Stochastic Matching via Local Sparsification

Sara Ahmadian
Google Research
sahmadian@google.com

Edith Cohen
Google Research and Tel Aviv University
edith@cohenwang.com

Mohammad Roghani
Google Research
roghani@google.com

Abstract

The classic online stochastic matching problem typically requires immediate and irrevocable matching decisions. However, in many modern decentralized systems such as real-time ride-hailing and distributed cloud computing, the primary bottleneck is often local communication bandwidth rather than the timing of the match itself. We formalize this challenge by introducing a two-stage local sparsification framework. In this setting, arriving requests must prune their realized compatibility sets to a strict budget of k edges before a central coordinator optimizes the global matching. This creates a "middle ground" between local information constraints and global optimization utility.

We propose a local selection strategy, parametrized by a fractional solution of the expected instance. Theoretically, we quantify the approximation ratio as a function of the solution's *spread*. For every budget k, the sparsifier retains at least a $1 - 1/e$ fraction of the fractional benchmark and under sufficient spread it retains a $1 - \varepsilon$ fraction with $k = \varepsilon^{-2}$. Empirically, we demonstrate the robustness of our approach using the New York City ride-hailing datasets and adversarial synthetic benchmarks. Our results show that near-optimal global matching is achievable even with highly constrained local budgets, significantly outperforming standard online baselines.

1 Introduction

Online bipartite matching is a fundamental problem in resource allocation and algorithm design. In its standard mathematical framework, the problem is modeled on a bipartite graph $G = (U \cup V, E)$, where a set of offline resources V (e.g., servers, ad inventory, or ride-hailing drivers) is known in advance. Online requests U arrive sequentially over time, revealing their compatible neighborhood $\Gamma(u) \subseteq V$. In the traditional online paradigm, the algorithm must make an immediate and irrevocable decision to either match u to an available resource or drop the request forever.

The seminal work of Karp, Vazirani, and Vazirani [30] established that randomized ranking achieves a tight $1 - 1/e$ competitive ratio under strictly adversarial arrivals. To bypass this barrier, a rich line of literature has explored stochastic matching models. In the *known distribution* model, request types are drawn independently from a known prior distribution [16, 32, 24, 5]. By leveraging this statistical knowledge, algorithms can successfully break the $1 - 1/e$ barrier. Subsequent research has generalized these arrival processes, refined approximation bounds, and relaxed the required knowledge of the distributions [5, 29, 17].

A defining characteristic of this classic literature is the strict requirement of immediate, irrevocable decision-making. However, in many modern decentralized systems, the primary bottleneck is not the irrevocability of a match, but rather *local communication bandwidth and memory limits*. In distributed

cloud computing, querying the real-time availability of every compatible server incurs prohibitive control-plane latency. Similarly, in ride-hailing or real-time ad exchanges, platform interfaces and latency constraints restrict the system to querying or presenting only a small budget of options per user. In these settings, an algorithm must act locally to narrow down the potential edges before a central coordinator can optimize the global matching.

In this paper, we depart from the strict online matching paradigm to study stochastic matching through the lens of *local sparsification*. We formalize this as a two-stage stochastic process:

1. **Local Pruning:** Each arriving request $u_i \in U$ observes its compatible resources and must immediately apply a local selection rule to choose a subset S_i of at most k edges, relying only on the prior distribution without coordination with other arrivals.
2. **Global Matching:** Once all n selections are finalized, a central coordinator receives the resulting sparse subgraph $G_S = (U \cup V, \bigcup S_i)$ and computes a maximum bipartite matching $M(G_S)$.

The objective is to design a selection rule that maximizes the *preservation ratio* $\alpha = \mathbb{E}[|M(G_S)|]/\mathbb{E}[|M(G)|]$. This framework isolates a fundamental challenge: how to design capacity-constrained local strategies that globally preserve the maximum matching size under uncertainty.

1.1 Technical Overview

Our approach parameterizes local edge-selection using offline statistical knowledge. We relax the stochastic problem via the *Expected Instance Linear Program* (LP), whose fractional solutions route expected demand to resources. In the online phase, each arriving request uses *Variance-Optimal* (VarOpt) dependent sampling to select at most k edges, with the LP's fractional values as sampling weights.

To bound the performance of this sparsifier, we introduce a *Heavy-Light* decomposition of the fractional solution: an edge is "heavy" if its fractional value exceeds $1/k$ and "light" otherwise. Two structural facts drive the analysis. First, sparsification never discards heavy mass: VarOpt retains heavy edges with probability one. Second, light mass is forwarded in pieces of size at most $1/k$, so the overflow of light mass at a resource vanishes as $O(1/\sqrt{k})$. Our main result (Theorem 3.4) lower-bounds the expected matching size on the sparsifier by $(1 - U_k(\eta))$ times the fractional objective, where η is the heavy fraction and $U_k \leq 1/e$ is an explicit concave function. Consequently, for every budget the sparsifier retains a $1 - 1/e$ fraction of the fractional benchmark, which is tight for $k = 1$, and if the fractional matching is "well-spread", a local budget of $k = \epsilon^{-2}$ guarantees an expected matching size within a $1 - \epsilon$ factor of the benchmark (Corollary 3.5). Our algorithmic objective is then to construct high-value, well-spread fractional solutions to guide local choices.

This well-spread assumption naturally captures the reality of dense platforms such as ride-hailing, where riders are surrounded by many nearly interchangeable vehicles rather than relying on a single critical connection (see Claim 3.7 for the formal definition of interchangeable resources). Guided by the theory, we also propose a simple Monte Carlo heuristic for constructing high-spread fractional solutions: we repeatedly sample realizations of the stochastic instance, compute offline optimal matchings under random shuffling, and average the resulting edge incidences to obtain fractional selection weights. These weights are feasible for the LP under the empirical arrival frequencies (Lemma C.1) and are used by the local VarOpt sparsifier in our experiments.

On NYC Taxi data, our local sparsifier closely tracks the performance of a centralized, full-information offline maximum matching. Our framework remains robust when the spread assumption is violated: in several adversarial synthetic benchmarks, heavy edges carry a large share of the fractional solution, yet the algorithm continues to perform well, and by returning $k > 1$ edges per request it bypasses the 0.901 efficiency barrier for online stochastic matching algorithms (Section 4).

1.2 Related work

Online Random-Order Arrival. To bypass the $1 - 1/e$ adversarial barrier, a line of literature has studied random order arrivals. Under this condition, the RANKING algorithm is at least 0.696-competitive [31] and at most 0.727-competitive [29] for unweighted bipartite graphs. For vertex-weighted graphs, the ratio of RANKING has been improved to 0.686 [22, 25, 33], and for

edge-weighted oblivious (query-commit) bipartite matching, a Ranking-style algorithm achieves 0.659 [23]. For general graphs, a series of works has steadily advanced our understanding, recently improving the approximation ratio for the RANKING algorithm under random vertex arrivals to 0.560 [2, 8, 15, 34, 35, 14, 13]. Weighted variants of randomized greedy matching in general graphs remain particularly challenging; algorithms have recently been shown to achieve ratios of 0.5015 for the vertex-weighted case [9] and 0.5014 for the edge-weighted variant [35].

Streaming and Communication Complexity. A related line of research examines maximum matching in the streaming setting under vertex arrivals [28, 19, 26]. While streaming algorithms can store $O(n \operatorname{polylog} n)$ edges, no single-pass algorithm of this space under adversarial arrivals can exceed the $1 - 1/e$ barrier [26]. For edge-arrival streams, research typically relies on constructing sparsifiers to preserve matching size [27, 7, 3]. Our work differs by requiring that the sparsification happens *locally* and *independently* at the time of vertex arrival, rather than having access to the full stream history or global coordination.

The Power of Spread and Regularity. The utility of spread solutions has been implicitly recognized across several matching paradigms. In online stochastic matching, algorithms guided by two suggested matchings route each arrival using two precomputed choices [16, 5, 32, 20], and on d-regular graphs a simple randomized online algorithm achieves $1 - e^{-d}d^d/d! = 1 - O(1/\sqrt{d})$ [5], the same expression as our bound for fully light solutions (Corollary 3.5(c)). Similar structural guarantees are used to achieve sublinear time in offline matching for d-regular graphs [18, 12]. The importance of this structure is underscored by recent hardness results: without such guarantees, even estimating the size of a $(1 - \epsilon)$-approximate maximum matching requires near-quadratic time [6, 4]. Our framework formalizes the intuition that when weights are sufficiently dispersed, localized sampling provides a powerful mechanism to recover near-optimal global matchings without the need for exhaustive exploration.

2 Problem Definition and the Global Baseline

2.1 The Stochastic Matching environment

We operate under the *known distribution* model, a standard benchmark in stochastic optimization [16, 32]. An instance is defined by $(V, \mathcal{D}, n)$, where:

- **Supply (V):** A fixed set of resources $V = \{v_i\}$.
- **Demand Distribution ($\mathcal{D}$):** A known probability distribution over 2^V possible subsets of V representing different demand types.
- **Number of requests (n):** The total number of requests that will arrive.

The demand is realized as a set $U = \{u_1, \ldots, u_n\}$ where each request u_i is independently associated with a random subset of compatible resources $R_i \subseteq V$ sampled *i.i.d.* from the distribution $\mathcal{D}$. This process induces a *realized bipartite graph* $G = (U \cup V, E)$, where an edge (u_i, v) exists if and only if $v \in R_i$.

2.2 Local Sparsification

In many real-world systems, the full graph G is too large to process centrally due to latency or bandwidth limits. We introduce a *local budget constraint* k, which dictates that each request u_i can only "report" a subset of its compatible edges to the central coordinator.

Definition 2.1 (The Selection Rule Φ_k)**.** A sparsifier is defined by a local randomized selection function $\Phi_k : 2^V \to \Delta(2^V)$, where $\Delta(2^V)$ denotes the space of probability distributions over the subsets of V. For each arriving request u_i with compatibility set R_i, the function draws a random subset $S_i \sim \Phi_k(R_i)$ such that $S_i \subseteq R_i$ and $|S_i| \leq k$.

Crucially, Φ_k must be **local**: the selection for request u_i is made using only the distribution $\mathcal{D}$, the realization R_i, and private randomness, without any knowledge of other requests u_j ($j \neq i$).

