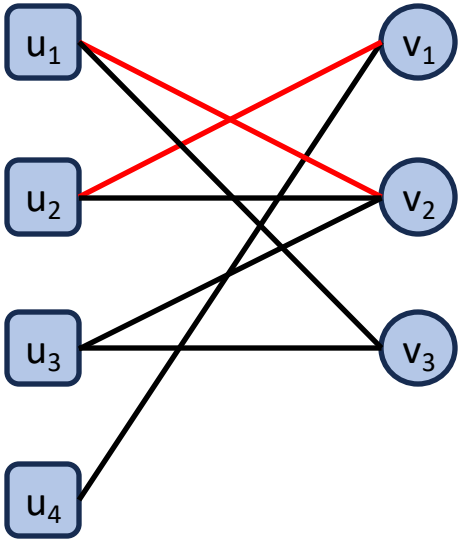
Online bipartite matching

- Offline set $U = \{u_1, \ldots, u_n\}$ fixed and known
- Online set $V = \{v_1, \ldots, v_n\}$ arrive one by one
- When an online vertex v_i arrives
  - Its neighbors $N(v_i)$ are revealed
  - We must make an irrevocable decision whether, and how, to match v_i to something in $N(v_i)$



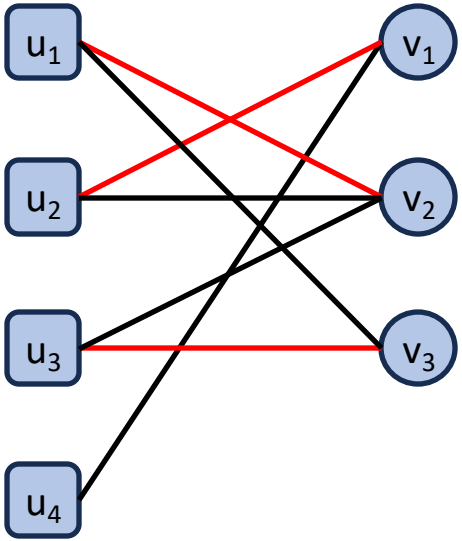
Online bipartite matching

- Offline set $U = \{u_1, \ldots, u_n\}$ fixed and known
- Online set $V = \{v_1, \ldots, v_n\}$ arrive one by one
- When an online vertex v_i arrives
  - Its neighbors $N(v_i)$ are revealed
  - We must make an irrevocable decision whether, and how, to match v_i to something in $N(v_i)$



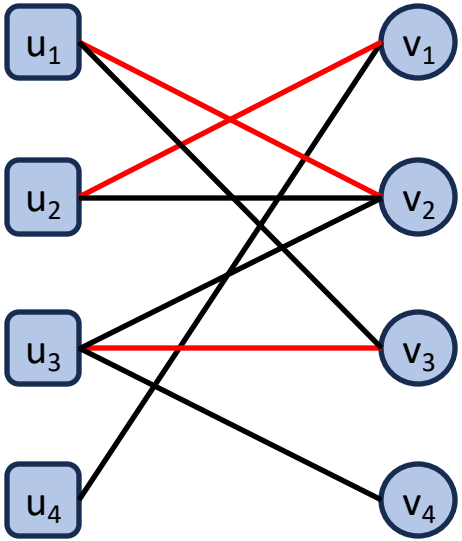
Online bipartite matching

- Offline set $U = \{u_1, \ldots, u_n\}$ fixed and known
- Online set $V = \{v_1, \ldots, v_n\}$ arrive one by one
- When an online vertex v_i arrives
  - Its neighbors $N(v_i)$ are revealed
  - We must make an irrevocable decision whether, and how, to match v_i to something in $N(v_i)$



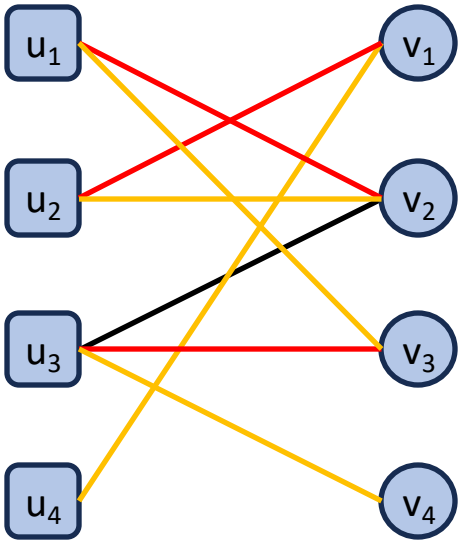
Online bipartite matching

- Offline set $U = \{u_1, \ldots, u_n\}$ fixed and known
- Online set $V = \{v_1, \ldots, v_n\}$ arrive one by one
- When an online vertex v_i arrives
  - Its neighbors $N(v_i)$ are revealed
  - We must make an irrevocable decision whether, and how, to match v_i to something in $N(v_i)$



Online bipartite matching

- Offline set $U = \{u_1, \dots, u_n\}$ fixed and known
- Online set $V = \{v_1, \dots, v_n\}$ arrive one by one
- When an online vertex v_i arrives
  - Its neighbors $N(v_i)$ are revealed
  - We must make an irrevocable decision whether, and how, to match v_i to something in $N(v_i)$
- Final offline graph $G^* = (U \cup V, E)$
  - $E = N(v_1) \cup \cdots \cup N(v_n)$
  - Maximum matching $M^* \subseteq E$ of size $|M^*| = n^* \leq n$



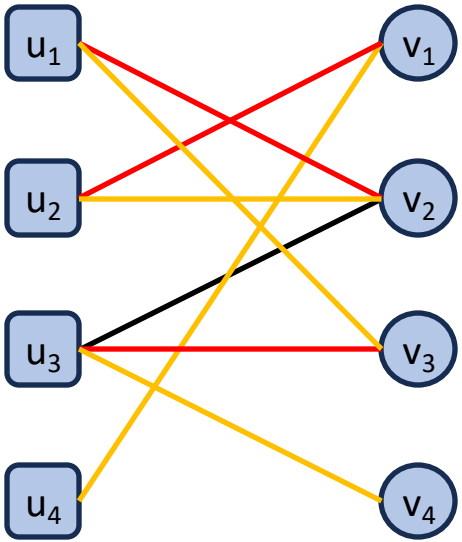
Online bipartite matching

- Offline set $U = \{u_1, \ldots, u_n\}$ fixed and known
- Online set $V = \{v_1, \ldots, v_n\}$ arrive one by one
- Final offline graph $G^* = (U \cup V, E)$
  - $E = N(v_1) \cup \cdots \cup N(v_n)$
  - Maximum matching $M^* \subseteq E$ of size $|M^*| = n^* \leq n$

Goal of online bipartite matching problem

Produce a matching M such that the resulting competitive ratio $\frac{|M|}{|M^*|}$ is **maximized**

For this talk, let's treat $n^* = n$



Here, the ratio is $3/4$

What is known?

- Why is online bipartite matching hard?
  - Maximum bipartite matching is poly time computable…
  - But we don't know the future in the online setting!

What is known?

- Why is online bipartite matching hard?
  - Maximum bipartite matching is poly time computable…
  - But we don't know the future in the online setting!

What is known?

