Optimizing Mixture Importance Sampling Via Online Learning: Algorithms and Applications

Tingwei Liu[1], Hong Xie[2], John C.S. Lui[3],
Department of Computer Science & Engineering, The Chinese University of Hong Kong, HKSAR[1,3]
College of Computer Science, Chongqing University, China[2]
Email: tingweiliu2013@gmail.com[1], xiehong2018@foxmail.com[2], cslui@cse.cuhk.edu.hk[3]

***Abstract*—Importance sampling (IS) is widely used in rare event simulation, but it is costly to deal with *many rare events* simultaneously. For example, a rare event can be the failure to provide the quality-of-service guarantee for a critical network flow. Since network providers often need to deal with many critical flows (i.e., rare events) simultaneously, if using IS, providers have to simulate each rare event with its customized importance distribution individually. To reduce such cost, we propose an efficient mixture importance distribution for multiple rare events, and formulate the mixture importance sampling optimization problem (MISOP) to select the optimal mixture. We first show that the "*search direction*" of mixture is computationally expensive to evaluate, making it challenging to locate the optimal mixture. We then formulate a "*zero learning cost*" online learning framework to estimate the "*search direction*", and learn the optimal mixture from simulation samples of events. We develop two multi-armed bandit online learning algorithms to: (1) Minimize the sum of estimation variances with a regret of $(\ln T)^2/T$; (2) Minimize the simulation cost with a regret of $\sqrt{\ln T/T}$, where T denotes the number of simulation samples. We demonstrate our method on a realistic network and show that it can reduce the cost measures (i.e., sum of estimation variances and simulation cost) by as high as 61.6% compared with the uniform mixture IS.**

I. Introduction

Rare events are events that occur rarely but have catastrophic impacts or consequences. In power transmission networks, network component damages caused by some sudden unforeseen events (e.g., excessive load demands, lighting, or floods) can result in large-scale blackouts [1]. In communication networks, network component failures caused by some unwanted events (e.g., equipment aging or power shortage) can significantly degrade the intended network service [2]. On the Internet, some unexpected node and link failures can result in undeliveries of some critical flows. To quantify such *rare threats*, one needs to evaluate their risks accurately. For large networks, this is often computationally expensive as shown below:

Example 1. *Consider a large-scale network and a rare threat $\mathcal{E}$, which corresponds to the failure to provide promised quality-of-service (QoS) guarantees for a critical flow. The rare threat is induced by a subset of potential causes, e.g., link and node failures, which are indexed by m, $m\in[M]$ and may occur rarely. Let $\boldsymbol{x}\in\{0,1\}^M$ denote an occurrence profile of all these causes, associating with a probability mass function $P(\boldsymbol{x})$. The $\mathcal{E}$ is represented by a set of profiles $\boldsymbol{x}$, which is often unknown and of a large cardinality, say $O(2^M)$. For a given $\boldsymbol{x}$, a black-box $\mathbf{1}_{\mathcal{E}}(\boldsymbol{x})$ can simulate the network to test the occurrence of $\mathcal{E}$ (i.e., $\boldsymbol{x}\in\mathcal{E}$), but have no functional description of $\mathcal{E}$. So as many as $O(2^M)$ enumerations are needed to evaluate $\mathcal{E}$'s occurrence probability.*

Monte Carlo (MC) sampling is a typical method to address the high computational cost problem illustrated in Example 1. It estimates the occurrence of a rare threat $\mathcal{E}$ via generating samples $\boldsymbol{x}$ from $P(\boldsymbol{x})$. However, to obtain accurate estimations, MC needs to simulate a large number of samples to capture sufficient occurrences of $\mathcal{E}$. Importance Sampling (IS) improves the estimation efficiency of MC through a customized importance distribution $Q(\boldsymbol{x})$ to "*boost*" the occurrence of $\mathcal{E}$. One limitation of IS is that it has to simulate each rare threat with its customized importance distribution individually. This leads to an excessively high simulation cost, especially when dealing with *a large set of rare threats* as shown below:

Example 2. *Consider N critical flows in Example 1. The rare threat $\mathcal{E}_n$ denotes the failure to provide promised QoS guarantees for flow n, where $n\in[N]$. Each $\mathcal{E}_n$ is associated with a customized importance distribution $Q_n(\boldsymbol{x})$, and a black-box $\mathbf{1}_{\mathcal{E}_n}(\boldsymbol{x})$ that tests whether $\boldsymbol{x}\in\mathcal{E}_n$. Suppose, using IS to estimate each $\mathcal{E}_n$ requires T samples from $Q_n(\boldsymbol{x})$. To estimate all N events, we need TN samples, which is expensive for a large network where even simulating a single sample can take hours.*

To reduce such simulation cost burden of IS, we consider the *mixture importance sampling (MIS)* with mixture parameter $\boldsymbol{w}$:

$$Q(\boldsymbol{x};\boldsymbol{w})=\sum_{n\in[N]} w_n Q_n(\boldsymbol{x}),$$

where $w_n\geq 0, n\in[N]$ and $\sum_{n\in[N]} w_n=1$. Through this, each sample $\boldsymbol{x}$ drawn from distribution $Q(\boldsymbol{x};\boldsymbol{w})$ can be used for "*all*" rare threats $\{\mathcal{E}_n\}_{n=1}^N$. We aim to answer two questions:

(1) *How to quantify the "simulation cost" for a mixture $\boldsymbol{w}$?*
(2) *How to locate the optimal mixture $\boldsymbol{w}^*$?*