2.3 Maximizing the Preservation Ratio

Let $G_S = (U \cup V, E_S)$ be the sparsified graph with edge set $E_S = \bigcup_{i=1}^{n} \{(u_i, v) : v \in S_i\}$, and let $M(G)$ denote a maximum matching of a graph G. We measure the performance of a selection strategy by the *preservation ratio*

$$\alpha = \frac{\mathbb{E}[|M(G_S)|]}{\mathbb{E}[|M(G)|]}, \tag{1}$$

and our goal is to design a rule Φ_k that maximizes α. Although α is a ratio of expectations, it also describes individual realizations when matchings are large ($|M(G)| \gg \sqrt{n}$), since $|M(G)|$ and $|M(G_S)|$ concentrate around their means (see Corollary 3.6).

The budget k forces a form of strategic pruning. If $k = 1$, each request must commit to a single resource without knowledge of the other arrivals. On the complete bipartite graph $K_{n,n}$, no Φ_1 rule achieves $\alpha > 1 - (1 - 1/n)^n \approx 1 - 1/e$ (Corollary 3.5). As k increases, we move into a "middle ground" where we aim to maintain the global utility of the graph with minimal local information.

2.4 The Expected Instance LP

Our framework relies on the *Expected Instance Linear Program* (LP), a well-established formulation that captures the average-case behavior of the stochastic instance by fractionally routing expected demand. This LP serves as a global roadmap, enforcing matching constraints while relaxing the discrete arrival process.

For an instance of $(V, \mathcal{D}, n)$,

- Let $T = \{t_1, \ldots, t_m\}$ be the set of all possible request types.
- Let p_j be the single-draw probability of type t_j, where $\sum_{j=1}^{m} p_j = 1$. The expected number of arrivals for type t_j is np_j.
- Let $\Gamma(t_j) \subseteq V$ be the subset of resources compatible with type t_j.

Definition 2.2 (Expected Instance LP)**.** Let x_{ij} be a fractional variable representing the conditional probability that a request of type t_j is matched to resource v_i, therefore the expected match on the edge (t_j, v_i) is $np_j x_{ij}$. The goal is to maximize the total expected matched mass:

$$\text{Maximize} \quad Z(x) := \sum_{j=1}^{m} np_j \sum_{v_i \in \Gamma(t_j)} x_{ij}$$

$$\begin{aligned}
\text{subject to:} \quad \Lambda_i(x) := \sum_{j : v_i \in \Gamma(t_j)} np_j x_{ij} \leq 1, \quad & \forall v_i \in V && \text{(Resource Capacity)} \\
\sum_{v_i \in \Gamma(t_j)} x_{ij} \leq 1, \quad & \forall t_j \in T && \text{(Type Matching Limit)} \\
x_{ij} \geq 0, \quad & \forall i, j && \text{(Non-negativity)}
\end{aligned}$$

We call $\Lambda_i(x)$ the (expected fractional) *load* of resource v_i, so that $Z(x) = \sum_i \Lambda_i(x)$. Let $\mathrm{OPT}_{\mathrm{LP}}$ denote the optimal value of the Expected Instance LP. This value is a natural upper bound on the expected size of any matching in the realized graph. Any gap between $\mathrm{OPT}_{\mathrm{LP}}$ and $\mathbb{E}[|M(G)|]$ arises from *stochastic collisions*, events where realized demand assigned to a resource (with respect to x^*_{ij}) exceeds its capacity. Previous work establishes a tight relationship between these two quantities; for completeness, we provide a proof of the following bound in Appendix A.

Proposition 2.3 (Relation between $\mathrm{OPT}_{\mathrm{LP}}$ and $\mathbb{E}_G[|M(G)|]$)**.** *The expected size of the offline maximum matching is bounded by:*

$$\left(1 - \frac{1}{e}\right) \mathrm{OPT}_{\mathrm{LP}} \leq \mathop{\mathbb{E}}_{G}[|M(G)|] \leq \mathrm{OPT}_{\mathrm{LP}}$$

3 Algorithmic Framework: From Global Plans to Local Choices

To navigate the communication bottleneck, we propose a two-stage strategy: an offline phase that computes a globally robust fractional plan via the Expected Instance LP, and an online phase that

uses this plan to guide local edge selection. Our algorithm translates the global fractional plan x^* into local edge selections using Variance-Optimal (VarOpt) sampling.

3.1 Sampling with Hard Constraints: VarOpt

While the LP provides the optimal fractional flow x^*, the local communication budget k requires a discrete selection of edges. A naive approach would be independent Bernoulli sampling, but this lacks control over the exact sample size. We instead utilize Variance-Optimal (VarOpt) sampling [11], a family of unequal probability dependent sampling schemes [10].

Given a set of items $e \in E$ with weights $x_e \geq 0$, let $E^+ := \{e \in E : x_e > 0\}$. VarOpt determines a uniform threshold multiplier $\tau > 0$ and assigns an inclusion probability π_e to each item such that:

$$\pi_e = \min\{1, \tau \cdot x_e\} \quad \text{where} \quad \sum_{e \in E} \pi_e = \min\{k, |E^+|\}. \tag{2}$$

This thresholding mechanism ensures that "heavy" items (where $\tau \cdot x_e \geq 1$) are included deterministically ($\pi_e = 1$), while "light" items are sampled strictly proportional to their weights. VarOpt sampling is very efficient and requires a single pass over the edges [11] (and does not require a root-finding search for τ).

To provide unbiased estimation, each sampled item is assigned an inverse probability weight $w_e := x_e/\pi_e$ (and $w_e := 0$ if e is not sampled), so that $\mathbb{E}[w_e] = x_e$. Our analysis uses only the following properties.

Let S be a VarOpt sample of budget k from items with weights $x_e \geq 0$ satisfying $\sum_e x_e \leq 1$ (as is the case for the edges of a type, by the Type Matching Limit). Then, deterministically:

(V1) $|S| = \min\{k, |E^+|\}$; in particular the local budget is never exceeded.
(V2) $\sum_{e \in S} w_e = \sum_{e \in E} x_e$.
(V3) Every item with $x_e > 1/k$ has $\pi_e = 1$ and $w_e = x_e$; every sampled item with $x_e \leq 1/k$ has $w_e \leq 1/k$; and all weights are at most 1.

(V1) is the defining fixed-size property of VarOpt [11]. For (V2), items with $\pi_e = 1$ are always sampled and contribute x_e, and each other sampled item contributes $x_e/(\tau x_e) = 1/\tau$; by (2) there are exactly $\tau \sum_{e:\pi_e<1} x_e$ of them. For (V3), if $|E^+| < k$ then every positive item has $\pi_e = 1$ and $w_e = x_e$. Otherwise (2) gives $k = \sum_e \min\{1, \tau x_e\} \leq \tau \sum_e x_e \leq \tau$, so $x_e > 1/k$ implies $\tau x_e > 1$ and $\pi_e = 1$, while for $x_e \leq 1/k$ we have $w_e \in \{x_e, 1/\tau\}$, both at most $1/k$.

3.2 Sparsifier Selection and the Role of Solution Geometry

Given a feasible solution x to the Expected Instance LP, we define our local selection strategy. This strategy effectively "rounds" the fractional roadmap provided by the LP into a sparse realized graph.

Definition 3.1 (Sparsifier Selection Strategy $\Phi_{k,x}$)**.** When a request u_r of type t_j arrives, the algorithm observes its realized compatibility set $\Gamma(t_j)$. The strategy $\Phi_{k,x}$ selects a subset of edges $S_r \subseteq \{(u_r, v) \mid v \in \Gamma(t_j)\}$ by applying VarOpt sampling with budget k to the fractional values x_{ij} as the sampling weights.

The performance of this local rule is intrinsically tied to the "shape" of the LP solution. While any optimal solution x yields the same global objective $Z(x) = \mathrm{OPT}_{\mathrm{LP}}$, the local budget k introduces a sensitivity to how fractional mass is distributed. Because requests cannot coordinate to avoid resource collisions, the strategy benefits from solutions that are "spread out" rather than "concentrated."

Observation 3.2 (The Importance of the Fractional Geometry)**.** Consider a fully connected bipartite instance with n resources and n uniform types ($p_j = 1/n$).

- **A Concentrated Solution:** If the LP sets $x_{ij} = 1$ for $i = j$ and 0 otherwise, each type is tied to a specific resource. When a type t_j arrives, the local sparsifier $\Phi_{k,x}$ is forced to pick this single edge regardless of the budget k, and the expected size of the maximum matching is exactly the number of distinct types that arrive, which is $n(1-(1-1/n)^n) \approx n(1-1/e)$.
- **A Spread Solution:** If the LP sets $x_{ij} = 1/n$ for all i, j, the fractional weights are small. The sparsifier $\Phi_{k,x}$ samples a uniformly random set of k edges per arrival. As k increases,

the probability that a single resource is ignored or overwhelmed decreases, and the expected size of the matching approaches n; by Corollary 3.5(c) it is at least $n(1 - 1/\sqrt{2\pi k})$ for $k \leq n$.

This observation motivates the need to carefully select the feasible solution x so that "light" fractional values that spread demand and mitigate stochastic collisions are preferred, leading to the following decomposition.

3.3 Quantifying Sparsifier Quality: Heavy-Light Decomposition

To bridge the gap between the local budget k and global performance, we partition the edges in the support of x based on their value relative to the budget's inverse, $1/k$.

Definition 3.3 (Heavy-Light Decomposition)**.** Given a sparsification parameter k and a feasible fractional solution x, we partition the edges into:

- *Light Edges ($\mathcal{E}_L$):* $\{(i,j) \mid x_{ij} \leq 1/k\}$. These represent demand that can be spread across the budget.
- *Heavy Edges ($\mathcal{E}_H$):* $\{(i,j) \mid x_{ij} > 1/k\}$. These represent concentrated demand that the budget k cannot further dilute.

This partition induces a natural decomposition of the expected fractional objective into $Z(x) = Z_L(x) + Z_H(x)$, where:

$$Z_L(x) = \sum_{(i,j)\in\mathcal{E}_L} np_j x_{ij} \quad \text{and} \quad Z_H(x) = \sum_{(i,j)\in\mathcal{E}_H} np_j x_{ij}.$$

We call $\eta := Z_H(x)/Z(x) \in [0,1]$ the *heavy fraction* of x (for budget k).

We now establish that our sparsifier preserves the light fractional components, confining the stochastic collision penalty to the heavy components. The proof is in Appendix B.