- Why is online bipartite matching hard?
  - Maximum bipartite matching is poly time computable…
  - But we don't know the future in the online setting!
- Any reasonable greedy algorithm has competitive ratio $\geq 1/2$
  - Size of maximal matching is at least half of size of maximum matching

What is known?

$$\min_{G} \min_{\substack{V'\text{s arrival} \\ \text{sequence}}} \frac{\text{(Expected) number of matches}}{n^*}$$

	(Expected) Competitive ratio
Deterministic algorithm	$\frac{1}{2}$ ← Greedy
Deterministic hardness	$\frac{1}{2}$
Randomized algorithm	$1-\frac{1}{e}$ [KVV90] ← Ranking
Randomized hardness	$1-\frac{1}{e}+o(1)$ [KVV90]

- The Ranking algorithm [KVV90]
  - Pick a random permutation π over the offline vertices U
  - When vertex v_i arrive with $N(v_i)$, match v_i to the smallest indexed (with respect to π) unmatched neighbor

[KVV90] Richard M. Karp, Umesh V. Vazirani, Vijay V. Vazirani. *An optimal algorithm for on-line bipartite matching*. Symposium on Theory of Computing (STOC), 1990

What if there is additional side information?

- Learning-augmented algorithms
  - Designing algorithms using advice, predictions, etc.
    - α-consistent: α-competitive with no advice error
    - β-robust: β-competitive with any advice error

A natural goal is to design an algorithm with $\alpha = 1$ while β being the best possible classically

Example settings with side information

Offline vertices	Online vertices	Presence of edge	Advice	Error
Advertisers	Ad slots	New ad slot fits the advertisers' requirements	Historical data	Data may have noise, bias, etc.
Job opening	Hiring company	Applicant's suitability for the job role	LinkedIn qualifications	May lie about credentials
Food bento boxes	Conference attendee	Attendee's dietary options match the food type	Food preferences	May change mind if see a tastier option

Example settings with side information

Offline vertices	Online vertices	Presence of edge	Advice	Error
Advertisers	Ad slots	New ad slot fits the advertisers'	Historical data	Data may have noise, bias, etc.
Job				bout ials
Food bento boxes	Conference attendee	options match the food type	Food preferences	May change mind if see a tastier option

Takeaway: Advice can come in many forms. Nuances in problem dictate which kind of advice are practical and useful

Research question

- If we have “perfect information” about G^*, can we get n matches?
- Also, we know that Ranking achieves competitive ratio of $1 - \frac{1}{e}$

Can we get an algorithm that is both 1-consistent and $\left(1 - \frac{1}{e}\right)$-robust?

Prior related attempts

- [AGKK20] Prediction on edge weights adjacent to V under an optimal offline matching
  - Random vertex arrivals and weighted edges
  - Require hyper-parameter to quantify confidence in advice, so their consistency/robustness tradeoffs are not directly comparable
- [ACI22] Prediction of vertex degrees $\hat{d}(u_1), \ldots, \hat{d}(u_n)$ of the offline vertices in U
  - Adversarial arrival model
  - Optimal under the Chung-Lu-Vu random graph model [CLV03]
  - Unable to attain 1-consistency in general
- [JM22] Advice is a proposed matching for the first batch of arrived vertices
  - Two-staged arrival model [FNS21], where best possible robustness is ¾
  - For any $R \in [0, ¾]$, they can achieve consistency of $1 - \left(1 - \sqrt{1-R}\right)^2$
- [LYR23] Augment any "expert algorithm" with a pre-trained RL model
  - For any $\rho \in [0,1]$, their method is ρ-competitive to the given "expert algorithm"

[AGKK20] Antonios Antoniadis, Themis Gouleakis, Pieter Kleer, and Pavel Kolev. *Secretary and online matching problems with machine learned advice*. Neural Information Processing Systems (NeurIPS), 2020
[ACI22] Anders Aamand, Justin Chen, and Piotr Indyk. *(Optimal) Online Bipartite Matching with Degree Information*. Neural Information Processing Systems (NeurIPS), 2022
[CLV03] Fan Chung, Linyuan Lu, and Van Vu. *Spectra of random graphs with given expected degrees*. Proceedings of the National Academy of Sciences (PNAS), 2003
[JM22] Billy Jin and Will Ma. *Online bipartite matching with advice: Tight robustness-consistency tradeoffs for the two-stage model.* Neural Information Processing Systems (NeurIPS), 2022
[FNS21] Yiding Feng, Rad Niazadeh, and Amin Saberi. *Two-stage stochastic matching with application to ride hailing*. Symposium on Discrete Algorithms (SODA), 2021.
[LYR23] Pengfei Li, Jianyi Yang, and Shaolei Ren. Learning for edge-weighted online bipartite matching with robustness guarantees. International Conference on Machine Learning (ICML), 2023

Prior related attempts

- [AGKK20] Prediction on edge weights adjacent to V under an optimal offline matching
  - Random vertex arrivals and weighted edges
  - Require hyper-parameter to quantify confidence in advice, so their consistency/robustness trade
- [ACI22
  - Ad
  - Op
  - Un
- [JM22]
  - Tw
  - For
- [LYR23] Augment any "expert algorithm" with a pre-trained RL model
  - For any $\rho \in [0,1]$, their method is ρ-competitive to the given "expert algorithm"

Do <u>not</u> yield an algorithm that is both 1-consistent and $\left(1 - \frac{1}{e}\right)$-robust

[AGKK20] Antonios Antoniadis, Themis Gouleakis, Pieter Kleer, and Pavel Kolev. *Secretary and online matching problems with machine learned advice*. Neural Information Processing Systems (NeurIPS), 2020
[ACI22] Anders Aamand, Justin Chen, and Piotr Indyk. *(Optimal) Online Bipartite Matching with Degree Information*. Neural Information Processing Systems (NeurIPS), 2022
[CLV03] Fan Chung, Linyuan Lu, and Van Vu. *Spectra of random graphs with given expected degrees*. Proceedings of the National Academy of Sciences (PNAS), 2003
[JM22] Billy Jin and Will Ma. *Online bipartite matching with advice: Tight robustness-consistency tradeoffs for the two-stage model.* Neural Information Processing Systems (NeurIPS), 2022
[FNS21] Yiding Feng, Rad Niazadeh, and Amin Saberi. *Two-stage stochastic matching with application to ride hailing*. Symposium on Discrete Algorithms (SODA), 2021.
[LYR23] Pengfei Li, Jianyi Yang, and Shaolei Ren. Learning for edge-weighted online bipartite matching with robustness guarantees. International Conference on Machine Learning (ICML), 2023

Our first main result

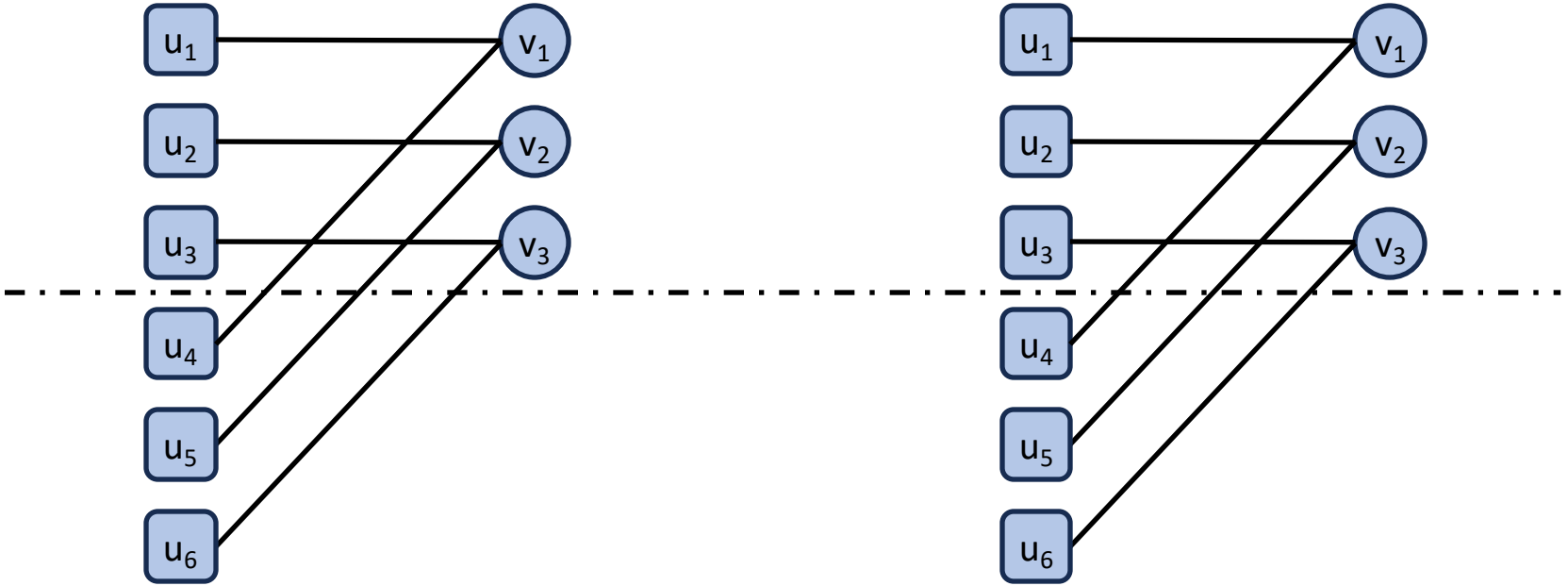
Impossibility result (Informal)