To design proper simulation cost metrics for $\boldsymbol{w}$, one needs to consider the simulation cost resulted from $Q(\boldsymbol{x};\boldsymbol{w})$ for each $\mathcal{E}_n$. Such metrics (i.e., cost measures) are functions of $Q(\boldsymbol{x};\boldsymbol{w})$. To search $\boldsymbol{w}^*$ minimizing the simulation cost, one needs to evaluate the search direction by marginalizing $\boldsymbol{x}$ in the metric with a sample space of size 2^M. To solve this challenge, we formulate a multi-armed bandit (MAB) online learning (OL) framework to estimate the "*search direction*" and learn $\boldsymbol{w}^*$ from simulation samples $\boldsymbol{x}$ drawn from $\{Q_n(\boldsymbol{x})\}_{n=1}^N$. One may use the classical stochastic optimization (SO) method to derive $\boldsymbol{w}^*$, but it has a significantly larger sample cost than our framework. To guarantee the convergence speed, SO needs sufficient samples from $Q(\boldsymbol{x};\boldsymbol{w}^{(t)})$ to locate an efficient "*search direction*" in each round

t, and $\boldsymbol{w}^{(t)}$ is the estimated mixture in round t. Our framework only needs a single sample $\boldsymbol{x}$ from one of $\{Q_n(\boldsymbol{x})\}_{n=1}^N$ in each round. This makes it challenging to estimate the "*search direction*" as well as learn $\boldsymbol{w}^*$. Our contributions are:

- We formulate *two metrics* to quantify the simulation cost for a mixture strategy and propose *a mixture importance sampling optimization problem (MISOP)* to select the optimal mixture. We first show the search direction of mixture is costly to evaluate, making it challenging to locate the optimal.
- We then formulate *a MAB OL framework* to estimate the search direction and learn the optimal mixture from "simulation samples". So instead of using sufficient simulation samples from $Q(\boldsymbol{x};\boldsymbol{w}^{(t)})$, our OL framework reduces the simulation cost by generating only a single simulation sample $\boldsymbol{x}$ from one of $\{Q_n(\boldsymbol{x})\}_{n=1}^N$ in each round of learning. Hence, achieving a zero cost on extra samples.
- We develop *two MAB learning algorithms* to efficiently learn the optimal mixture $\boldsymbol{w}^*$ under different cost measures, i.e.: (1) SumVar, to minimize the sum of variances with a regret of $(\ln T)^2/T$, and (2) SimCos, to minimize the simulation cost with a regret of $\sqrt{\ln T/T}$, where T is the number of samples. For each algorithm, we provide: (1) convexity and smoothness analysis; (2) algorithm to estimate the search direction of $\boldsymbol{w}$ with zero cost on extra samples, as well as provable concentration; (3) regret analysis and reveal the impact of key factors, e.g., similarity of $\{Q_n(\boldsymbol{x})\}_{n=1}^N$, on the regret.
- We demonstrate the efficiency of our methods on a realistic network. And our SumVar and SimCos algorithms reduce the associated cost measure by 37.8% and 61.6% respectively, compared with the uniform mixture IS.

II. Problem Formulation

We first introduce the MIS model with a mixture parameter $\boldsymbol{w}$. We then formulate an optimization framework that selects $\boldsymbol{w}$ to minimize a general cost measure. To address the computational challenge in locating the optimal mixture $\boldsymbol{w}^*$, we formulate an OL framework to estimate $\boldsymbol{w}^*$. Finally, we present two important instances of the OL framework with specific cost measures.

A. Mixture Importance Sampling

Consider N rare events, and we aim to estimate the occurrence probability for each individual event. Each event is induced by a subset of M potential *causes* denoted by $[M]$. Let $\Omega \triangleq \{0,1\}^M$. We denote $\boldsymbol{x}=(x_1,...,x_M)\in\Omega$ as the *occurrence profile* of all M causes, where x_m indicates whether the cause m occurs (1: yes, 0: no). Let $\boldsymbol{x}$ occur with a probability $P(\boldsymbol{x})\in[0,1]$, where $\sum_{\boldsymbol{x}\in\Omega} P(\boldsymbol{x})=1$. We formally denote the event $n\in[N]$ as $\mathcal{E}_n\subset\Omega$, of which the occurrence is indicated by a *membership oracle*:

$$\mathbf{1}_{\mathcal{E}_n}(\boldsymbol{x}) \triangleq \begin{cases} 1, & \text{if } \boldsymbol{x}\in\mathcal{E}_n, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Given a sample $\boldsymbol{x}$, $\mathbf{1}_{\mathcal{E}_n}(\boldsymbol{x})$ can run a simulation with causes indicated by $\boldsymbol{x}$ to test whether $\mathcal{E}$ occurs, i.e., $\boldsymbol{x}\in\mathcal{E}$. Yet, $\mathbf{1}_{\mathcal{E}_n}$ has no other functional description of $\mathcal{E}_n$. And the occurrence probability is denoted by:

$$\mu_n=\mathbb{P}_{\boldsymbol{x}\sim P}[\mathbf{1}_{\mathcal{E}_n}(\boldsymbol{x})=1]=\textstyle\sum_{\boldsymbol{x}\in\mathcal{E}_n} P(\boldsymbol{x}). \tag{2}$$

In many real-life applications, the exact value of μ_n is computationally expensive to evaluate due to a large cardinality of $\mathcal{E}_n$. For instance, consider an Internet-scale network with M physical links, where the m-th link fails with a probability of p_m. There are N competing flows, of which the undelivery of the n-th flow is represented by $\mathcal{E}_n\subset\Omega$. For each $\boldsymbol{x}\in\Omega$, we have:

$$P(\boldsymbol{x})=\textstyle\prod_{m\in[M]} p_m^{x_m}(1-p_m)^{1-x_m}. \tag{3}$$

Due to the high complexity of traffic engineering, $\mathcal{E}_n$ is usually unknown and with a large cardinality, resulting in a computational complexity of $O(2^M)$ to evaluate the exact value of μ_n.