Theorem 3.4 (Sparsifier Approximation Guarantee)**.** *Let x be a feasible solution of the Expected Instance LP with heavy fraction $\eta = Z_H(x)/Z(x)$, and let G_S be the sparsifier constructed by the local strategy $\Phi_{k,x}$ with budget $k \geq 1$. Then $\mathbb{E}[|M(G_S)|] \geq \big(1 - U_k(\eta)\big)\, Z(x)$, where $U_k : [0,1] \to [0, 1/e]$ is the concave function*

$$U_k(\eta) := \min\left\{\frac{1}{e},\ \frac{1}{2}\sqrt{\eta + \frac{1-\eta}{k}},\ e^{-\eta}\Big(\eta + \frac{1}{2}\sqrt{\frac{1-\eta}{k}}\Big)\right\}.$$

We spell out what the bound gives in three regimes of the heavy fraction: arbitrary η (universal), small η (spread), and $\eta = 0$ (fully light):

Corollary 3.5 (Consequences of Theorem 3.4)**.** Under the conditions of Theorem 3.4:

(a) **Universal.** For every feasible x and every $k \geq 1$, $\mathbb{E}[|M(G_S)|] \geq (1 - 1/e)\, Z(x)$. For $k = 1$ this is tight: on the complete bipartite graph $K_{n,n}$ with a single request type, every local rule with $k = 1$ has $\alpha \leq 1 - (1 - 1/n)^n$.
(b) **Spread.** If $\eta \leq \epsilon/2$ and $k \geq \epsilon^{-2}$, then $\mathbb{E}[|M(G_S)|] \geq (1 - \epsilon)\, Z(x)$.
(c) **Fully light.** If $\eta = 0$, then $\mathbb{E}[|M(G_S)|] \geq \big(1 - \frac{e^{-k}k^k}{k!}\big)Z(x) \geq \big(1 - \frac{1}{\sqrt{2\pi k}}\big)Z(x)$.

In each case, if $Z(x) \geq \mathbb{E}[|M(G)|]$ (e.g., if x is optimal for the LP), the same bound holds for the preservation ratio α.

Proof. (a) follows from $U_k \leq 1/e$. For tightness, a $k = 1$ local rule on $K_{n,n}$ with a single type sends each request to an independent resource drawn from a fixed distribution q, so $\mathbb{E}[|M(G_S)|] = \sum_v \big(1 - (1 - q_v)^n\big) \leq n\big(1 - (1 - 1/n)^n\big)$ by concavity, while $|M(G)| = n$. (b) $U_k(\eta) \leq e^{-\eta}\big(\eta + \frac{1}{2}\sqrt{(1-\eta)/k}\big) \leq \eta + \frac{1}{2\sqrt{k}} \leq \frac{\epsilon}{2} + \frac{\epsilon}{2}$. (c) is proved in Appendix B (Theorem B.6 and Lemma B.4(e)). The last sentence uses $\mathrm{OPT}_{\mathrm{LP}} \geq \mathbb{E}[|M(G)|]$ (Proposition 2.3). □

Heavy edges are never pruned. By (V3), every heavy edge is forwarded, so the loss that Theorem 3.4 charges to heavy mass is a collision loss relative to $Z(x)$, which can exceed what any matching achieves. For example, with disjoint neighborhoods $\Gamma(t_j) = \{v_j\}$ and uniform types, every optimal x has $\eta = 1$ and the theorem gives only $1 - 1/e$ relative to $Z(x)$, yet $G_S = G$ and $\alpha = 1$.

Corollary 3.6 (Concentration)**.** For every $\delta \in (0, 1)$, with probability at least $1 - \delta$, $|M(G_S)| \geq \big(1 - U_k(\eta)\big)Z(x) - \sqrt{\frac{n}{2} \ln \frac{1}{\delta}}$. If moreover $Z(x) \geq \mathbb{E}[|M(G)|]$, then with probability at least $1 - \delta$, $|M(G)| - |M(G_S)| \leq U_k(\eta)\, Z(x) + \sqrt{2n \ln \frac{1}{\delta}}$.

Proof. Let $\omega_r = (R_r, \xi_r)$, where ξ_r is the local randomness of request r. Both $|M(G)|$ and $|M(G_S)|$ are functions of the independent $\omega_1, \ldots, \omega_n$. Because $\Phi_{k,x}$ is local, changing a single ω_r changes only the edges incident to u_r, in G and in G_S, and hence changes each maximum matching size by at most one. Thus $|M(G_S)|$ has bounded differences 1 and $|M(G)| - |M(G_S)|$ has bounded differences 2. By McDiarmid's inequality, each of the following holds with probability at least $1 - \delta$: $|M(G_S)| \geq \mathbb{E}[|M(G_S)|] - \sqrt{\frac{n}{2} \ln \frac{1}{\delta}}$ and $|M(G)| - |M(G_S)| \leq \mathbb{E}[|M(G)|] - \mathbb{E}[|M(G_S)|] + \sqrt{2n \ln \frac{1}{\delta}}$. Combine the first with Theorem 3.4, and the second with Theorem 3.4 and $\mathbb{E}[|M(G)|] \leq Z(x)$. The same argument gives two-sided bounds: each of $|M(G)|$ and $|M(G_S)|$ is within $\sqrt{\frac{n}{2} \ln \frac{2}{\delta}}$ of its mean with probability at least $1 - \delta$. □

In practice, it is often the case that large sets of resources are highly symmetric or interchangeable. The following claim demonstrates that we can exploit these symmetries to naturally spread the fractional demand, shifting weight from Z_H to Z_L.

Claim 3.7 (Spread Solutions via Equivalence Classes)**.** We say two resources v, v' are interchangeable if for all types $t_j \in T$, they share the exact same compatibility: $v \in \Gamma(t_j) \iff v' \in \Gamma(t_j)$. If an instance contains an equivalence class of interchangeable resources of size ℓ, there exists an optimal fractional solution x^* where all edges incident to this class are ℓ-light (i.e., $x^*_{ij} \leq 1/\ell$).

Proof. Let x be any optimal solution. For any type t_j and the equivalence class C of size ℓ, we construct x^* by uniformly averaging the fractional weights across the class: $x^*_{ij} = \frac{1}{\ell} \sum_{v \in C} x_{v,j}$ for all $v_i \in C$. Because the resources are strictly interchangeable, this uniform redistribution preserves both the type constraints and the global objective value $Z(x)$, and the load of every resource in C becomes the average load over C, which is at most 1. Furthermore, since the type matching limit enforces $\sum_{v \in C} x_{v,j} \leq 1$, the averaged weight on any single edge incident to the class is bounded by $x^*_{ij} \leq 1/\ell$. □

4 Experiments

We evaluate our sparsifier in two environments: (1) a large-scale ride-hailing simulation using real-world taxi data, and (2) synthetic adversarial benchmarks designed to stress-test online matching algorithms. We compare against standard online algorithms and a heuristic sparsification strategy.

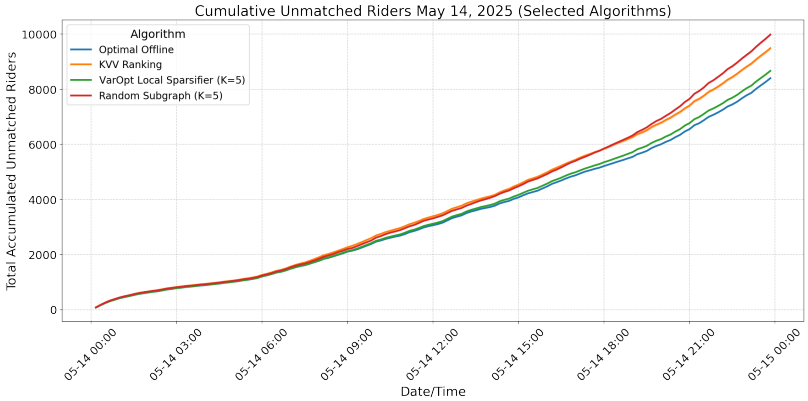
4.1 Real-World Application: NYC Ride-Hailing

Dataset and Graph Construction. We utilize the NYC Yellow Taxi Trip Data from May 14, 2025[1]. The spatial framework is discretized into taxi zones, with an adjacency matrix A where $A_{uv} = 1$ if zones u and v share a boundary.

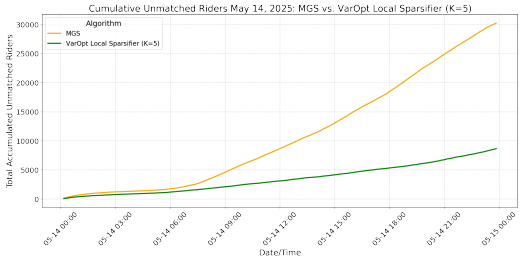
The simulation proceeds in 10-minute intervals. For each interval t, we construct a bipartite graph $G_t = (U_t \cup V_t, E_t)$ where:

- V_t (Offline/Supply): Vehicles sampled from drop-off events in $[t - 5\text{m}, t]$.
- U_t (Online/Demand): Riders sampled from pick-up events in $[t, t + 5\text{m}]$.

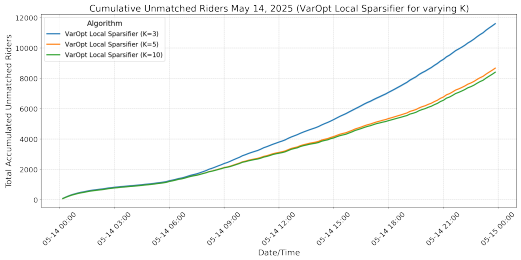
[1] https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page



(a) Baseline comparison of cumulative unmet demand. The VarOpt local sparsifier ($k = 5$) outperforms uniform random selection and KVV baseline and tracks closely with the full-information optimal offline.



(b) Performance against the standard MGS algorithm.



(c) Ablation over the budget $k \in \{3, 5, 10\}$. Increasing k from 3 to 5 markedly reduces unmet demand, while $k = 10$ yields only marginal further gains.

Figure 1: Empirical evaluation of stochastic matching algorithms on NYC taxi data (May 14, 2025). The plots are reported with $\pm 95\%$ Confidence Intervals. Because the primary metric aggregates cumulative unmet demand over time, the variance remains exceedingly small relative to the total volume. Consequently, these confidence bands are tight and visibly indistinguishable from the mean lines.

- E_t: An edge exists if the car's drop-off zone and the rider's pick-up zone are identical or adjacent.

The construction of graph G_t is detailed in Section C.1.