With adversarial vertex arrivals, no algorithm can be both 1-consistent and $> \frac{1}{2}$-robust, regardless of advice

- Extends to $(1 - a)$-consistent and $\left(\frac{1}{2} + a\right)$-robust, for any $a \in [0, ½]$
- Proof sketch (for $a = 0$ case):
  - Restrict G^* to be one of two possible graphs (next slide)
  - **Any** advice is equivalent to getting 1 bit of information
  - In first $\frac{n}{2}$ arrivals, no algorithm can distinguish between the two graphs
  - **Any** 1-consistent algorithm must behave as if the advice is perfect initially

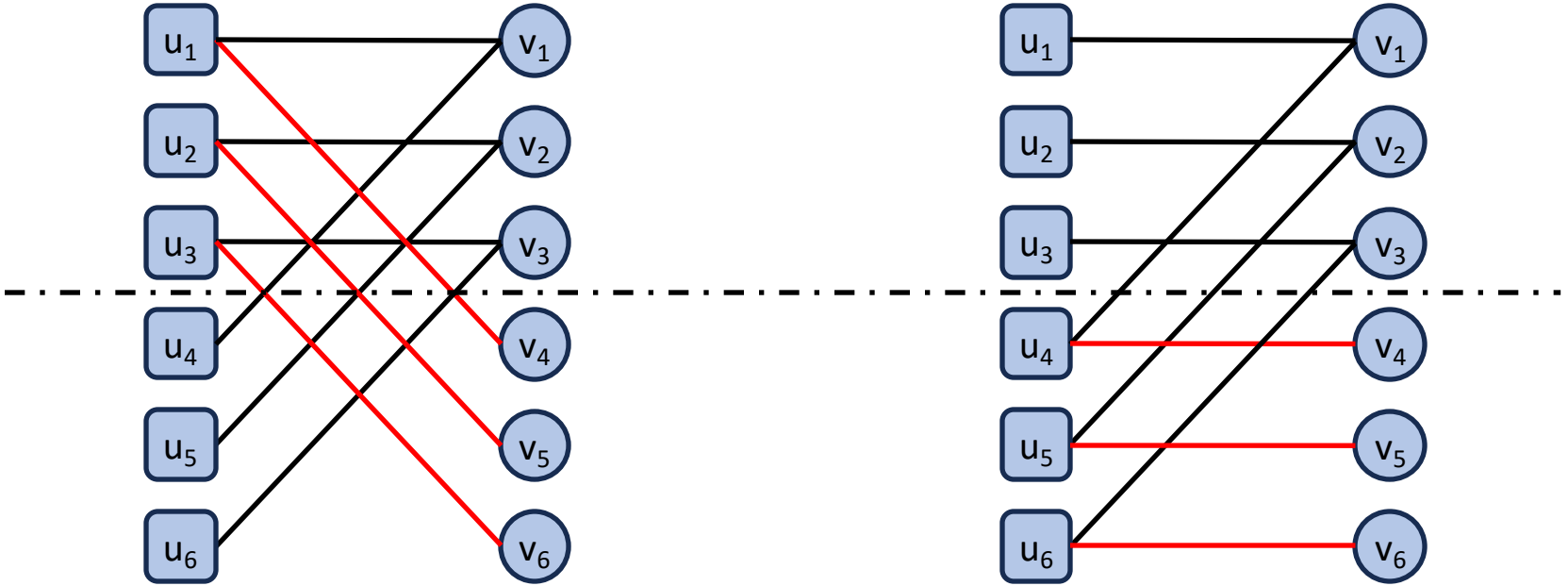
Impossibility result (Informal)

With adversarial vertex arrivals, no algorithm can be both 1-consistent and $> \frac{1}{2}$ -robust, regardless of advice



Impossibility result (Informal)

With adversarial vertex arrivals, no algorithm can be both 1-consistent and $> \frac{1}{2}$ -robust, regardless of advice



Hierarchy of arrival models [M13]

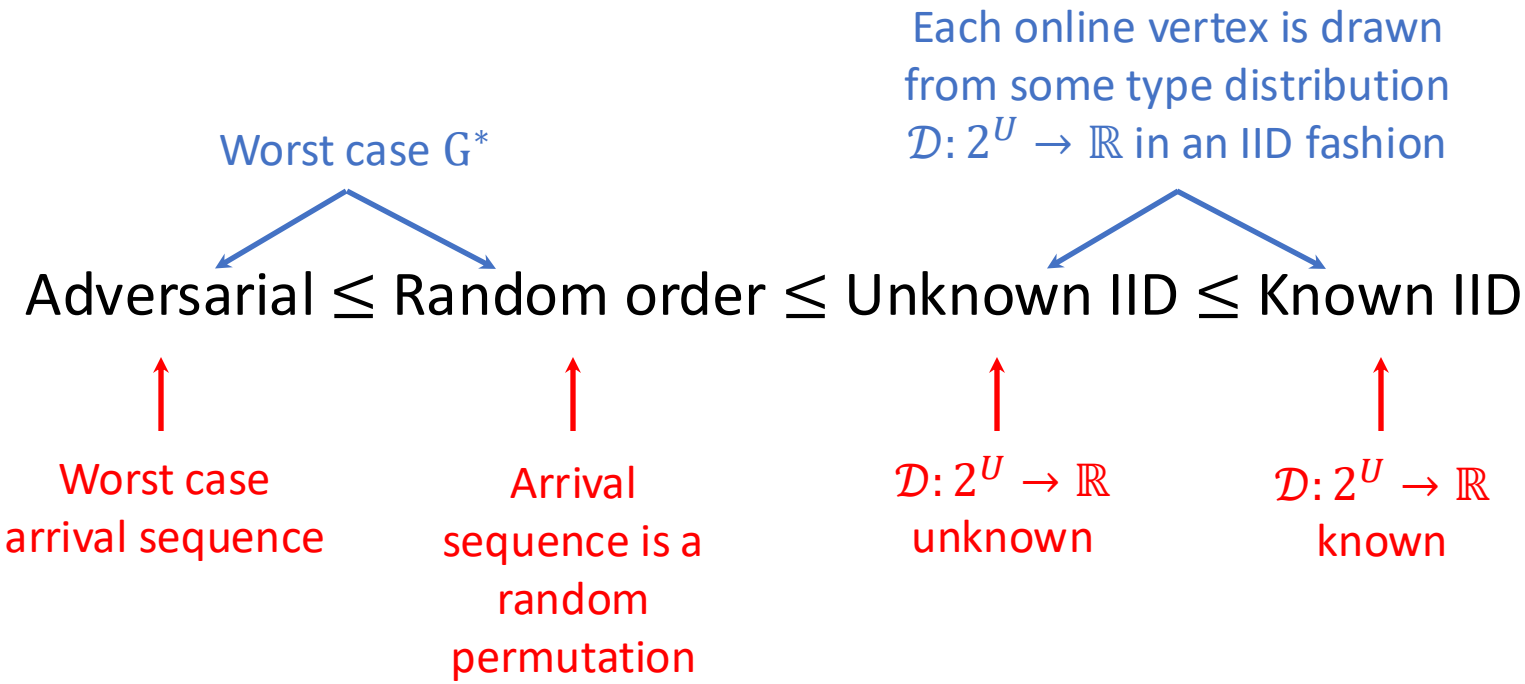
Harder  Easier

Adversarial $\leq$ Random order $\leq$ Unknown IID $\leq$ Known IID

Easier models can achieve
higher competitive ratios

[M13] Aranyak Mehta. *Online matching and ad allocation*. Foundations and Trends in Theoretical Computer Science, 2013

Hierarchy of arrival models [M13]



What is known?