The rare occurrence of $\mathcal{E}_n$ makes it costly to estimate μ_n via simulating $\boldsymbol{x}$ with $P(\boldsymbol{x})$, i.e., the classical MC method. One typical method to address this challenge is the IS method [3], [4]. Assume each $\mathcal{E}_n$ corresponds to a *customised* pure importance distribution $Q_n(\boldsymbol{x})$. IS provides an efficient estimation of μ_n if taking $Q_n(\boldsymbol{x})$ to simulate $\boldsymbol{x}$, but $Q_n(\boldsymbol{x})$ may not work for other events. The *one-run variance* for estimating μ_n with $Q_n(\boldsymbol{x})$ to simulate $\boldsymbol{x}$, which determines the simulation cost, is:

$$\mathbb{V}_{\boldsymbol{x}\sim Q_n}\Big[\mathbf{1}_{\mathcal{E}_n}(\boldsymbol{x})\tfrac{P(\boldsymbol{x})}{Q_n(\boldsymbol{x})}\Big] \triangleq \mathbb{E}_{\boldsymbol{x}\sim Q_n}\Big[\mathbf{1}_{\mathcal{E}_n}(\boldsymbol{x})\tfrac{P^2(\boldsymbol{x})}{Q_n^2(\boldsymbol{x})}\Big]-\mu_n^2. \tag{4}$$

Here $\{Q_n(\boldsymbol{x})\}_{n=1}^N$ can be obtained using IS or Sequential IS methods proposed in [5].

Yet, given a limited simulation budget and a large N, one usually could not afford to estimate each μ_n individually with the corresponding $Q_n(\boldsymbol{x})$. What one needs is an efficient sampling distribution working for *multiple* interested events simultaneously. Assume we take a mixture distribution of $\{Q_n(\boldsymbol{x})\}_{n=1}^N$. Formally, we have:

$$Q(\boldsymbol{x};\boldsymbol{w})\triangleq \textstyle\sum_{n\in[N]} w_n Q_n(\boldsymbol{x}), \tag{5}$$

where $\boldsymbol{w}\triangleq(w_1,\ldots,w_N)$, $w_n\geq 0$ and $\sum_{n\in[N]} w_n=1$. For the ease of presentation, denote the set of all possible choices of $\boldsymbol{w}$ as the probability simplex $\Delta\triangleq\{\boldsymbol{w}|w_n\geq 0, \sum_{n=1}^N w_n=1\}$.

Here we define the "*ξ-similarity*" as a metric to quantify how well the occurrences of interested events $\{\mathcal{E}_n\}_{n=1}^N$ can be efficiently estimated together by the following definition:

Definition 1 (**ξ-similarity**). *Events $\{\mathcal{E}_n\}_{n=1}^N$ are ξ-similar if their corresponding pure importance distributions $\{Q_n(\boldsymbol{x})\}_{n=1}^N$ satisfy: for $\xi\in[0,\infty]$, $\forall\boldsymbol{x}\in\Omega, \forall n,n'\in[N]$, $\frac{1}{\xi}\leq\frac{Q_n(\boldsymbol{x})}{Q_{n'}(\boldsymbol{x})}\leq\xi$.*

To illustrate, consider $\{Q_n(\boldsymbol{x})\}_{n=1}^N$ have different (or even disjoint) supports, then $\xi=\infty$. Figure 1 shows more examples with different levels of ξ-similarities.

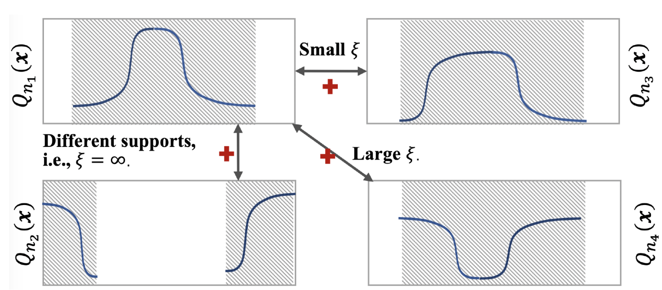


Fig. 1: Examples of different levels of ξ-similarities: an infinite ξ of $\{\mathcal{E}_{n_1},\mathcal{E}_{n_2}\}$ implies that even the optimal mixture $Q(\boldsymbol{x};\boldsymbol{w}^*)$ would not work for both $\mathcal{E}_{n_1}$ and $\mathcal{E}_{n_2}$; a large ξ for $\{\mathcal{E}_{n_1},\mathcal{E}_{n_4}\}$ implies a slow convergence to $Q(\boldsymbol{x};\boldsymbol{w}^*)$; a small ξ for $\{\mathcal{E}_{n_1},\mathcal{E}_{n_3}\}$ implies a fast convergence to $Q(\boldsymbol{x};\boldsymbol{w}^*)$.

B. General Optimization & Learning Framework

Given $Q(\boldsymbol{x};\boldsymbol{w})$ to simulate $\boldsymbol{x}$, the one-run variance of $\mathcal{E}_n$ is:

$$\sigma_n^2(\boldsymbol{w}) \triangleq \mathbb{V}_{\boldsymbol{x}\sim Q}\left[\mathbf{1}_{\mathcal{E}_n}(\boldsymbol{x})\frac{P(\boldsymbol{x})}{Q(\boldsymbol{x};\boldsymbol{w})}\right]. \quad (6)$$

One can evaluate the overall simulation efficiency associated with the mixture parameter $\boldsymbol{w}$ by the cost measure $L(\boldsymbol{\sigma}(\boldsymbol{w}))\in\mathbb{R}$ where $\boldsymbol{\sigma}(\boldsymbol{w})\triangleq(\sigma_1(\boldsymbol{w})\sigma_2(\boldsymbol{w}),...,\sigma_N(\boldsymbol{w}))$ (Refer to Section II-C for some examples). We formulate the following mixture importance sampling optimization problem.