Evaluated Algorithms. We compare five strategies to observe the impact of limited information and restricted choice:

1. **Optimal Offline:** The maximum matching $M(G_t)$ computed via Hopcroft-Karp [21] with full knowledge of the interval. This is the absolute upper bound.
2. **KVV Ranking [30]:** The classic online algorithm assigning random ranks to offline cars; arriving riders greedily pick the highest-ranked available neighbor.
3. **Random Subgraph:** A naive sparsifier where riders only consider a random subset of at most k valid edges.
4. **MGS Algorithm [32]:** Following Algorithm 1 in [32], this approach uses offline statistics to guide online matching decisions. Specifically, it computes the marginal probability that each edge appears in an optimal offline matching and then it samples two matchings as guidance for the online algorithm.
5. **VarOpt Local Sparsifier:** Our proposed method. We estimate the conditional match probabilities $x_{ij} = \mathbb{P}((u_j, v_i) \in M(G_t))$ by averaging the results of 100 Monte Carlo simulations (details provided in Section C.2). Arriving riders then apply VarOpt sampling (see Algorithm 3) to select k edges based on these weights.

Table 1: Average matching efficiency (approximation ratio) across synthetic adversarial benchmarks. Each graph consists of 100 vertices per side. Results are averaged over 100 independent trials, with 100 Monte Carlo simulations used for offline learning. Values are reported as Mean $\pm$ 95% Confidence Interval.

			Random Subgraph			VarOpt Local Sparsifier		
Adversarial Graph	KVV	MGS	K=3	K=5	K=10	K=3	K=5	K=10
Partitioned Block [29]	$82.82\% \pm 0.54\%$	$70.07\% \pm 1.07\%$	$74.16\% \pm 0.65\%$	$77.48\% \pm 0.73\%$	$83.47\% \pm 0.55\%$	$93.61\% \pm 0.50\%$	$98.44\% \pm 0.27\%$	$99.97\% \pm 0.04\%$
KVV Upper Triangular	$91.28\% \pm 0.45\%$	$68.53\% \pm 0.89\%$	$76.27\% \pm 0.53\%$	$84.53\% \pm 0.56\%$	$93.33\% \pm 0.42\%$	$95.21\% \pm 0.40\%$	$99.11\% \pm 0.16\%$	$99.98\% \pm 0.03\%$
Bahmani 0.901 Bound [5]	$83.40\% \pm 0.53\%$	$68.78\% \pm 0.72\%$	$61.77\% \pm 0.77\%$	$66.49\% \pm 0.87\%$	$76.80\% \pm 0.84\%$	$89.69\% \pm 0.44\%$	$95.11\% \pm 0.40\%$	$99.26\% \pm 0.17\%$
TSM Tight [16]	$94.58\% \pm 0.42\%$	$74.93\% \pm 0.95\%$	$98.13\% \pm 0.44\%$	$99.61\% \pm 0.20\%$	$99.93\% \pm 0.08\%$	$97.54\% \pm 0.50\%$	$99.58\% \pm 0.19\%$	$99.96\% \pm 0.05\%$

Results. Our primary metric is the volume of unmatched riders (unmet demand). As shown in Figure 1, the *VarOpt Local Sparsifier* outperforms the other baseline algorithms in this metric. Furthermore, when $k = 10$, its performance closely approaches the offline optimal.

Over the 24-hour simulation, the *VarOpt Local Sparsifier* with $k = 5$ yields significantly fewer unmatched riders than the naive *Random Subgraph* with $k = 5$ and than KVV (Figure 1a), and vastly fewer than the MGS baseline in this high-variance setting (Figure 1b). In the budget ablation (Figure 1c), moving from $k = 3$ to $k = 5$ markedly reduces unmet demand, while the gap between $k = 5$ and $k = 10$ is minimal, consistent with our theoretical claim that a small k suffices to capture most of the optimal matching.

4.2 Synthetic Tests

We evaluate the algorithms against structured hard examples from the literature of online matching. These adversarial benchmarks are designed to highlight the failure modes of standard algorithms and demonstrate the robustness of probabilistic edge selection.

The Partitioned Block Graph (Karande et al. [29]): Following Section 4.2 of [29], we construct a bipartite graph over n vertices partitioned into three row blocks (R_1, R_2, R_3) and three column blocks (C_1, C_2, C_3). The column and row blocks contain $0.3n$, $0.4n$, and $0.3n$ vertices, respectively. Edges exist on the main diagonal ($i = j$), between R_1 and C_2, and between R_2 and C_3. Arriving columns are random, while rows are randomly permuted. Under random-order arrivals this structure limits KVV Ranking to a ratio of about 0.727; this bound does not directly apply to the i.i.d. arrival setting considered here.

The TSM Tightness Graph (Feldman et al. [16]): We implement the construction from Section 4.2.5 of [16], originally designed as a tight adversarial instance for the Two-Suggested-Matching (TSM) algorithm. For a graph of size n (a multiple of 4), the structure consists of $n/4$ disjoint 6-cycles ($\{u_i - x_i - v_i - y_i - w_i - z_i - u_i\}$) interspersed with two dense complete bipartite subgraphs (between K and X, and L and W). The optimal matching achieves an assignment size of $n(1 - 1/2e)$.

The KVV Upper Triangular Graph: We test the classic adversarial construction for the Ranking algorithm. In this $n \times n$ graph, arriving node i is connected to offline nodes $\{i, i + 1, \ldots, n\}$. Our VarOpt local sparsifier leverages its offline Monte Carlo weights to heavily prioritize the exact diagonal.

The Online Upper-Bound Graph (Bahmani and Kapralov [5]): We implement the graph construction from Theorem 7 in [5], which establishes an upper bound of $1 - 1/(e + e^2) \approx 0.901$ for any online stochastic matching algorithm. The bipartite graph features offline nodes $A_1 \cup A_2$ and online types $I_1 \cup I_2$, sized such that $|I_1| = |A_2| = n$ and $|A_1| = |I_2| = n/e$. The edge set consists of a hidden perfect matching between I_1 and A_2, obscured by complete bipartite subgraphs between $I_2 \cup A_2$ and $I_1 \cup A_1$. No online algorithm, which must commit to a single edge upon arrival, can exceed ≈ 0.901 on this instance; we use it to show that returning $k > 1$ edges bypasses this limit.

Spread of the learned weights. Table 2 reports the heavy fraction of the Monte Carlo weights. Since an edge is heavy when $x_{ij} > 1/k$, η is nondecreasing in k. On NYC the weights are light: in our construction compatibility depends only on zones, so vehicles sharing a drop-off zone are interchangeable in the sense of Claim 3.7, and heavy edges arise only for riders whose accessible supply is small. The spread condition of Corollary 3.5(b), $\eta \leq 1/(2\sqrt{k})$ (i.e., $28.9\%/22.4\%/15.8\%$), holds for NYC and KVV at $k = 3, 5$, and the other three synthetic instances are substantially heavy. The certified fractions are far below the measured ratios: on Partitioned Block, 65% of the objective is

Table 2: Heavy fraction $\eta = Z_H/Z$ of the Monte Carlo weights and the fraction $1 - \widehat{U}_k(\eta)$ of Z certified by Theorem B.6. The weights do not depend on k, and a certificate for budget k' also holds for every budget $k \geq k'$ (Theorem B.6), so we report the best certificate over $k' \leq k$ (*: attained at $k' = 5$). The Monte Carlo objective Z equals the average maximum matching size (Lemma C.1), so these fractions are comparable to the measured ratios in Table 1. For NYC, η is averaged over the 10-minute intervals; by concavity of $\widehat{U}_k$, the reported value lower-bounds the average certified fraction.

	Heavy fraction η (%)			Certified fraction of Z		
Instance	$k = 3$	$k = 5$	$k = 10$	$k = 3$	$k = 5$	$k = 10$
NYC taxi	4.69	11.09	21.76	0.77	0.79	0.79
KVV Upper Triangular	1.05	7.90	44.46	0.77	0.80	0.80^*
TSM Tight	38.52	41.81	47.17	0.70	0.71	0.71^*
Bahmani 0.901 Bound	45.51	45.51	45.51	0.69	0.70	0.71
Partitioned Block	64.79	65.36	65.36	0.66	0.66	0.66

heavy and the certificate is 0.66, while the measured ratio is 93.6% already at $k = 3$. The certificates are pessimistic, especially on heavy mass, because they charge for collisions that the Monte Carlo weights already account for (Lemma C.1).

Computational Resources All experiments were conducted on a standard cloud-based computing platform. The runtime environment for each simulation was configured with 2 vCPUs and 32 GB of system RAM.

5 Conclusion

Limitations. Our model assumes that requests arrive i.i.d. from a known distribution $\mathcal{D}$. The algorithm uses $\mathcal{D}$ only through a feasible fractional solution x. In our experiments, x is estimated by Monte Carlo simulation from $\mathcal{D}$. These weights are exactly feasible for the empirical arrival frequencies (Lemma C.1), but under $\mathcal{D}$ itself, sampling error can push some resources slightly over capacity. By Theorem B.6, this costs at most the total excess load $\sum_i (\Lambda_i - 1)_+$, which vanishes as the number of simulations grows.

Conclusion. We introduced a two-stage local sparsification framework for stochastic bipartite matching, motivated by communication and latency bottlenecks in decentralized platforms, and a VarOpt-based selection strategy guided by a fractional solution of the expected instance. The sparsifier retains a $1 - 1/e$ fraction of the fractional benchmark for every budget (tight for $k = 1$) and a $1 - O(1/\sqrt{k})$ fraction for well-spread solutions. Empirically it vastly outperforms random selection and strict online algorithms on real-world and adversarial instances.

Open questions and follow-up work. A natural question raised by our analysis is whether some local sparsification strategy achieves $\mathbb{E}[|M(G_S)|]/\mathbb{E}[|M(G)|] \to 1$ as k grows across *all* graph topologies, including those whose fractional solutions are necessarily heavy. In follow-up work [1], we answer this affirmatively for the broader model of independent, not necessarily identical, vertex arrivals: for every $\epsilon > 0$ there is a budget k_ϵ depending only on ϵ such that menus derived from a benchmark matching rule, including fixed-size VarOpt samples of its edge marginals, preserve a $1 - \epsilon$ fraction of $\mathbb{E}[|M(G)|]$ without any spread assumption. The budget k_ϵ obtained there is doubly exponential in $1/\epsilon$, whereas in that model some instances force every local rule to lose an $\Omega(1/k)$ fraction. Determining the optimal dependence of the loss on k remains open; the present analysis gives $O(1/\sqrt{k})$ for well-spread solutions, and it is natural to ask whether $\Theta(1/k)$ is achievable in general.