Adversarial $\leq$ Random order $\leq$ Unknown IID $\leq$ Known IID

	(Expected) Competitive ratio	
	Adversarial arrival	**Random order arrival**
Deterministic algorithm	$\frac{1}{2}$ Greedy	$1-\frac{1}{e}$ [GM08]
Deterministic hardness	$\frac{1}{2}$	$\frac{3}{4}$
Randomized algorithm	$1-\frac{1}{e}$ [KVV90] Ranking	0.696 [MY11]
Randomized hardness	$1-\frac{1}{e}+o(1)$ [KVV90]	0.823 [MGS12]

[GM08] Gagan Goel and Aranyak Mehta. *Online budgeted matching in random input models with applications to Adwords*. Symposium on Discrete Algorithms (SODA), 2008
[MY11] Mohammad Mahdian and Qiqi Yan. *Online Bipartite Matching with Random Arrivals: An Approach Based on Strongly Factor-Revealing LPs*. Symposium on Theory of Computing (STOC), 2011
[MGS12] Vahideh H Manshadi, Shayan Oveis Gharan, and Amin Saberi. *Online stochastic matching: Online actions based on offline statistics*. Mathematics of Operations Research, 2012

Research question

Can we get an algorithm that is both 1-consistent and $\left(1 - \frac{1}{e}\right)$-robust?

β

- Let β denote the "best possible competitive ratio"
- Our first result says: This is not possible for adversarial arrivals!
- What about random order arrivals?

Adversarial $\leq$ Random order $\leq$ Unknown IID $\leq$ Known IID

Our second main result

Can we get an algorithm that is both 1-consistent and $\left(1 - \frac{1}{e}\right)$-robust?

β

Goal achievable in random order (Informal)

With random order, there is an algorithm achieves competitive ratio interpolating between 1 and $\beta \cdot \left(1 - o(1)\right)$, depending on advice quality

- Our method is a meta-algorithm that uses any Baseline that achieves β
- So, we are simultaneously 1-consistent and $\beta \cdot \left(1 - o(1)\right)$-robust
- For random arrival model, we know that $0.696 \leq \beta \leq 0.823$

e.g. use Ranking

Realized type counts as advice

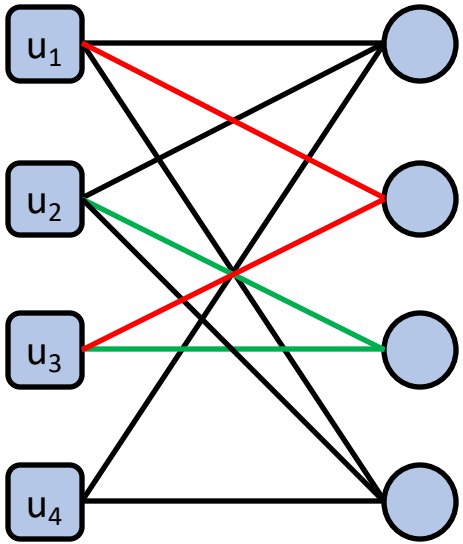
- Classify online vertex in $G^* = (U \cup V, E)$ based on their types
  - Type of v_i is the set of offline vertices in $N(v_i)$ are adjacent to [BKP20]
- Define integer vector $c^* \in \mathbb{N}^{2^n}$ indexed by all possible types 2^U
  - $c^*(t)$ = Number of times the type $t \in 2^U$ occurs in G^*
- Define $T^* \subseteq 2^U$ as the subset of non-zero counts in c^*
  - Note: $|T^*| \leq n \ll 2^{|U|} = 2^n$

[BKP20] Allan Borodin, Christodoulos Karavasilis, and Denis Pankratov. *An experimental study of algorithms for online bipartite matching*. Journal of Experimental Algorithmics (JEA), 2020

Realized type counts as advice

- Classify online vertex in $G^* = (U \cup V, E)$ based on their types
  - Type of v_i is the set of offline vertices in $N(v_i)$ are adjacent to [BKP20]
- Define integer vector $c^* \in \mathbb{N}^{2^n}$ indexed by all possible types 2^U
  - $c^*(t)$ = Number of times the type $t \in 2^U$ occurs in G^*
- Define $T^* \subseteq 2^U$ as the subset of non-zero counts in c^*
  - Note: $|T^*| \leq n \ll 2^{|U|} = 2^n$
- Advice is simply an estimate vector $\hat{c}$ which approximates c^*
  - Let $\hat{T}$ be non-zero counts in $\hat{c}$. Similarly, we have $|\hat{T}| \leq n$
  - Can represent $\hat{c}$ using $O(n)$ labels and numbers

Realized type counts as advice



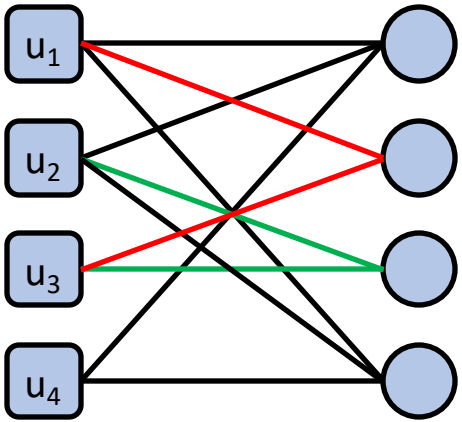
T^*

Type	c^*
$\{u_1, u_2, u_4\}$	2
$\{u_1, u_3\}$	1
$\{u_2, u_3\}$	1
$2^U \setminus T^*$	0

Here, $|T^*| = 3 \ll 2^4 = 16$

The Mimic algorithm

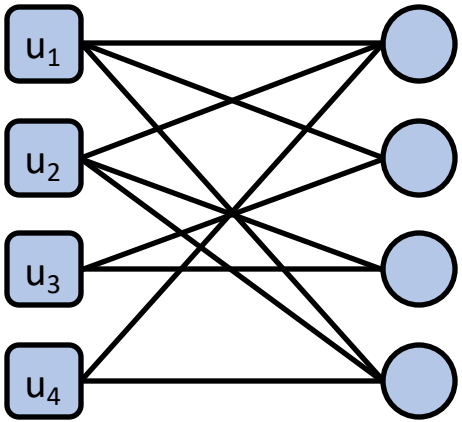
- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched



Type	c^*
$\{u_1, u_2, u_4\}$	2
$\{u_1, u_3\}$	1
$\{u_2, u_3\}$	1

The Mimic algorithm

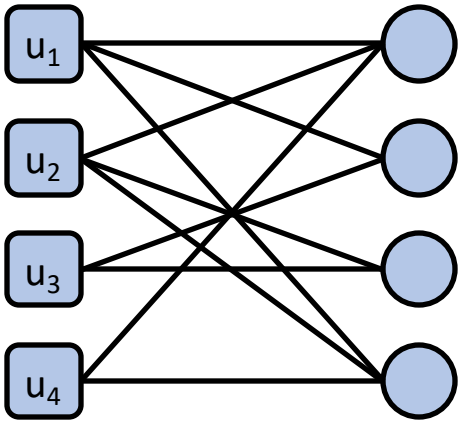
- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched



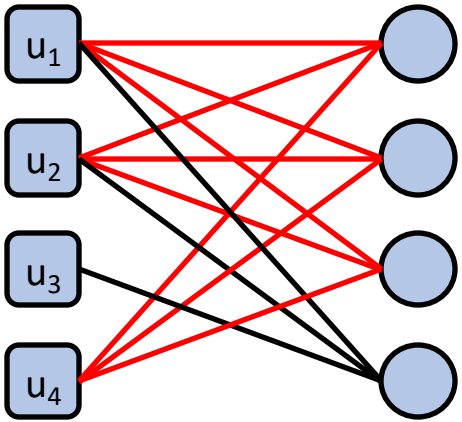
Type	c^*	$\hat{c}$
$\{u_1, u_2, u_4\}$	2	3
$\{u_1, u_3\}$	1	0
$\{u_2, u_3\}$	1	0
$\{u_1, u_2, u_3\}$	0	1

The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched

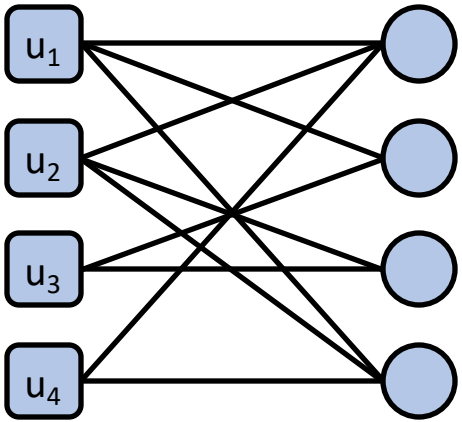


Type	c^*	$\hat{c}$
$\{u_1, u_2, u_4\}$	2	3
$\{u_1, u_3\}$	1	0
$\{u_2, u_3\}$	1	0
$\{u_1, u_2, u_3\}$	0	1



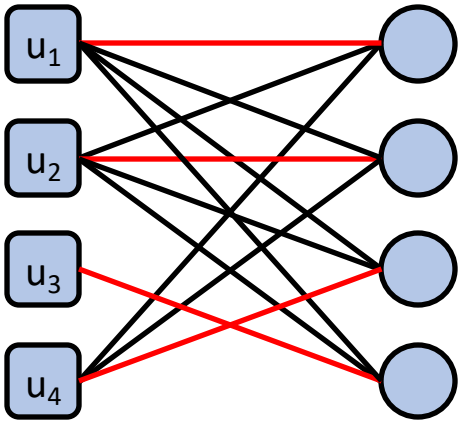
The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched



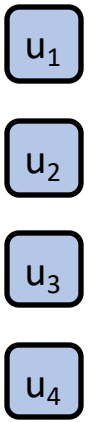
Type	c^*	$\hat{c}$
$\{u_1, u_2, u_4\}$	2	3
$\{u_1, u_3\}$	1	0
$\{u_2, u_3\}$	1	0
$\{u_1, u_2, u_3\}$	0	1

$\widehat{M}$

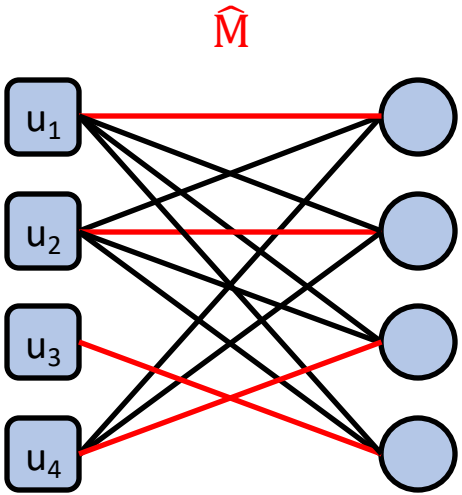


The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched

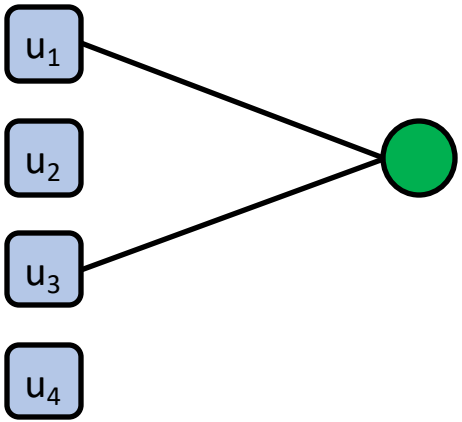


Type	c^*	$\hat{c}$
$\{u_1, u_2, u_4\}$	2	3
$\{u_1, u_3\}$	1	0
$\{u_2, u_3\}$	1	0
$\{u_1, u_2, u_3\}$	0	1

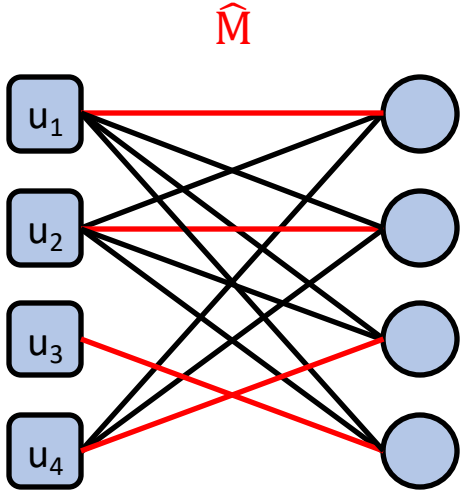


The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched

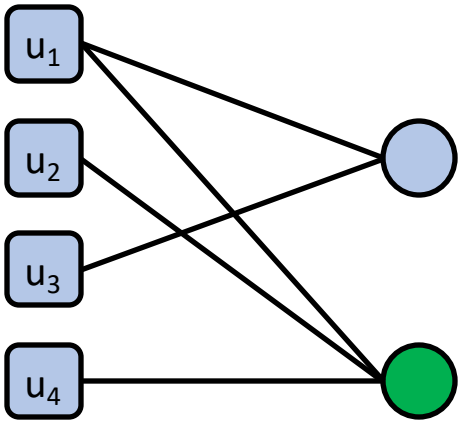


Type	c^*	$\hat{c}$
$\{u_1, u_2, u_4\}$	2	3
$\{u_1, u_3\}$	1	0
$\{u_2, u_3\}$	1	0
$\{u_1, u_2, u_3\}$	0	1



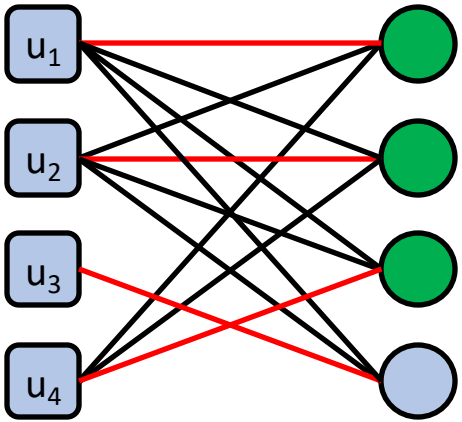
The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched



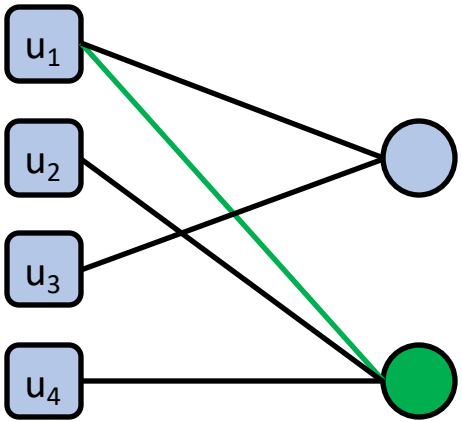
Type	c^*	$\hat{c}$
$\{u_1, u_2, u_4\}$	2	3
$\{u_1, u_3\}$	1	0
$\{u_2, u_3\}$	1	0
$\{u_1, u_2, u_3\}$	0	1

$\widehat{M}$



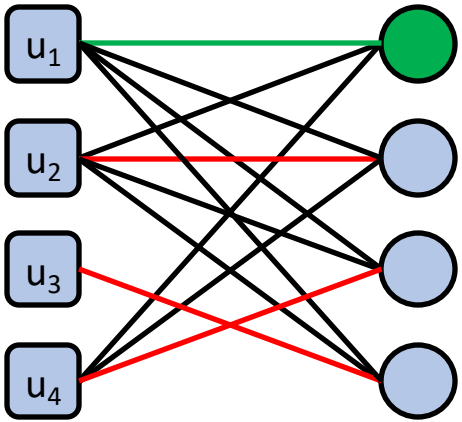
The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched



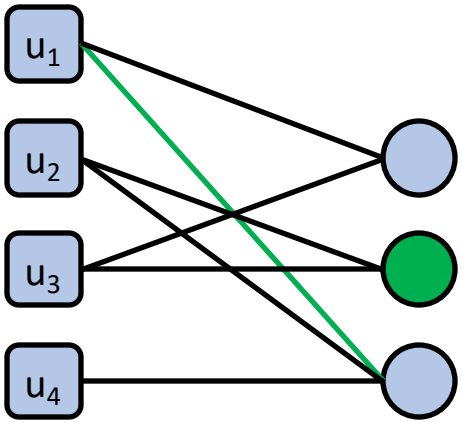
Type	c^*	$\hat{c}$
$\{u_1, u_2, u_4\}$	2	~~3~~ 2
$\{u_1, u_3\}$	1	0
$\{u_2, u_3\}$	1	0
$\{u_1, u_2, u_3\}$	0	1

$\widehat{M}$



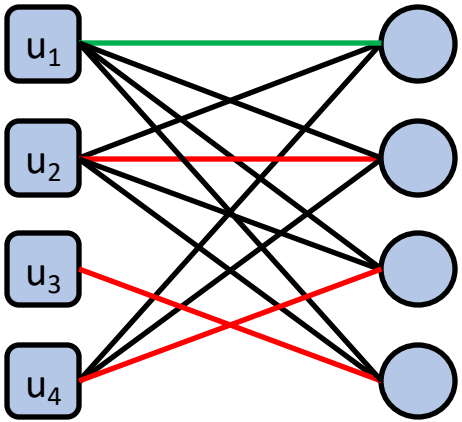
The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched



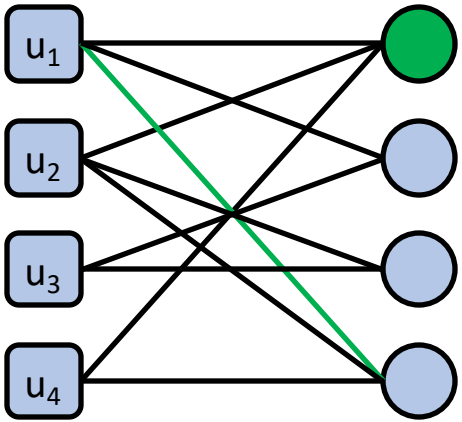
Type	c^*	$\hat{c}$
$\{u_1, u_2, u_4\}$	2	~~3~~ 2
$\{u_1, u_3\}$	1	0
$\{u_2, u_3\}$	1	0
$\{u_1, u_2, u_3\}$	0	1

$\widehat{M}$



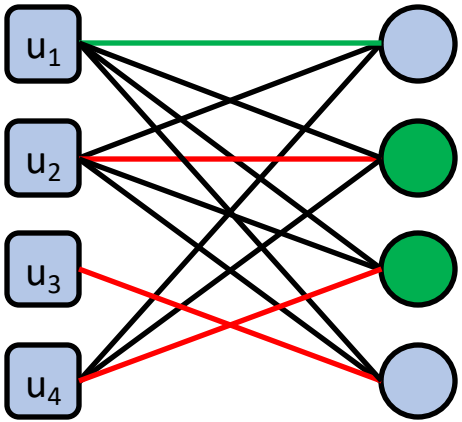
The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched



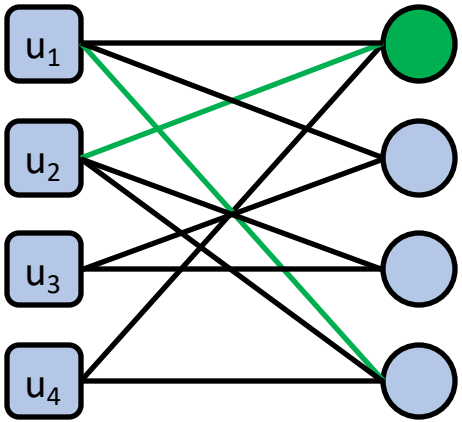
Type	c^*	$\hat{c}$
$\{u_1, u_2, u_4\}$	2	~~3~~ 2
$\{u_1, u_3\}$	1	0
$\{u_2, u_3\}$	1	0
$\{u_1, u_2, u_3\}$	0	1

$\widehat{M}$



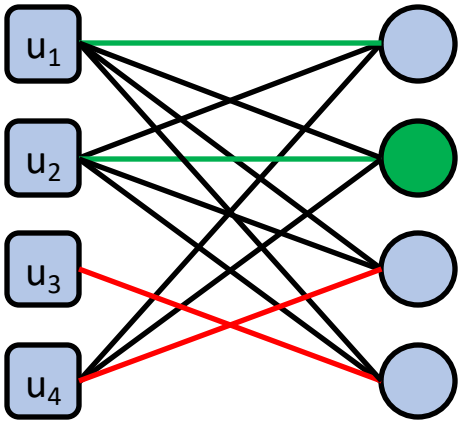
The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched



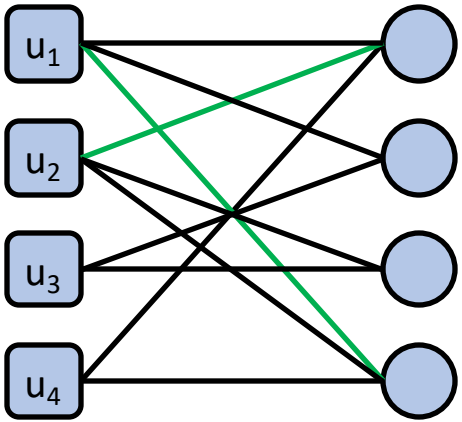
Type	c^*	$\hat{c}$
$\{u_1, u_2, u_4\}$	2	~~3~~ ~~2~~ 1
$\{u_1, u_3\}$	1	0
$\{u_2, u_3\}$	1	0
$\{u_1, u_2, u_3\}$	0	1