Problem 1 (Mixture Importance Sampling Optimization (MISOP)). *Given M causes, associated with a natural occurrence distribution $P(\boldsymbol{x})$; N interested events, associated with efficient pure importance distributions $\{Q_n(\boldsymbol{x})\}_{n=1}^N$; and the cost measure $L(\boldsymbol{\sigma}(\boldsymbol{w}))$. Select $\boldsymbol{w}$ to minimize the cost:*

$$\min_{\boldsymbol{w}\in\Delta} L(\boldsymbol{\sigma}(\boldsymbol{w})). \quad (7)$$

In general, Problem 1 is a non-linear optimization problem. One challenge in solving Problem 1 is both $L(\boldsymbol{\sigma}(\boldsymbol{w}))$ and $\nabla L(\boldsymbol{\sigma}(\boldsymbol{w}))$ are computationally expensive to compute, i.e., the exact computational complexities are $O(2^M)$ for the large state space of $\boldsymbol{x}$. To overcome this challenge, we use a MAB framework to estimate (or online learn) $\boldsymbol{w}^*$ from simulation samples.

Problem 2 (Mixture Importance Sampling Learning (MIS-Learning)). *Given M causes, N interested events and the number of rounds (or data samples) $T\in\mathbb{N}_+$. At round $t=1,...,T$:*

- *Select an arm (or event) $I_t\in[N]$ based on an algorithm $\mathcal{A}$ and the sample history $\{(I_s,\boldsymbol{x}^{(s)})\}_{s=1}^{t-1}$;*
- *Draw a simulation sample of profile $\boldsymbol{x}^{(t)}$ from $Q_{I_t}(\boldsymbol{x})$;*
- *Update the proportions of selecting arms (or events) which denoted by $\boldsymbol{w}^{(t)}=(w_1^{(t)},...,w_N^{(t)})$, where $\boldsymbol{w}^{(t)}=\frac{1}{t}\sum_{s\in[t]}\boldsymbol{e}_{I_s}$;*

Objective: Design an algorithm $\mathcal{A}$ to achieve a low and sublinear regret, where the regret is defined as

$$R_T \triangleq L(\boldsymbol{\sigma}(\boldsymbol{w}^{(T)})) - \min_{\boldsymbol{w}\in\Delta} L(\boldsymbol{\sigma}(\boldsymbol{w})). \quad (8)$$

In Problem 2, each arm (or event) indexed by n corresponds to a customized pure distribution $Q_n(\boldsymbol{x})$, and a general cost function $L(\boldsymbol{\sigma}(\boldsymbol{w}^{(T)}))$ is considered. In the following, we consider two important instances of $L(\boldsymbol{\sigma}(\boldsymbol{w}^{(T)}))$.

C. Two Instances of the MIS Learning Problem

Given $Q(\boldsymbol{x};\boldsymbol{w})$ to simulate $\boldsymbol{x}$, let $\ell_n(\boldsymbol{w})$ measure the simulation cost to achieve the desired estimation accuracy for μ_n, i.e., the confidence interval (CI) is bounded by a threshold δ_n. Also, let $\ell_{max}(\boldsymbol{w})$ measure the simulation cost to achieve desired estimation accuracies for all $\{\mu_n\}_{n=1}^N$. Then:

$$\ell_n(\boldsymbol{w}) \triangleq \frac{\sigma_n^2(\boldsymbol{w})}{\delta_n^2} \quad \text{and} \quad \ell_{max}(\boldsymbol{w}) \triangleq \max_{n\in[N]} \ell_n(\boldsymbol{w}). \quad (9)$$

Next, we consider the cost measure $L(\boldsymbol{\sigma}(\boldsymbol{w}))$ with various accuracy requirements $\{\delta_n\}_{n=1}^N$, and introduce the corresponding MIS-Learning problems.

MIS-Learning to Minimize the Sum of Variances: We start with the simplest case where we assume homogeneous accuracy requirements (i.e., $\{\delta_n\}_{n=1}^N$ are equal) and consider bounding $\sum_{n\in[N]}\ell_n(\boldsymbol{w})$ in order to bound $\ell_{max}(\boldsymbol{w})$. Then:

$$\min_{\boldsymbol{w}\in\Delta}\sum_{n\in[N]}\ell_n(\boldsymbol{w}) \iff \min_{\boldsymbol{w}\in\Delta}\sum_{n\in[N]}\sigma_n^2(\boldsymbol{w}) \iff \min_{\boldsymbol{w}\in\Delta}\sum_{n\in[N]}\sigma_n^2(\boldsymbol{w})+\mu_n^2. \quad (10)$$

We can define the total loss (i.e., cost measure) in terms of the sum of one-run variances as follows:

$$L(\boldsymbol{\sigma}(\boldsymbol{w})) = \sum_{n\in[N]}\sigma_n^2(\boldsymbol{w})+\mu_n^2 \triangleq L_{\text{SumVar}}(\boldsymbol{w}), \quad (11)$$

and name the MIS-Learning with cost measure in Eq. (11) as minimizing the sum of variances (SumVar) MIS-Learning.