Acknowledgments and Disclosure of Funding

Edith Cohen is partially supported by the Israel Science Foundation (grant no. 1156/23).

References

[1] Sara Ahmadian, Edith Cohen, and Mohammad Roghani. Locally sparsified, globally near-optimal: Matching under independent vertex arrivals. arXiv:2609.27161, 2026.

[2] Jonathan Aronson, Martin E. Dyer, Alan M. Frieze, and Stephen Suen. Randomized greedy matching II. *Random Struct. Algorithms*, 6(1):55–73, 1995. doi: 10.1002/RSA.3240060107. URL https://doi.org/10.1002/rsa.3240060107.

[3] Sepehr Assadi and Soheil Behnezhad. Beating two-thirds for random-order streaming matching. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, Glasgow, Scotland (Virtual Conference), July 12-16, 2021*, LIPIcs, pages 19:1–19:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi: 10.4230/LIPICS.ICALP.2021.19. URL https://doi.org/10.4230/LIPIcs.ICALP.2021.19.

[4] Amir Azarmehr, Soheil Behnezhad, Mohammad Roghani, and Aviad Rubinstein. Tight pair query lower bounds for matching and earth mover's distance. In *Proceedings of the 66th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2666–2693, 2025. doi: 10.1109/FOCS63196.2025.00138.

[5] Bahman Bahmani and Michael Kapralov. Improved bounds for online stochastic matching. In *Proceedings of the 18th Annual European Symposium on Algorithms (ESA)*, pages 170–181. Springer, 2010. doi: 10.1007/978-3-642-15775-2_15. URL https://theory.epfl.ch/kapralov/papers/stochastic-final.pdf.

[6] Soheil Behnezhad, Mohammad Roghani, and Aviad Rubinstein. Approximating maximum matching requires almost quadratic time. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 444–454, 2024. doi: 10.1145/3618260.3649785.

[7] Aaron Bernstein. Improved bounds for matching in random-order streams. *Theory of Computing Systems*, 68(4):758–772, 2024. doi: 10.1007/s00224-023-10155-7.

[8] T.-H. Hubert Chan, Fei Chen, Xiaowei Wu, and Zhichao Zhao. Ranking on arbitrary graphs: Rematch via continuous LP with monotone and boundary condition constraints. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, pages 1112–1122. SIAM, 2014. doi: 10.1137/1.9781611973402.82. URL https://doi.org/10.1137/1.9781611973402.82.

[9] T-H Hubert Chan, Fei Chen, and Xiaowei Wu. Analyzing node-weighted oblivious matching problem via continuous LP with jump discontinuity. *ACM Transactions on Algorithms (TALG)*, 14(2):12:1–12:25, 2018. doi: 10.1145/3168008.

[10] Ming-Yi Chao. A general purpose unequal probability sampling plan. *Biometrika*, 69(3): 653–656, 1982. doi: 10.1093/biomet/69.3.653.

[11] Edith Cohen, Nick Duffield, Haim Kaplan, Carsten Lund, and Mikkel Thorup. Efficient stream sampling for variance-optimal estimation of subset sums. *SIAM Journal on Computing*, 40(5): 1402–1431, 2011. doi: 10.1137/10079817X. URL https://doi.org/10.1137/10079817X.

[12] Varsha Dani and Thomas P Hayes. A sublinear-time algorithm for nearly-perfect matchings in regular non-bipartite graphs. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4899–4913. SIAM, 2025. doi: 10.1137/1.9781611978322.166.

[13] Mahsa Derakhshan and Tao Yu. A unified framework for analysis of randomized greedy matching algorithms. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1960–1971. ACM, 2026. doi: 10.1145/3798129.3800901.

[14] Mahsa Derakhshan, Mohammad Roghani, Mohammad Saneian, and Tao Yu. A simple analysis of Ranking in general graphs. In *2026 Symposium on Simplicity in Algorithms (SOSA)*, pages 438–443. SIAM, 2026. doi: 10.1137/1.9781611978964.36.

[15] Mahsa Derakhshan, Mohammad Roghani, Mohammad Saneian, and Tao Yu. Improved approximation for Ranking on general graphs. In *Proceedings of the 2026 ACM-SIAM Symposium on Discrete Algorithms, SODA 2026*, pages 52–84. SIAM, 2026. doi: 10.1137/1.9781611978971.3.

[16] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating $1-1/e$. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 117–126. IEEE, 2009. doi: 10.1109/FOCS.2009.72. URL https://doi.org/10.1109/FOCS.2009.72.

[17] Buddhima Gamlath, Michael Kapralov, Andreas Maggiori, Ola Svensson, and David Wajc. Online matching with general arrivals. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 26–37. IEEE, 2019. doi: 10.1109/FOCS.2019.00011. URL https://arxiv.org/abs/1904.08255.

[18] Ashish Goel, Michael Kapralov, and Sanjeev Khanna. Perfect matchings in O(n log n) time in regular bipartite graphs. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, pages 39–46, 2010.

[19] Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 468–485. SIAM, 2012.

[20] Bernhard Haeupler, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Online stochastic weighted matching: Improved approximation algorithms. In *Proceedings of the 7th International Workshop on Internet and Network Economics (WINE)*, volume 7090 of *Lecture Notes in Computer Science*, pages 170–181. Springer, 2011. doi: 10.1007/978-3-642-25510-6_15.

[21] John E Hopcroft and Richard M Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973. doi: 10.1137/0202019.

[22] Zhiyi Huang, Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. Online vertex-weighted bipartite matching: Beating $1-1/e$ with random arrivals. *ACM Transactions on Algorithms (TALG)*, 15(3):38:1–38:15, 2019. doi: 10.1145/3326169.

[23] Zhiyi Huang, Enze Sun, Xiaowei Wu, and Jiahao Zhao. Edge-weighted matching in the dark. In *66th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2025, Sydney, Australia, December 14-17, 2025*, pages 797–817. IEEE, 2025. doi: 10.1109/FOCS63196.2025.00043. URL https://doi.org/10.1109/FOCS63196.2025.00043.

[24] Patrick Jaillet and Xin Lu. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research*, 39(3):624–646, 2014. doi: 10.1287/moor.2013.0621.

[25] Billy Jin and David P Williamson. Improved analysis of RANKING for online vertex-weighted bipartite matching in the random order model. In *International Conference on Web and Internet Economics (WINE)*, pages 207–225. Springer, 2021. doi: 10.1007/978-3-030-94676-0_12.

[26] Michael Kapralov. Better bounds for matchings in the streaming model. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1679–1697. SIAM, 2013. doi: 10.1137/1.9781611973105.121.

[27] Michael Kapralov. Space lower bounds for approximating maximum matching in the edge arrival model. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1874–1893. SIAM, 2021.

[28] Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Approximating matching size from random streams. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 734–751. SIAM, 2014. doi: 10.1137/1.9781611973402.55. URL https://doi.org/10.1137/1.9781611973402.55.

[29] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 587–596. ACM, 2011. doi: 10.1145/1993636.1993715. URL https://doi.org/10.1145/1993636.1993715.

[30] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for online bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 352–358. ACM, 1990. doi: 10.1145/100216.100262. URL https://doi.org/10.1145/100216.100262.

[31] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: An approach based on strongly factor-revealing LPs. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC)*, pages 597–606. ACM, 2011. doi: 10.1145/1993636.1993716.

[32] Vahideh H. Manshadi, Shayan Oveis Gharan, and Amin Saberi. Online stochastic matching: Online actions based on offline statistics. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1285–1294. SIAM, 2011. doi: 10.1137/1.9781611973082.98. URL https://epubs.siam.org/doi/10.1137/1.9781611973082.98.

[33] Bo Peng and Zhihao Gavin Tang. Revisiting Ranking for online bipartite matching with random arrivals: the primal-dual analysis. In *Proceedings of the 26th ACM Conference on Economics and Computation (EC)*, pages 820–835, 2025. doi: 10.1145/3736252.3742631.

[34] Matthias Poloczek and Mario Szegedy. Randomized greedy algorithms for the maximum matching problem with new analysis. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012*, pages 708–717. IEEE Computer Society, 2012. doi: 10.1109/FOCS.2012.20. URL https://doi.org/10.1109/FOCS.2012.20.

[35] Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. Towards a better understanding of randomized greedy matching. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020*, pages 1097–1110. ACM, 2020. doi: 10.1145/3357713.3384265. URL https://doi.org/10.1145/3357713.3384265.

A Proof of Proposition 2.3

The proof is divided into two parts: showing the upper bound via the LP relaxation, and proving the lower bound via an independent randomized assignment. Let x^* be an optimal fractional solution to the Expected Instance LP, such that $\sum_{i,j} np_j x^*_{ij} = \text{OPT}_{\text{LP}}$.

Upper Bound Consider any realized graph G and let $M(G)$ be the maximum bipartite matching on this graph. We can construct a corresponding expected fractional matching $\bar{x}$. Let $\bar{x}_{ij}$ be the expected number of times a request of type t_j is matched to resource v_i in $M(G)$, normalized by the expected number of arrivals np_j. Because $M(G)$ is a valid matching for every realization, it must respect the matching constraint: (1) $\sum_j np_j \bar{x}_{ij} \leq 1$, and (2) $\sum_i \bar{x}_{ij} \leq 1$. Since $\bar{x}_{ij}$ satisfies all the constraints of the Expected Instance LP, it is a feasible solution. Because x^* is the optimal solution, the objective value produced by $\bar{x}$ cannot exceed OPT_{LP}:

$$\mathbb{E}[|M(G)|] = \sum_{j=1}^{m} np_j \sum_{v_i \in \Gamma(t_j)} \bar{x}_{ij} \leq \text{OPT}_{\text{LP}}$$

Lower Bound To prove the lower bound, we consider the following randomized routing strategy. Because the maximum offline matching $M(G)$ is optimal, its size is bounded below by the number of matches produced by any valid allocation strategy. Suppose that upon arrival, each of the n requests independently draws its type from distribution D. If the request is of type t_j, it independently chooses to route a match to a compatible resource v_i with probability x^*_{ij}. (Because $\sum_i x^*_{ij} \leq 1$, this is a

valid probability distribution). Let Λ_i be the expected number of routing attempts directed at resource v_i. From the LP constraints, we know:

$$\Lambda_i = \sum_j np_j x^*_{ij} \leq 1$$

Since there are exactly n independent arrivals, the probability that resource v_i receives exactly zero routing attempts is:

$$\mathbb{P}(v_i \text{ is unmatched}) = \left(1 - \frac{\sum_j np_j x^*_{ij}}{n}\right)^n = \left(1 - \frac{\Lambda_i}{n}\right)^n$$

A resource is successfully matched if it receives at least one routing attempt (any collisions are simply dropped). The probability that resource v_i is matched is therefore $1 - (1 - \Lambda_i/n)^n$. Since this function is concave in Λ_i and vanishes at $\Lambda_i = 0$, for $\Lambda_i \in [0, 1]$ we have:

$$1 - \left(1 - \frac{\Lambda_i}{n}\right)^n \geq \left(1 - \left(1 - \frac{1}{n}\right)^n\right)\Lambda_i \geq \left(1 - \frac{1}{e}\right)\Lambda_i$$

By linearity of expectation, we sum the probability of being matched across all resources $v_i \in V$ to bound the expected size of this matching, which in turn lower bounds the maximum matching:

$$\mathbb{E}[|M(G)|] \geq \sum_{i \in V}\left(1 - \frac{1}{e}\right)\Lambda_i = \left(1 - \frac{1}{e}\right)\sum_{i \in V}\sum_{j=1}^{m} np_j x^*_{ij} = \left(1 - \frac{1}{e}\right)\text{OPT}_{\text{LP}}$$

This concludes the proof. (The lower bound also follows from Corollary 3.5(a) with $k = 1$, since $G_S \subseteq G$.)