$\widehat{M}$



The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched

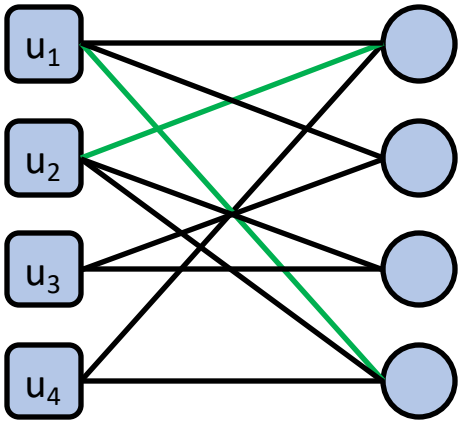


Type	c^*	$\hat{c}$
$\{u_1, u_2, u_4\}$	2	3
$\{u_1, u_3\}$	1	0
$\{u_2, u_3\}$	1	0
$\{u_1, u_2, u_3\}$	0	1

Produced matching size = 2

The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched



Type	c^*	$\hat{c}$
$\{u_1, u_2, u_4\}$	2	3
$\{u_1, u_3\}$	1	0
$\{u_2, u_3\}$	1	0
$\{u_1, u_2, u_3\}$	0	1

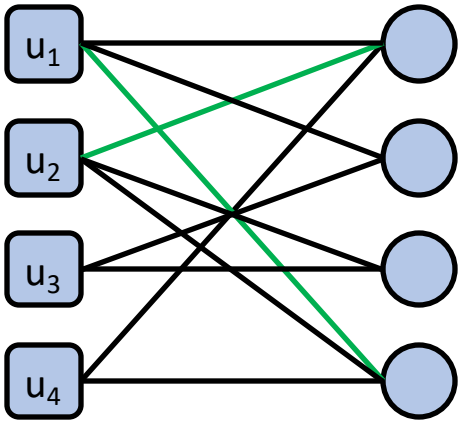
Produced matching size

= 2

$$L_1(c^*, \hat{c}) = |3 - 2| + |0 - 1| + |0 - 1| + |1 - 0| + 0 \ldots = 4$$

The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched



Type	c^*	$\hat{c}$
$\{u_1, u_2, u_4\}$	2	3
$\{u_1, u_3\}$	1	0
$\{u_2, u_3\}$	1	0
$\{u_1, u_2, u_3\}$	0	1

Produced matching size

$$= 2 = \left|\widehat{M}\right| - \frac{L_1(c^*, \hat{c})}{2}$$

Error is "double counted" in L_1

$$L_1(c^*, \hat{c}) = |3-2| + |0-1| + |0-1| + |1-0| + 0 \ldots = 4$$

The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched
- Analysis
  - $0 \leq L_1(c^*, \hat{c}) \leq 2n$ measures how close $\hat{c}$ is to c^*
  - By blindly following advice, Mimic gets a matching of size $\left|\widehat{M}\right| - \frac{L_1(c^*,\hat{c})}{2}$
  - Mimic beats an advice-free Baseline whenever $\left|\widehat{M}\right| - \frac{L_1(c^*,\hat{c})}{2} > \beta \cdot n$

The Mimic algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Try to mimic edge matches in $\widehat{M}$ while tracking the types of each arrival
  - If unable to mimic, leave arrival unmatched
- Analysis
  - $0 \leq L_1(c^*, \hat{c}) \leq 2n$ measures how close $\hat{c}$ is to c^*
  - By blindly following advice, Mimic gets a matching of size $|\widehat{M}| - \frac{L_1(c^*,\hat{c})}{2}$
  - Mimic beats an advice-free Baseline whenever $|\widehat{M}| - \frac{L_1(c^*,\hat{c})}{2} > \beta \cdot n$
  - Mimic beats an advice-free Baseline whenever $\frac{L_1(c^*,\hat{c})}{n} < 2(1-\beta)$

For this talk, let's treat $|\widehat{M}| = n$

How to test advice quality?

Insight: Use sublinear property testing to estimate $L_1(c^*, \hat{c})$!

- Define $p = \frac{c^*}{n}$ and $q = \frac{\hat{c}}{n}$ as distributions over the 2^U types

How to test advice quality?

Insight: Use sublinear property testing to estimate $L_1(c^*, \hat{c})$!

- Define $p = \frac{c^*}{n}$ and $q = \frac{\hat{c}}{n}$ as distributions over the 2^U types
- [VV11, JHW18]: Can estimate $L_1(p, q)$ "well" using $o(n)$ IID samples
  - Some adjustments needed to apply this property testing idea to our online bipartite matching setup, but it can be done (talk to me to find out more)

[VV11] Gregory Valiant and Paul Valiant. *The power of linear estimators*. Foundations of Computer Science (FOCS), 2011.
[JHW18] Jiantao Jiao, Yanjun Han, and Tsachy Weissman. *Minimax estimation of the L_1 distance*. IEEE Transactions on Information Theory, 2018.

The TestAndMatch algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{\mathrm{M}}$ on the graph defined by advice $\hat{\mathrm{c}}$
  - Run Mimic while testing quality of $\hat{\mathrm{c}}$ by estimating $\mathrm{L}_1(\mathrm{c}^*, \hat{\mathrm{c}})$
  - If test declares $\mathrm{L}_1(\mathrm{c}^*, \hat{\mathrm{c}})$ is “large”, use Baseline for remaining arrivals
  - Otherwise, continue using Mimic for remaining arrivals

The TestAndMatch algorithm

- Algorithm
  - Fix any arbitrary maximum matching $\widehat{M}$ on the graph defined by advice $\hat{c}$
  - Run Mimic while testing quality of $\hat{c}$ by estimating $L_1(c^*, \hat{c})$
  - If test declares $L_1(c^*, \hat{c})$ is "large", use Baseline for remaining arrivals
  - Otherwise, continue using Mimic for remaining arrivals
- Analysis
  - If $\hat{L}_1 \lesssim 2(1-\beta)$, then TestAndMatch attains ratio of at least $1 - \frac{L_1(c^*, \hat{c})}{2n}$
  - Otherwise, TestAndMatch attains ratio of at least $\beta \cdot (1 - o(1))$

Our second main result

Can we get an algorithm that is both 1-consistent and $\left(1 - \frac{1}{e}\right)$-robust?

β

Goal achievable in random order (Informal)

With random order, there is an algorithm achieves competitive ratio interpolating between 1 and $\beta \cdot \left(1 - o(1)\right)$, depending on advice quality

- Our method is a meta-algorithm that uses any Baseline that achieves β
- So, we are simultaneously 1-consistent and $\beta \cdot \left(1 - o(1)\right)$-robust
- For random arrival model, we know that $0.696 \leq \beta \leq 0.823$

Our second main result

Can we get an algorithm that is both 1-consistent and $\left(1-\frac{1}{e}\right)$-robust? β

Goal achievable in random order (Informal)

Let $\hat{L}_1$ be estimate of $L_1(c^*, \hat{c})$ from $o(n)$ vertex arrivals. TestAndMatch achieves a competitive ratio of at least

- At least $1 - \frac{L_1(c^*,\hat{c})}{2n} \geq \beta$, when $\hat{L}_1$ "small"
- At least $\beta \cdot \left(1 - o(1)\right)$, when $\hat{L}_1$ "large"