MIS-Learning to Minimize the Simulation Cost: Consider $\{\mathcal{E}_n\}_{n=1}^N$ with heterogenous accuracy requirements. Specifically, assume each $\mathcal{E}_n$ has a predefined occurrence probability threshold o_n, e.g., $\mathcal{E}_n$ represents the undelivery of a specific flow and we want to *accurately state whether the undelivery probability* $\mu_n\le o_n$. Then the CI width should not exceed $\delta_n=|\mu_n-o_n|$ and:

$$\min_{\boldsymbol{w}\in\Delta}\ell_{max}(\boldsymbol{w}) \iff \min_{\boldsymbol{w}\in\Delta}\max_{n\in[N]}\frac{\sigma_n^2(\boldsymbol{w})}{(\mu_n-o_n)^2}. \quad (12)$$

We define the total loss in terms of the simulation cost to achieve all desired estimation accuracies as:

$$L(\boldsymbol{\sigma}(\boldsymbol{w})) = \max_{n\in[N]}\frac{\sigma_n^2(\boldsymbol{w})}{(\mu_n-o_n)^2} \triangleq L_{\text{SimCos}}(\boldsymbol{w}), \quad (13)$$

and name the MIS-Learning with cost measure in Eq. (13) as minimizing the simulation cost (SimCos) MIS-Learning.

III. Learning to Minimize the Sum of Variances

In this section, we first present the design of our SumVar algorithm, which learns the optimal mixture $\boldsymbol{w}^*$ to minimize the sum of variances in an online manner. Then, we present the regret upper bound of our algorithm, and reveal the impact of the ξ-similarity on the learning speed. Finally, we present the fundamental idea of our proof for our regret upper bound.

A. The Design of SumVar Algorithm

The key idea of our SumVar algorithm is that at each round of learning: (1) *First estimate the gradient $\nabla L_{\text{SumVar}}(\boldsymbol{w})$ via historical data samples*; (2) *Then select the arm (or event) based on the estimated gradient.*

Gradient estimation. Consider at round t, we aim to estimate $\nabla L_{\text{SumVar}}(\boldsymbol{w}^{(t-1)})$ from historical data samples. Let us first derive $\nabla L_{\text{SumVar}}(\boldsymbol{w})$ as:

$$\begin{aligned}\nabla L_{\text{SumVar}}(\boldsymbol{w}) &= \nabla\Big\{\sum_{n\in[N]}\mathbb{E}_{\boldsymbol{x}\sim Q(\boldsymbol{x};\boldsymbol{w})}\Big[\frac{P^2(\boldsymbol{x})\mathbf{1}_{\mathcal{E}_n}(\boldsymbol{x})}{Q^2(\boldsymbol{x};\boldsymbol{w})}\Big]\Big\}\\ &= -\sum_{n\in[N]}\sum_{\boldsymbol{x}\in\Omega}\Big[\frac{P^2(\boldsymbol{x})\mathbf{1}_{\mathcal{E}_n}(\boldsymbol{x})}{Q^2(\boldsymbol{x};\boldsymbol{w})}\Big](Q_1(\boldsymbol{x}),...,Q_N(\boldsymbol{x}))\\ &= \mathbb{E}_{\boldsymbol{x}\sim Q(\boldsymbol{x};\boldsymbol{w})}[(-Z_1(\boldsymbol{x}),\ldots,-Z_N(\boldsymbol{x}))], \end{aligned} \quad (14)$$

where $Z_n(\boldsymbol{x})\triangleq\frac{P^2(\boldsymbol{x})\sum_{i\in[N]}\mathbf{1}_{\mathcal{E}_i}(\boldsymbol{x})}{Q^3(\boldsymbol{x};\boldsymbol{w}^{(t-1)})}Q_n(\boldsymbol{x}),\forall n\in[N]$. If historical data samples $\{\boldsymbol{x}^{(s)}\}_{s=1}^{t-1}$ were IID samples of $\boldsymbol{x}\sim Q\big(\boldsymbol{x};\boldsymbol{w}^{(t-1)}\big)$, then the gradient $\nabla L_{\text{SumVar}}(\boldsymbol{w}^{(t-1)})$ can be estimated by $\boldsymbol{g}^{(t)}$, where:

$$g_n^{(t)} = \frac{-1}{t-1}\sum_{s\in[t-1]}Z_n(\boldsymbol{x}^{(s)}),\forall n\in[N]. \quad (15)$$

Nevertheless, the challenge is that $\{\boldsymbol{x}^{(s)}\}_{s=1}^{t-1}$ are generated from $\boldsymbol{x}^{(s)}\sim Q_{I_s}(\boldsymbol{x})$. To address this challenge, the following theorem proves that Eq. (15) is asymptotically accurate in estimating the gradient $\nabla L_{\text{SumVar}}(\boldsymbol{w}^{(t-1)})$.

Theorem 1. *Consider the MIS-Learning framework, where at round t, $t\in[T]$ take the I_t-th distribution $Q_{I_t}(\boldsymbol{x})$ to generate $\boldsymbol{x}^{(t)}$. Then, $\lim_{t\to\infty}\|\boldsymbol{g}^{(t)}-\nabla L_{\text{SumVar}}(\boldsymbol{w}^{(t-1)})\|=0$.*

Remark: Such asymptotic property owns much to the role of mixture parameter $\boldsymbol{w}^{(t)}$, i.e., the observed proportions of selecting the distribution $Q_{I_t}(\boldsymbol{x})$ till round t. Hence, after sufficient t rounds of MIS-Learning, all samples $\{\boldsymbol{x}^{(s)}\}_{s=1}^t$ can be approximately considered as simulated by $Q(\boldsymbol{x};\boldsymbol{w}^{(t)})$.