B Proof of Theorem 3.4

We prove a per-resource statement (Theorem B.6), from which Theorem 3.4 and Corollary 3.5(c) follow. Throughout, fix an integer $k \geq 1$ and a nonnegative x satisfying the type constraints $\sum_i x_{ij} \leq 1$; the capacity constraints are used only where stated. For each resource v_i let

$$\Lambda_i := \sum_j np_j x_{ij}, \qquad H_i := \sum_{j:\, x_{ij} > 1/k} np_j x_{ij}, \qquad \eta_i := H_i/\Lambda_i$$

(with $\eta_i := 0$ if $\Lambda_i = 0$) denote its load, its heavy load, and its heavy fraction, so that $Z(x) = \sum_i \Lambda_i$ and $Z_H(x) = \sum_i H_i$.

B.1 A fractional matching on the sparsifier

Our strategy is to specify a feasible fractional matching f on the realized sparsifier G_S and bound its expected size. Since the bipartite matching polytope is integral, $|M(G_S)| \geq \|f\|_1$ for every realization.

For each arriving request u of type t_j and each sampled edge $e = (u, v_i) \in S_u$, let $w_e := x_{ij}/\pi_{ij}$ be its VarOpt weight. By (V2), $\sum_{e \in S_u} w_e = \sum_i x_{ij} \leq 1$, so the request-side constraints hold. By (V3), a sampled light edge has $w_e \leq 1/k$, a heavy edge has $\pi_{ij} = 1$ and $w_e = x_{ij} \in (1/k, 1]$, and all weights are at most 1. Let $Y_i := \sum_{e \in E_S \cap \delta(v_i)} w_e$ be the total weight arriving at v_i, and scale down the edges at overloaded resources:

$$f_e := \frac{w_e}{\max(1, Y_i)} \quad \text{for edge } e = (u, v_i).$$

Then f is a fractional matching on G_S and $\|f\|_1 = \sum_i \min\{Y_i, 1\} = \sum_i \big(Y_i - (Y_i - 1)_+\big)$. Since $\mathbb{E}[Y_i] = \sum_j np_j \pi_{ij} w_{ij} = \Lambda_i$,

$$\mathbb{E}[|M(G_S)|] \;\geq\; \mathbb{E}[\|f\|_1] \;=\; Z(x) - \sum_{i \in V}\mathbb{E}[\psi(Y_i)], \qquad \psi(y) := (y - 1)_+. \tag{3}$$

It remains to bound the expected overflow $\mathbb{E}[\psi(Y_i)]$ of each resource separately.

B.2 Reduction to a two-size compound Poisson load

A finitely supported measure μ on $(0,1]$ describes a stream of items: for each size w, $\mu(\{w\})$ is the expected number of items of size w. The corresponding compound Poisson random variable is $\mathrm{CP}(\mu) := \sum_w wN_w$, where the counts $N_w \sim \mathrm{Poisson}(\mu(\{w\}))$ are independent. The *load* of μ is the mean $\mathbb{E}[\mathrm{CP}(\mu)] = \sum_w w\,\mu(\{w\})$. Equivalently, $\mathrm{CP}(\mu) = \sum_{l=1}^{N} W_l$ with $N \sim \mathrm{Poisson}(\mu((0,1]))$ and W_l i.i.d. with law $\mu/\mu((0,1])$. We write $X \preceq_{\mathrm{cx}} X'$ if $\mathbb{E}[\phi(X)] \le \mathbb{E}[\phi(X')]$ for every convex ϕ. We use two elementary facts: (F1) if $\mathbb{E}[X' \mid \mathcal{G}] = X$ for a σ-algebra $\mathcal{G}$ with X $\mathcal{G}$-measurable, then $X \preceq_{\mathrm{cx}} X'$ (conditional Jensen); (F2) if $X \preceq_{\mathrm{cx}} X'$ and B is independent of both, then $X + B \preceq_{\mathrm{cx}} X' + B$ (condition on B). We also use repeatedly that for convex ϕ and $a, b \ge 0$,

$$\phi(y + a + b) - \phi(y + a) - \phi(y + b) + \phi(y) \;\ge\; 0, \tag{4}$$

since increments of a convex function over intervals of equal length are nondecreasing.

Lemma B.1 (Poisson domination of the resource load)**.** *For every resource v_i, $Y_i \preceq_{\mathrm{cx}} \mathrm{CP}(\nu_i)$, where $\nu_i := \sum_j np_j\pi_{ij}\,\delta_{w_{ij}}$ has load Λ_i.*

Proof. Requests are i.i.d. (type and local randomness), and each contributes at most one edge at v_i. Hence $Y_i = \sum_{r=1}^{n} X_r$ for i.i.d. X_r taking value w_{ij} with probability $p_j\pi_{ij}$ for each j, and 0 otherwise. Let $\mu := \nu_i/n$, $q := \mu((0,1]) \le 1$, and let W have law μ/q, so $X_r \overset{d}{=} B \cdot W$ with $B \sim \mathrm{Bern}(q)$ independent of W. Fix a convex ϕ and let $h(m) := \mathbb{E}[\phi(W_1 + \cdots + W_m)]$ for i.i.d. copies W_l of W. By exchangeability and (4), $h(m+2) - 2h(m+1) + h(m) \ge 0$, so the sequence h is convex and lies above its chord through $m = 0, 1$: $h(m) \ge h(0) + m\,(h(1) - h(0))$. For $C \sim \mathrm{CP}(\mu)$, i.e., $C = \sum_{l \le N} W_l$ with $N \sim \mathrm{Poisson}(q)$,

$$\mathbb{E}[\phi(C)] = \mathbb{E}[h(N)] \ge h(0) + q\,(h(1) - h(0)) = (1-q)\phi(0) + q\,\mathbb{E}[\phi(W)] = \mathbb{E}[\phi(X_r)].$$

Thus $X_r \preceq_{\mathrm{cx}} C$. Replacing $X_1, \ldots, X_n$ one at a time by independent copies of C and applying (F2) gives $Y_i \preceq_{\mathrm{cx}} \mathrm{CP}(n\mu) = \mathrm{CP}(\nu_i)$. □

Lemma B.2 (Rounding item sizes up)**.** *Let μ be a finitely supported measure on $(0,1]$, and let $0 < s < s' \le 1$ and $0 < a \le \mu(\{s\})$. Let $\mu' := \mu - a\delta_s + \frac{as}{s'}\delta_{s'}$, i.e., rate a of size-s items is replaced by rate as/s' of size-s' items. Then μ and μ' have the same load, and $\mathrm{CP}(\mu) \preceq_{\mathrm{cx}} \mathrm{CP}(\mu')$.*

Proof. Write $\mathrm{CP}(\mu) = B + sN$, where $N \sim \mathrm{Poisson}(a)$ and B is the independent load of the remaining intensity. Given (B, N), draw $N' \sim \mathrm{Bin}(N, s/s')$. By Poisson thinning, $N' \sim \mathrm{Poisson}(as/s')$ independently of B, so $B + s'N' \sim \mathrm{CP}(\mu')$, and $\mathbb{E}[B + s'N' \mid B, N] = B + sN$. The claim follows from (F1). □

Applying Lemma B.2 atom by atom, we may move every heavy atom of ν_i (size $w_{ij} = x_{ij} \in (1/k, 1]$) to size 1 and every light atom (size $w_{ij} \le 1/k$) to size $1/k$, preserving loads. By transitivity, with $L_i := \Lambda_i - H_i$,

$$\mathrm{CP}(\nu_i) \preceq_{\mathrm{cx}} \mathrm{CP}\big(\Lambda_i\bar{\mu}_i\big), \qquad \bar{\mu}_i := \eta_i\,\delta_1 + k(1-\eta_i)\,\delta_{1/k}, \tag{5}$$

where $\bar{\mu}_i$ has load exactly 1 and heavy fraction η_i.