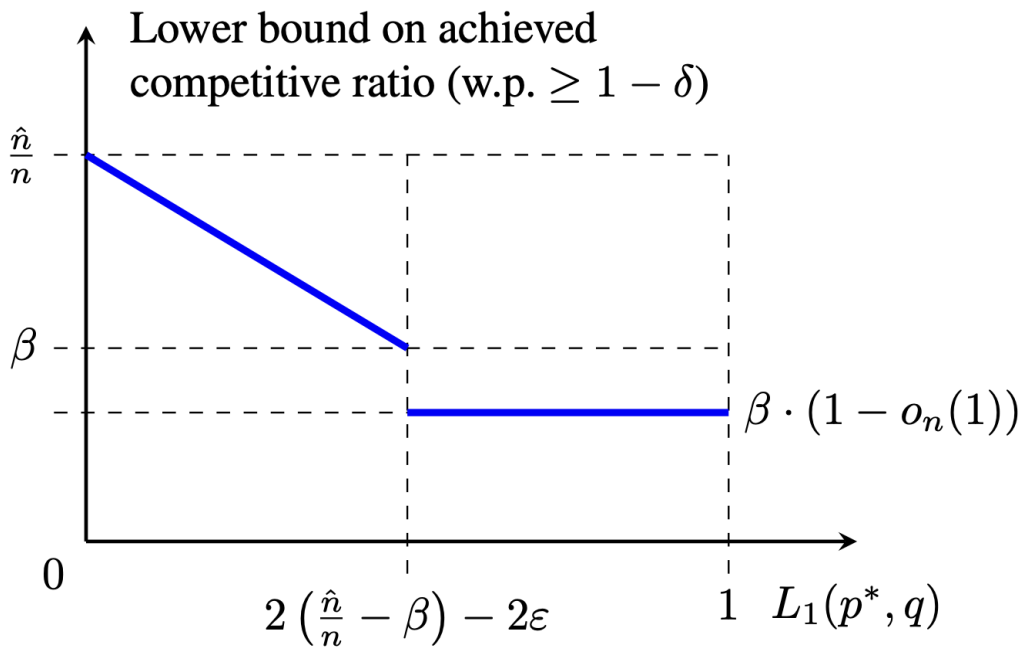
i.e., TestAndMatch is 1-consistent and $\beta \cdot \left(1 - o(1)\right)$-robust

Our second main result

Can we ge… β …obust?

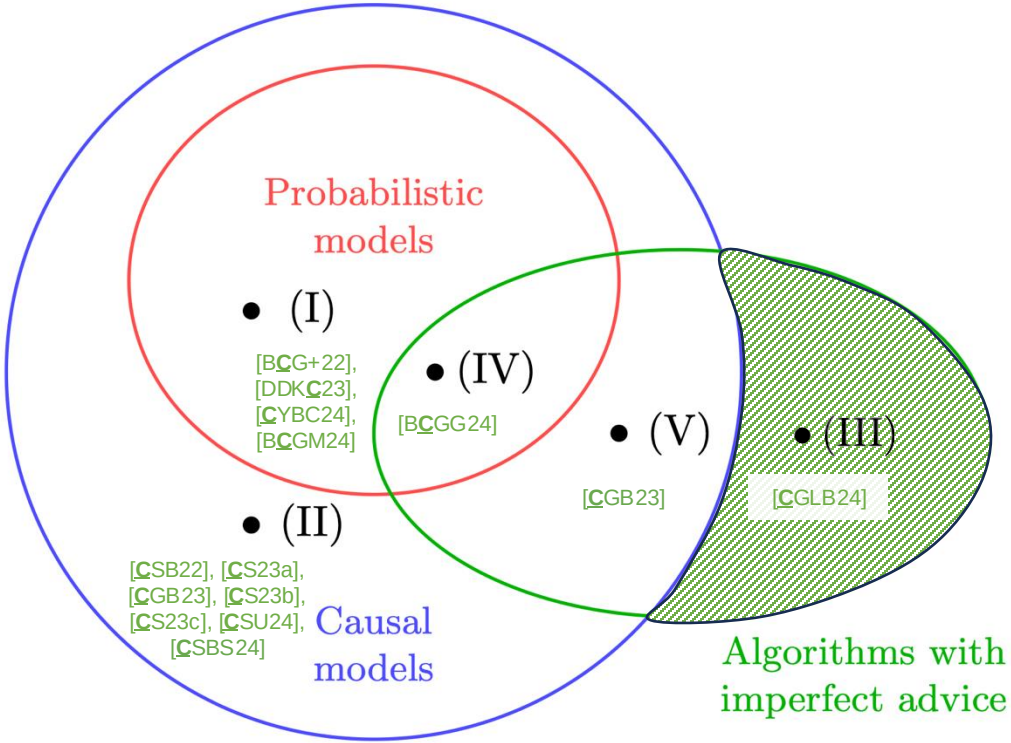
Let $\hat{L}_1$ be est… …ndMatch

- At lea…
- At lea…



i.e., TestAndMatch is 1-consistent and $\beta \cdot \big(1 - o(1)\big)$-robust

Recap: Main themes explored in my PhD



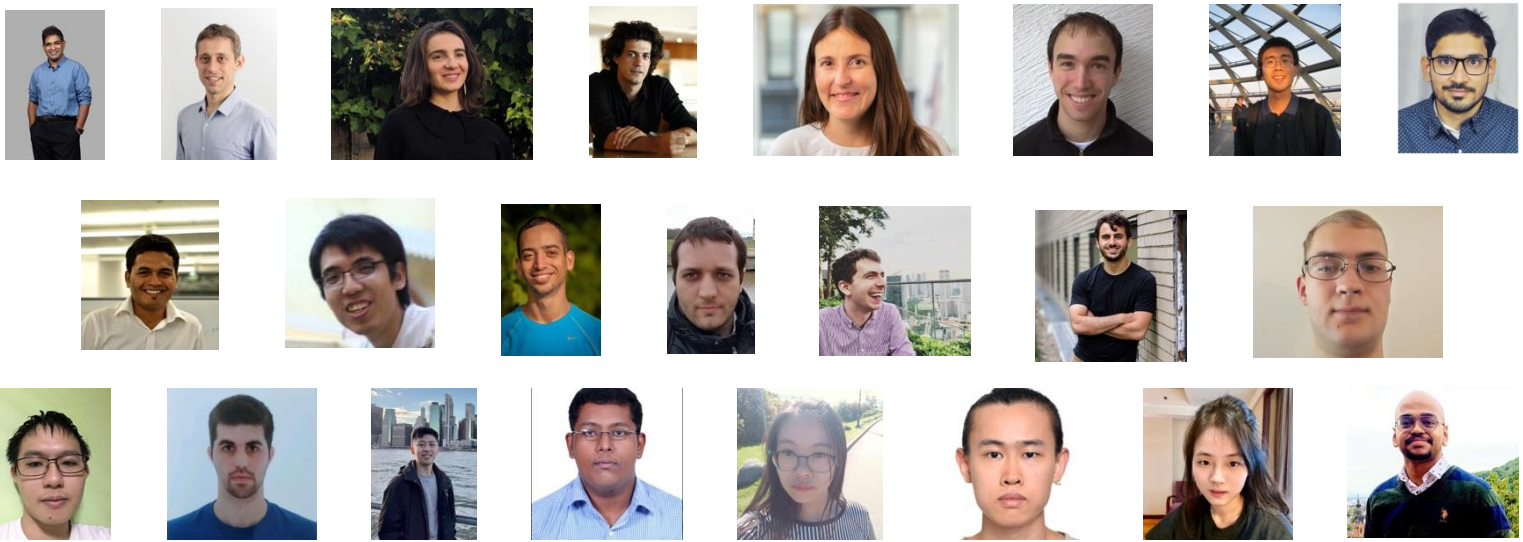
Future research directions

1. Developing causality-informed AI/ML methods
   - Most AI/ML methods are built on statistical/associational relations
   - Naively making real-world decisions can lead to unexpected outcomes

Future research directions

1. Developing causality-informed AI/ML methods
   - Most AI/ML methods are built on statistical/associational relations
   - Naively making real-world decisions can lead to unexpected outcomes
2. Incorporating human knowledge as imperfect advice
   - First came out of the study of online algorithms: main difficulty is not knowing the future and advice is partial future knowledge
   - I believe framework is much more broadly applicable
     - How can we principally elicit and incorporate human domain experts' knowledge?
     - Note: "Just Bayesian it" doesn't always work as an expert can be confidently wrong

Thank you to all my amazing collaborators during my PhD journey!



Thank you for your kind attention!

(Some of the works with them are not mentioned in this presentation or are not part of my thesis)