Algorithm 1 SumVar MIS-Learning

Input: N, $\boldsymbol{w}{=}(\frac{1}{N},...,\frac{1}{N})$, $c_n^{(t)}, \forall n{\in}[N], t{=}1,...,T$
for all $t{\le}N$ **do**
  Draw $\boldsymbol{x}^{(t)}$ according to distribution $Q_t(\boldsymbol{x})$, and then record history: $Q_t(\boldsymbol{x}^{(t)})$ and $\mathbf{1}_{\mathcal{E}_t}(\boldsymbol{x}^{(t)})$.
for all $t{>}N$ **do**
  Estimate the gradient $\nabla L_{\text{SumVar}}(\boldsymbol{w}^{(t-1)})$ using $\boldsymbol{g}^{(t)}$ in Eq. (15).
  Compute the LCB $\underline{\boldsymbol{g}}^{(t)}$, where $\underline{g}_n^{(t)}{=}g_n^{(t)}{-}c_n^{(t)}$.
  Select $I_t{\in}\operatorname{argmin}_{n\in[N]} \underline{g}_n^{(t)}$ and draw $\boldsymbol{x}^{(t)}$ from $Q_{I_t}(\boldsymbol{x})$.
  Record history: $Q_{I_t}(\boldsymbol{x}^{(t)})$ and $\mathbf{1}_{\mathcal{E}_{I_t}}(\boldsymbol{x}^{(t)})$.
  Update $\boldsymbol{w}^{(t)}{\leftarrow}\boldsymbol{w}^{(t-1)}{+}\frac{1}{t}(\boldsymbol{e}_{I_t}{-}\boldsymbol{w}^{(t-1)})$.

Arm selection. We outline the arm selection in Algorithm 1. From [6], we know that finding the minimizer of lower bound confidence $\min_{n\in[N]} \underline{g}_n^{(t)}$ is equivalent to making a step of size $\frac{1}{t+1}$ in the direction of corner of simplex Δ that $\min_{\boldsymbol{z}\in\Delta} \boldsymbol{z}^\top \underline{g}^{(t)}$, which is precisely the Frank-Wolfe algorithm [7]. Hence, we apply the LCB Frank-Wolfe algorithm to select the arm based on the estimated gradient in Eq. (15). Note that in Algorithm 1, one can select $c_n^{(t)}$ to control the exploration and exploitation tradeoffs[1]. Selecting the $c_n^{(t)}$ is closely related to the regret of Algorithm 1. We thus delay the selection in the next subsection, where we analyze the regret bound.

B. Regret Analysis of SumVar Algorithm

We first establish two building blocks for the regret analysis of Algorithm 1: (1) *The strong convexity and smoothness properties of* $L_{\text{SumVar}}(\boldsymbol{w})$, and (2) *The concentration property of* $\boldsymbol{g}^{(t)}$ *in estimating* $\nabla L_{\text{SumVar}}(\boldsymbol{w}^{(t-1)})$. Then we apply these two building blocks to derive the regret upper bound of Algorithm 1.

Strong convexity and smoothness of $L_{\text{SumVar}}(\boldsymbol{w})$. Let us first formally define the strong convexity and smoothness.

Definition 2 (Strong convexity and smoothness)**.** *Let X be a convex set in the vector space and f: $X{\to}\mathbb{R}$ be a function. f is called α-strongly convex if and only if $\forall \boldsymbol{x}{\in}X, \nabla^2 f(\boldsymbol{x}){\succeq}\alpha I$, or equivalently,*

$$\forall \boldsymbol{x},\boldsymbol{y}{\in}X, \;\; f(\boldsymbol{y}){\ge}f(\boldsymbol{x}){+}\nabla^\top f(\boldsymbol{x})(\boldsymbol{y}{-}\boldsymbol{x}){+}\tfrac{\alpha}{2}\|\boldsymbol{y}{-}\boldsymbol{x}\|_2^2. \quad (16)$$

Similarly, f is β-smooth if and only if $\forall \boldsymbol{x}{\in}X, \nabla^2 f(\boldsymbol{x}){\preceq}\beta I$, or equivalently,

$$\forall \boldsymbol{x},\boldsymbol{y}{\in}X, \;\; f(\boldsymbol{y}){\le}f(\boldsymbol{x}){+}\nabla^\top f(\boldsymbol{x})(\boldsymbol{y}{-}\boldsymbol{x}){+}\tfrac{\beta}{2}\|\boldsymbol{y}{-}\boldsymbol{x}\|_2^2. \quad (17)$$

In the following theorem, we prove the strong convexity and smoothness of $L_{\text{SumVar}}(\boldsymbol{w})$.

Theorem 2. *If $\{\mathcal{E}_n\}_{n=1}^N$ has a ξ-similarity, the $L_{\text{SumVar}}(\boldsymbol{w})$ given by Eq. (11) is α-strongly convex and β-smooth with:*

$$\alpha{=}\tfrac{2}{N\xi^2}(\textstyle\sum_{n\in[N]}\mu_n)^2 \text{ and } \beta{=}2\xi^3\textstyle\sum_{n\in[N]}\mu_n. \quad (18)$$

Remark: Theorem 2 quantifies the impact of ξ-similarities on $L_{\text{SumVar}}(\boldsymbol{w})$. In particular, the strong convexity of $L_{\text{SumVar}}(\boldsymbol{w})$ vanishes and $L_{\text{SumVar}}(\boldsymbol{w})$ becomes nonsmooth when $\xi{\to}\infty$, i.e., the event occurrences are not similar. This implies that the ξ-similarity is essential for learning the optimal mixture $\boldsymbol{w}^*$ as well. Besides, in case that $\xi{=}\infty$, one can divide $\{\mathcal{E}_n(x)\}_{n=1}^N$ into multiple sub-groups such that each sub-group has a finite ξ.

[1]In Algorithm 1, the derivation of c_n relies on Z_n and its moments, which are costly to compute exactly. In the implementation, we take the empirical estimation of Z_n and its moments. This will not affect our regret upper bound conclusion as the derivation utilizes the upper bounds of Z_n and its moments.