Lemma B.3 (Sub-proportionality of overflow)**.** *Let $\bar{\mu}$ be a finitely supported measure on $(0,1]$ with load 1, and let $g(s) := \mathbb{E}[\psi(\mathrm{CP}(s\bar{\mu}))]$ for $s \ge 0$. Then g is convex with $g(0) = 0$. Consequently $g(s) \le s\,g(1)$ for $s \in [0,1]$, and $g(s) \le g(1) + (s-1)$ for $s \ge 1$.*

Proof. For a rate vector $r = (r_w)$, let $F(r) := \mathbb{E}\big[\psi\big(\sum_w wN_{r_w}\big)\big]$ with independent $N_{r_w} \sim \mathrm{Poisson}(r_w)$. The Poisson identity $\frac{d}{d\lambda}\mathbb{E}[\phi(N_\lambda)] = \mathbb{E}[\phi(N_\lambda + 1) - \phi(N_\lambda)]$, valid for every ϕ of linear growth, gives for $Y \sim \mathrm{CP}(r)$ that

$$\frac{\partial^2 F}{\partial r_w\,\partial r_{w'}}(r) = \mathbb{E}\big[\psi(Y + w + w') - \psi(Y + w) - \psi(Y + w') + \psi(Y)\big] \ge 0$$

by (4). Hence $g''(s) = \sum_{w,w'} \bar{\mu}(\{w\})\bar{\mu}(\{w'\})\,\partial^2_{ww'}F(s\bar{\mu}) \ge 0$, and $g(0) = \psi(0) = 0$. Convexity and $g(0) = 0$ give $g(s) \le s\,g(1)$ on $[0,1]$. For $s \ge 1$, write $\mathrm{CP}(s\bar{\mu}) = Y_1 + Y'$ with independent $Y_1 \sim \mathrm{CP}(\bar{\mu})$ and $Y' \sim \mathrm{CP}((s-1)\bar{\mu})$; since $\psi(y+b) \le \psi(y) + b$ for $b \ge 0$, $g(s) \le g(1) + \mathbb{E}[Y'] = g(1) + (s-1)$. □

B.3 The unit two-size bin

For $\eta \in [0,1]$, the *unit bin* with heavy fraction η is $Y^{(\eta)} := \mathrm{CP}(\eta\delta_1 + k(1-\eta)\delta_{1/k}) = N + \frac{1}{k}M$ with independent $N \sim \mathrm{Poisson}(\eta)$ and $M \sim \mathrm{Poisson}(k(1-\eta))$. Its expected overflow is

$$\widetilde{U}_k(\eta) := \mathbb{E}\big[\psi(Y^{(\eta)})\big] = e^{-\eta} \sum_{m=0}^{k-1} \Big(1 - \frac{m}{k}\Big) \frac{e^{-k(1-\eta)}\big(k(1-\eta)\big)^m}{m!}. \tag{6}$$

Lemma B.4 (The unit-bin overflow)**.** *For every integer $k \geq 1$ and $\eta \in [0,1]$:*

(a) $\widetilde{U}_k(\eta) = e^{-\eta}\ \mathbb{E}[(1 - M/k)_+]$*, which gives (6);*
(b) $\widetilde{U}_k(\eta) \leq 1/e$*, with equality at* $\eta = 1$*;*
(c) $\widetilde{U}_k(\eta) \leq \frac{1}{2}\sqrt{\eta + (1-\eta)/k}$*;*
(d) $\widetilde{U}_k(\eta) \leq e^{-\eta}\big(\eta + \frac{1}{2}\sqrt{(1-\eta)/k}\big)$*;*
(e) $\widetilde{U}_k(0) = \mathsf{Pr}[M = k] = e^{-k}k^k/k! \leq 1/\sqrt{2\pi k}$*.*

Consequently $\widetilde{U}_k \leq U_k$ *on* $[0,1]$*.*

Proof. (a) Since $\mathbb{E}[Y^{(\eta)}] = 1$, we have $\mathbb{E}[(Y^{(\eta)} - 1)_+] = \mathbb{E}[(1 - Y^{(\eta)})_+]$. If $N \geq 1$ then $Y^{(\eta)} \geq 1$ and the deficit is 0; hence $\mathbb{E}[(1 - Y^{(\eta)})_+] = \mathsf{Pr}[N = 0]\ \mathbb{E}[(1 - M/k)_+]$. (b) For every integer $m \geq 0$, $(1 - m/k)_+ \leq (1 - 1/k)^m$, by Bernoulli's inequality for $m \leq k$ and nonnegativity otherwise. The Poisson probability generating function gives $\mathbb{E}[(1-1/k)^M] = e^{-(1-\eta)}$, so $\widetilde{U}_k(\eta) \leq e^{-\eta}e^{-(1-\eta)} = 1/e$. At $\eta = 1$, $M = 0$ and $\widetilde{U}_k(1) = e^{-1}$. (c) For a mean-zero X, $\mathbb{E}[X_+] = \frac{1}{2}\,\mathbb{E}\,|X| \leq \frac{1}{2}\sqrt{\mathbb{E}[X^2]}$. Apply it to $X = Y^{(\eta)} - 1$, whose variance is $\eta + (1-\eta)/k$. (d) Write $1 - M/k = \eta + \big((1-\eta) - M/k\big)$, so $(1 - M/k)_+ \leq \eta + \big((1-\eta) - M/k\big)_+$. The second term has mean-zero argument with variance $(1-\eta)/k$, so its expectation is at most $\frac{1}{2}\sqrt{(1-\eta)/k}$ as in (c). Multiply by $e^{-\eta}$ and use (a). (e) For $M \sim \mathrm{Poisson}(k)$, using $m\,\mathsf{Pr}[M = m] = k\,\mathsf{Pr}[M = m-1]$, $\mathbb{E}[(k - M)_+] = k\,\mathsf{Pr}[M \leq k-1] - k\,\mathsf{Pr}[M \leq k-2] = k\,\mathsf{Pr}[M = k-1] = k\,\mathsf{Pr}[M = k]$. Divide by k and use Stirling's bound $k! \geq \sqrt{2\pi k}\,(k/e)^k$. □

Claim B.5 (Concavity of the closed-form bounds)**.** The functions $B_1(\eta) = \frac{1}{2}\sqrt{\eta + \tau(1-\eta)}$ and $B_2(\eta) = e^{-\eta}\big(\eta + \frac{1}{2}\sqrt{\tau(1-\eta)}\big)$ are concave on $[0,1]$ for every $\tau \in (0,1]$. Consequently $U_k = \min\{1/e, B_1, B_2\}$ (with $\tau = 1/k$) is concave.

Proof. The concavity of B_1 is immediate because it is the square root of an affine function. For B_2, write $q := \sqrt{\tau}$ and $c := q/2 \leq 1/2$, so $B_2(x) = e^{-x}\big(x + c\sqrt{1-x}\big)$. For $x \in [0,1)$, set $u := \sqrt{1-x}$. A direct calculation gives

$$B_2''(x) = \frac{e^{-x}}{4(1-x)^2}\left[c(4u^5 + 4u^3 - u) - 4u^4(1+u^2)\right].$$

We show that the bracketed term is nonpositive. If $4u^5 + 4u^3 - u \leq 0$, this is immediate. Otherwise, using $c \leq 1/2$, it is enough to show $\frac{1}{2}(4u^5 + 4u^3 - u) \leq 4u^4(1+u^2)$, or equivalently

$$8u^5 - 4u^4 + 8u^3 - 4u^2 + 1 \geq 0.$$

If $u \geq 1/2$, this is immediate since $8u^5 - 4u^4 + 8u^3 - 4u^2 = 4u^4(2u-1) + 4u^2(2u-1) \geq 0$. If $0 \leq u \leq 1/2$, then $8u^5 - 4u^4 + 8u^3 - 4u^2 + 1 = 1 - 4u^2(1+u^2)(1-2u)$. The function $u^2(1-2u)$ is maximized on $[0,1/2]$ at $u = 1/3$, with value $1/27$, and $1 + u^2 \leq 5/4$ on this interval. Therefore $4u^2(1+u^2)(1-2u) \leq 4 \cdot \frac{5}{4} \cdot \frac{1}{27} = \frac{5}{27} < 1$. Hence the expression is nonnegative in all cases. Thus $B_2''(x) \leq 0$ on $[0,1)$, and B_2 is concave on $[0,1]$. Finally, a minimum of concave functions is concave. □

The exact overflow $\widetilde{U}_k$ is not concave in general (for larger k it is slightly convex near $\eta = 0$). We let $\widehat{U}_k$ denote its least concave majorant on $[0,1]$. Since U_k is concave and dominates $\widetilde{U}_k$, we have $\widetilde{U}_k \leq \widehat{U}_k \leq U_k$; numerically, $\widehat{U}_k - \widetilde{U}_k \leq 0.005$ for $k \leq 30$.

B.4 The per-resource bound and the proof of Theorem 3.4

Theorem B.6 (Per-resource guarantee)**.** *Let $x \geq 0$ satisfy the type constraints $\sum_i x_{ij} \leq 1$ (not necessarily the capacity constraints), and let G_S be produced by $\Phi_{k,x}$. Then*

$$\mathbb{E}[|M(G_S)|] \;\geq\; \sum_{i \in V} \min\{\Lambda_i, 1\} \left(1 - \widetilde{U}_k(\eta_i)\right). \tag{7}$$

If x is feasible for the Expected Instance LP, then $\mathbb{E}[|M(G_S)|] \geq \big(1 - \widehat{U}_k(\eta)\big) Z(x) \geq \big(1 - U_k(\eta)\big) Z(x)$ with $\eta = Z_H(x)/Z(x)$. Moreover, all statements remain valid if k is replaced, in the definitions of H_i, η_i, η and $\widetilde{U}_k, \widehat{U}_k, U_k$, by any integer $k' \in [1, k]$.

Proof. Fix a resource with $\Lambda_i > 0$. By Lemma B.1, (5), and convexity of ψ, $\mathbb{E}[\psi(Y_i)] \leq \mathbb{E}[\psi(\mathrm{CP}(\Lambda_i \bar{\mu}_i))]$, and $\mathrm{CP}(\bar{\mu}_i) \overset{d}{=} Y^{(\eta_i)}$. By Lemma B.3, this is at most $\Lambda_i \widetilde{U}_k(\eta_i)$ if $\Lambda_i \leq 1$, and at most $\widetilde{U}_k(\eta_i) + \Lambda_i - 1$ otherwise; that is, at most $\min\{\Lambda_i, 1\}\, \widetilde{U}_k(\eta_i) + (\Lambda_i - 1)_+$. Substituting into (3),

$$\mathbb{E}[|M(G_S)|] \geq \sum_i \Big(\Lambda_i - (\Lambda_i - 1)_+ - \min\{\Lambda_i, 1\}\, \widetilde{U}_k(\eta_i) \Big) = \sum_i \min\{\Lambda_i, 1\} \big(1 - \widetilde{U}_k(\eta_i)\big).$$

If x is feasible, then $\min\{\Lambda_i, 1\} = \Lambda_i$, and by Jensen's inequality for the concave $\widehat{U}_k$ with weights $\Lambda_i / Z(x)$, $\sum_i \Lambda_i \widetilde{U}_k(\eta_i) \leq \sum_i \Lambda_i \widehat{U}_k(\eta_i) \leq Z(x)\, \widehat{U}_k\big(\sum_i \Lambda_i \eta_i / Z(x)\big) = Z(x)\, \widehat{U}_k(\eta)$, since $\sum_i \Lambda_i \eta_i = \sum_i H_i = Z_H(x)$. Finally $\widehat{U}_k \leq U_k$. For $k' \leq k$: by (V3), every sampled edge with $x_{ij} \leq 1/k'$ has $w_{ij} \leq \max\{x_{ij}, 1/k\} \leq 1/k'$, and every weight is at most 1. Hence the size rounding leading to (5) applies with threshold $1/k'$, and the rest of the argument is unchanged. □

Proof of Theorem 3.4 and Corollary 3.5(c). The theorem is the second statement of Theorem B.6, and concavity of U_k is Claim B.5. If $\eta = 0$ then $\eta_i = 0$ for all i, and (7) together with Lemma B.4(e) gives Corollary 3.5(c). □

C Implementation and Pseudocodes

C.1 Graph Construction

The pseudocode for the graph construction is given in Algorithm 1. We specifically construct balanced graphs where supply equals demand (equal number of requests and resources). We target this balanced regime because it is the point of matching market efficiency. Structural imbalances artificially simplify the allocation task; when one side of the market is vastly over-provisioned with choices, the performance of even naive algorithms degenerates to the same high baseline. Operating precisely where supply equals demand provides the necessary resolution to differentiate sparsification strategies.