Concentration property of $\boldsymbol{g}^{(t)}$. We aim to characterize how well the estimator $g_n^{(t)}$ concentrates around $\nabla L_{\text{SumVar}}(\boldsymbol{w}^{(t)})\big|_n$. Such concentration is characterized by a balance between the confidence probability denoted by $\zeta^{(t)}{\in}[0,1]$ and the deviation denoted by $\epsilon_n^{(t)}$. One challenge is that in the estimator $g_n^{(t)}$ in Eq. (15), the historical data samples $\{\boldsymbol{x}^{(s)}\}_{s=1}^{t-1}$ are not IID. The following theorem resolves this challenge by quantifying the tradeoff between $\zeta^{(t)}$ and $\epsilon_n^{(t)}$.

Theorem 3. *Assume $\boldsymbol{x}{\sim}Q(\boldsymbol{x};\boldsymbol{w}^{(t-1)})$ for both $\mathbb{E}$ and $\mathbb{V}$. Suppose $\zeta^{(t)}$ and $\epsilon^{(t)}$ satisfy:*

$$\epsilon_n^{(t)}{=}\tfrac{1}{3t}\ln\tfrac{1}{\zeta^{(t)}}Z_n^{max}{+}\sqrt{\tfrac{1}{9t^2}(\ln\tfrac{1}{\zeta^{(t)}}Z_n^{max})^2{+}\tfrac{2}{t}\ln\tfrac{1}{\zeta^{(t)}}\mathbb{V}Z_n(\boldsymbol{x})}.$$

where $Z_n^{max}{\triangleq}\max_{\boldsymbol{x}\in\Omega}|Z_n(\boldsymbol{x}){-}\mathbb{E}[Z_n(\boldsymbol{y})]|$. Then, it holds that

$$\mathbb{P}\big[g_n^{(t)}{-}\nabla L_{\text{SumVar}}(\boldsymbol{w}^{(t)})\big|_n{\le}\epsilon_n^{(t)}\big]{\le}\zeta^{(t)}, \quad (19)$$

$$\mathbb{P}\big[g_n^{(t)}{-}\nabla L_{\text{SumVar}}(\boldsymbol{w}^{(t)})\big|_n{\ge}{-}\epsilon_n^{(t)}\big]{\le}\zeta^{(t)}. \quad (20)$$

Theorem 3 serves as a building block for one to vary $\zeta^{(t)}$ and $\epsilon^{(t)}$, to attain different confidence and variation tradeoffs. This confidence and variation tradeoff is essential to select the parameter $c_n^{(t)}$ of Algorithm 1 and analyze its regret later. We need to point out that, $Z_n^{max}{=}O(\xi^3)$ and $\mathbb{V}Z_n(\boldsymbol{x}){=}O(\xi^5)$, i.e, the CI width of $\boldsymbol{g}^{(t)}$ is proportional to ξ. This reveals the impact of ξ-similarity on the concentration of gradient estimation.

Regret upper bound. With the above two building blocks, we now select the parameter $c_n^{(t)}$ for Algorithm 1 and prove the regret upper bound. Due to page limit, we present a sketch proof in the next subsection.

Theorem 4 (Regret upper bound of SumVar algorithm)**.** *Suppose $\{\mathcal{E}_n\}_{n=1}^N$ has a "ξ-similarity". For MIS-Learning with cost measure $L_{\text{SumVar}}(\boldsymbol{w})$ in Eq. (11), after T steps of the SumVar algorithm, the choice of $c_n^{(t)}{=}\epsilon_n^{(t)}$ and*

$$\zeta^{(t)}{=}\begin{cases} T_0^{-2}, & \text{if } t{\le}T_0; \\ t^{-2}, & \text{if } t{>}T_0, \end{cases}$$

holds the following: when $\frac{\ln(1/\zeta^{(t)})}{t}{\ge}\frac{9\sum_{i\in[N]}\mu_i}{4N\xi}$,

$$\mathbb{E}_{\boldsymbol{x}\sim Q}[R_T]{\le}C_1\tfrac{1}{T}{+}C_2\tfrac{\ln T}{T}; \quad (21)$$

otherwise, we have:

$$\mathbb{E}_{\boldsymbol{x}\sim Q}[R_T]{\le}C_3\tfrac{1}{T}{+}C_4\tfrac{\operatorname{erf}\sqrt{\ln T/2}}{T}{+}C_5\tfrac{\ln T}{T}{+}C_6\tfrac{(\ln T)^2}{T}. \quad (22)$$

where, $C_1{=}O\Big(\frac{N^2(\ln T_0)^2\xi^6}{\alpha\eta^2}{+}\frac{N^{3/2}\xi^2\sum_{i\in[N]}\mu_i}{T_0}{+}\frac{N\ln T_0\beta\xi^3}{\alpha\eta^2}\Big)$,

$$C_3{=}O\Big(\frac{N^{3/2}\xi^2\sum_{i\in[N]}\mu_i}{T_0}{+}\frac{N(\ln T_0)^2\beta\xi^2}{\alpha\eta^2}\Big),$$

$$C_2{=}C_5{=}O(\beta), C_4{=}O\Big(\frac{\sqrt{N\xi^5}\beta}{\alpha\eta^2}\Big), C_6{=}O\Big(\frac{N\xi^5}{\alpha\eta^2}\Big).$$

Remark: Theorem 4 shows that the regret upper bound is proportional to the ξ-similarity. It also reveals that a small ξ implies a fast convergence to the optimal mixture.

C. Proof Sketch

Here we state the sketch proof for Theorem 4, in which we discuss the regret upper bound of the SumVar algorithm.