Algorithm 1: Graph Construction at Time t for NYC Yellow Taxis Dataset

Input: Time interval t, Trip dataset $\mathcal{D}$, Set of all spatial zones $\mathcal{Z}$, Spatial adjacency matrix A
Output: Realized bipartite graph $G_t = (V_t \cup U_t, E_t)$

// 1. Time Windows & Data Aggregation
$\mathcal{D}_{\text{drop}} \leftarrow$ All drop-off events from $\mathcal{D}$ in window $[t - 5\text{m}, t)$;
$\mathcal{D}_{\text{pick}} \leftarrow$ All pick-up events from $\mathcal{D}$ in window $[t, t + 5\text{m})$;
$n \leftarrow |\mathcal{D}_{\text{drop}}|$; // Define total graph size based on offline supply
for *each zone* $z \in \mathcal{Z}$ **do**
  $p_{\text{car}}(z) \leftarrow \frac{\text{count}(z \text{ in } \mathcal{D}_{\text{drop}})}{n}$;
  $p_{\text{rider}}(z) \leftarrow \frac{\text{count}(z \text{ in } \mathcal{D}_{\text{pick}})}{|\mathcal{D}_{\text{pick}}|}$

// 2. Proportional Node Sampling
$V_t \leftarrow \emptyset$ (Offline Cars), $U_t \leftarrow \emptyset$ (Online Riders);
for $i = 1, \ldots, n$ **do**
  Sample zone $z_v \sim p_{\text{car}}$ (with replacement);
  Create car node v_i with location z_v, add to V_t;
  Sample zone $z_u \sim p_{\text{rider}}$ (with replacement);
  Create rider node u_i with location z_u, add to U_t;

// 3. Edge Creation via Spatial Adjacency
$E_t \leftarrow \emptyset$;
for *each* $v \in V_t$ *and* $u \in U_t$ **do**
  if $v.zone == u.zone$ ***or*** $A[v.zone, u.zone] == 1$ **then**
    $E_t \leftarrow E_t \cup \{(v, u)\}$;

return $G_t = (V_t \cup U_t, E_t)$;

C.2 Monte Carlo construction of spread selection weights.

Our theoretical guarantee depends on the geometry of the fractional solution used to guide local sparsification: solutions whose mass is spread across many light edges incur smaller collision penalties. In the experiments, we therefore use a simple Monte Carlo heuristic, described in Algorithm 2, to construct spread fractional weights. We repeatedly sample a full realization of the stochastic instance, compute an offline maximum matching, and average the resulting matched-edge incidences. Because the matching algorithm is deterministic, we randomly shuffle the ordering of vertices before each run; this acts as random tie-breaking among multiple optimal matchings and helps spread mass across interchangeable or nearly interchangeable resources.

Formally, let G denote a random realization of the full graph, and let $M(G)$ be an offline maximum matching selected using this randomized tie-breaking rule. Let $A_j(G)$ be the number of arrivals of type j in G, and let $C_{ij}(G)$ be the number of type-j arrivals matched to resource v_i in $M(G)$. The expected edge incidence is $y_{ij} := \mathbb{E}[C_{ij}(G)]$. Equivalently, since $\mathbb{E}[A_j(G)] = np_j$, the corresponding conditional LP variable is

$$x_{ij} := \frac{y_{ij}}{np_j} = \mathsf{Pr}[\text{a type-}j \text{ arrival is matched to } v_i],$$

where the probability is over the stochastic realization and the randomized tie-breaking. The vector x is feasible for the expected-instance LP: every realized matching uses each resource at most once and matches each arrival at most once, so the resource and type constraints hold after taking expectations. Moreover, $\sum_{i,j} np_j x_{ij} = \sum_{i,j} y_{ij} = \mathbb{E}[|M(G)|]$. Thus the Monte Carlo estimates approximate a feasible fractional solution of value $Z = \mathbb{E}[|M(G)|]$, and these estimates are used as the local selection weights. The finite-sample estimates are themselves exactly feasible for the empirical arrival frequencies:

Lemma C.1 (Feasibility of the Monte Carlo weights)**.** *Let A_j and C_{ij} be the counters of Algorithm 2 after M simulations, and for types with $A_j > 0$ let $\hat{p}_j := A_j/(Mn)$ and $\hat{x}_{ij} := C_{ij}/A_j$. Then $\hat{x}$ satisfies the type constraints, is feasible for the Expected Instance LP with frequencies $\hat{p}$, and its objective under $\hat{p}$ equals the average realized matching size $\frac{1}{M}\sum_m |M^{(m)}|$.*

Proof. Each type-j arrival is matched at most once, so $\sum_i C_{ij} \le A_j$ and $\sum_i \hat{x}_{ij} \le 1$. Each resource is matched at most once per simulation, so $\sum_j C_{ij} \le M$; since $n\hat{p}_j\hat{x}_{ij} = C_{ij}/M$, this gives $\sum_j n\hat{p}_j\hat{x}_{ij} \le 1$. Finally $\sum_{i,j} n\hat{p}_j\hat{x}_{ij} = \frac{1}{M}\sum_{i,j} C_{ij} = \frac{1}{M}\sum_m |M^{(m)}|$. □

Under the true frequencies p, Theorem B.6 applies to $\hat{x}$ directly, since it requires only the type constraints; estimation error enters only through overloaded resources ($\Lambda_i > 1$). The uniform fallback in Algorithm 2 is used only for types with $\sum_i C_{ij} = 0$, i.e., with $\hat{x}_{\cdot j} \equiv 0$; such requests contribute no weight to the fractional matching f, and the edges they forward can only increase $|M(G_S)|$. As the number of simulations grows, $\hat{x}$ converges almost surely to the expected-incidence solution, which is feasible under p, so the total excess load $\sum_i(\Lambda_i - 1)_+$ vanishes.

Algorithm 2: Monte Carlo Weight Estimation with Random Tie-Breaking

Input: Rider distribution D, supply V, simulation count M, graph size n
Output: Conditional selection weights x_{ij} for type-resource edges (u_j, v_i)

Initialize counters $C_{ij} \leftarrow 0$ for all valid type-resource pairs (u_j, v_i)
Initialize counters $A_j \leftarrow 0$ for all request types t_j `// number of arrivals of type j across simulations`
for $m = 1, \dots, M$ **do**
  Generate mock arrivals $U^{(m)}$ by sampling n riders from D
  Construct full bipartite graph $G^{(m)} = (U^{(m)} \cup V, E^{(m)})$
  `// Randomize vertex order to implement tie-breaking and promote spread`
  Randomly shuffle the order of vertices in $U^{(m)}$ and V
  $M^{(m)} \leftarrow \text{HopcroftKarp}(G^{(m)})$
  for *each arriving rider* u *of type* t_j *in* $U^{(m)}$ **do**
    $A_j \leftarrow A_j + 1$
    if *rider is matched (there is* $v_i \in \Gamma(t_j)$ *such that* $(u, v_i) \in M^{(m)}$*)* **then**
      $C_{ij} \leftarrow C_{ij} + 1$
for *each request type* t_j **do**
  for *each* $v_i \in \Gamma(t_j)$ **do**
    $x_{ij} \leftarrow (\sum_{i'} C_{i'j}) > 0 \; ? \; C_{ij}/A_j : 1/|\Gamma(t_j)|$ `// Implementation-only uniform fallback for unseen or seen and unmatched types`
return $\{x_{ij}\}$

Algorithm 3: VarOpt Local Sparsifier Construction (Online)

Input: Arriving rider u of type t_j, Budget k, Pre-computed weights $\{x_{ij}\}$
Output: Selected edges $S_u \subseteq \{(u, v) \mid v \in \Gamma(t_j)\}$

Identify all valid candidate edges $E_u = \{(u, v_i) : v_i \in \Gamma(t_j)\}$
for *each* $e = (u, v_i) \in E_u$ **do**
  $\tilde{x}_e \leftarrow x_{ij}$
`// VarOpt`$_k$ `weighted sampling by` $\tilde{x}_e$
$S_u \leftarrow \text{VarOpt}_k((e, \tilde{x}_e)_{e \in E_u})$ `// Apply the stream-based algorithm [11]`
`// The threshold` τ `such that` $\sum_{e \in E_u} \min(1, \tau \cdot \tilde{x}_e) = \min(k, |E_u^+|)$ `and exact inclusion probabilities` $\pi_e = \min(1, \tau \cdot \tilde{x}_e)$ `are realized by this procedure.`
return S_u

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction claim a local sparsification model and spread-dependent guarantees, which match Theorem 3.4 and Corollary 3.5, and empirical performance, which matches Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See the Limitations paragraph in Section 5 and the discussion of Table 2.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Assumptions are stated with each result, and proofs are in Section 3 and Appendices A–C.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 gives the setup and parameters, and Appendix C gives pseudocode for all components.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The NYC data is public (Section 4.1). We do not release code, but Appendix C gives full pseudocode.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: No training is involved; all parameters are given in Section 4 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 1 and Figure 1 report 95% confidence intervals.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4 reports the environment (2 vCPUs, 32 GB RAM); all runs are lightweight CPU computations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is foundational algorithmic work with no direct path to negative applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No data or models with risk of misuse are released.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The public NYC TLC trip records are cited with their URL (Section 4.1), and synthetic instances are credited to their sources.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets are released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The research does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the methods. They assisted with writing, coding, and proof write-ups, and all results were verified by the authors.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.