Let I_t stand for the index of arm selected by $\mathcal{A}$ at round t. Recall that $\boldsymbol{w}^{(t)}$ is defined as the proportions of arm selections, i.e., $\boldsymbol{w}^{(t)}{=}\frac{1}{t}\sum_{s\in[t]}\boldsymbol{e}_{I_s}$. We can derive the following recurrence:

$$\boldsymbol{w}^{(t+1)}{=}\frac{t\boldsymbol{w}^{(t)}{+}\boldsymbol{e}_{I_{t+1}}}{t+1}{=}\boldsymbol{w}^{(t)}{+}\frac{\boldsymbol{e}_{I_{t+1}}{-}\boldsymbol{w}^{(t)}}{t+1}. \quad (23)$$

Let $\boldsymbol{w}^*$ be the optimal mixture parameter:

$$\boldsymbol{w}^* = \operatorname{argmin}_{\boldsymbol{w}\in\Delta} L_{\mathrm{SumVar}}(\boldsymbol{w}), \tag{24}$$

and define $\boldsymbol{e}_{*_{t+1}}$ as the following minimizer:

$$\boldsymbol{e}_{*_{t+1}} = \operatorname{argmin}_{\boldsymbol{z}\in\Delta} \boldsymbol{z}^\top \nabla L_{\mathrm{SumVar}}(\boldsymbol{w}^{(t)}), \tag{25}$$

which is also the steepest descent direction of $L_{\mathrm{SumVar}}(\boldsymbol{w}^{(t)})$ *with respect to the standard basis*. Note that the $\boldsymbol{e}_{*_{t+1}}$ is our desired searching direction, and we estimate it with $\boldsymbol{e}_{I_{t+1}}$ based on historical observations. For convenience, denote

$$\varepsilon^{(t+1)} = \nabla L_{\mathrm{SumVar}}(\boldsymbol{w}^{(t)})^\top (\boldsymbol{e}_{I_{t+1}} - \boldsymbol{e}_{*_{t+1}}). \tag{26}$$

The proof of Theorem 4 can be broken down into five steps.

Step 1: By the strong convexity and smoothness of $L_{\mathrm{SumVar}}(\boldsymbol{w})$ in Theorem 2, we first partition the regret R_T and show that:

$$R_T = \frac{1}{T}\left[\sum_{s\in[T]} \frac{\beta}{s} + \sum_{s\in[T]} \varepsilon^{(s)}\right] \le \beta \frac{\ln T}{T} + \frac{\sum_{s\in[T]} \varepsilon^{(s)}}{T}. \tag{27}$$

Step 2: To bound R_T, it is equivalent to bound $\sum_{s\in[T]} \varepsilon^{(s)}$. We start by looking at $c_n^{(t)}$, i.e., the confidence bound when estimating $\nabla L_{\mathrm{SumVar}}(\boldsymbol{w}^{(t)})\big|_n$ with $g_n^{(t)}$, which affects the accuracy of estimating $\boldsymbol{e}_{*_t}$ with $\boldsymbol{e}_{I_t}$ when $n = I_t$. The following claim reveals the relationship between $c_{I_t}^{(t)}$ and $\varepsilon^{(t+1)}$:

Claim 1. *Assume $c_n^{(t)}$ satisfies*

$$\mathbb{P}\big[g_n^{(t)} - \nabla L_{\mathrm{SumVar}}(\boldsymbol{w}^{(t)})\big|_n \le c_n^{(t)}\big] \le \zeta^{(t)}, \tag{28}$$

$$\mathbb{P}\big[g_n^{(t)} - \nabla L_{\mathrm{SumVar}}(\boldsymbol{w}^{(t)})\big|_n \ge -c_n^{(t)}\big] \le \zeta^{(t)}. \tag{29}$$

Then with a probability at least $1 - 2\zeta^{(t)}$, $\epsilon^{(t+1)} \le 2c_{I_{t+1}}^{(t)}$.

We then derive the expression of $c_n^{(t)}$. By Theorem 3, we know that Eq. (28) and (29) can be satisfied with the choice of $c_n^{(t)} = \epsilon_n^{(t)}$ where $\epsilon_n^{(t)}$ is defined in Theorem 3.

Finally, we bound $c_n^{(t)}$ by the following claim:

Claim 2. *With the choice of $c_n^{(t)} = \epsilon_n^{(t)}$, we have:*

$$c_i^{(t)} \le \begin{cases} \frac{4}{3} N \xi^3 \frac{\ln(1/\zeta^{(t)})}{t}, & \text{if } \frac{\ln(1/\zeta^{(t)})}{t} \ge \frac{9\sum_{m\in[N]}\mu_m}{4N\xi}; \\ 2\sqrt{N\xi^5 \sum_{m\in[N]} \mu_m \frac{\ln(1/\zeta^{(t)})}{t}}, & \text{otherwise.} \end{cases}$$

where $\epsilon_n^{(t)}$ is defined in Theorem 3.

Combine Claim 1, 2 and Theorem 3, we can show that with the choice of $c_n^{(t)}$ in Theorem 3, the regret R_T converges at a rate of $O\big(\frac{1}{T}\sum_t c_{I_t}^{(t)}\big)$ and the bound of $c_{I_t}^{(t)}$ is given by Claim 2.

Step 3: Next we show that R_T can converge at a faster rate of $O\big(\frac{1}{T}\sum_t \{c_{I_t}^{(t)}\}^2\big)$ instead of $O\big(\frac{1}{T}\sum_t c_{I_t}^{(t)}\big)$.

Let η be the distance from $\boldsymbol{w}^*$ to $\partial\Delta$ and let $c^{(t)} \triangleq \max_{n\in[N]} c_n^{(t)}$. Claim 2 provides an upper bound of $c_n^{(t)}$. Next, we utilize $c^{(t)}$ to bound R_T by the following claim:

Claim 3. *Assume we select $\zeta^{(t)}$ properly such that:*

$$\frac{1}{t}\sum_{s\in[t]} \big(c^{(t)}\big)^2 \ge \big(c^{(t+1)}\big)^2. \tag{30}$$

Then with a probability at least $1 - N\sum_t \zeta^{(t)}$ that:

$$TR_T \le \frac{\alpha\eta^2}{2} + \frac{\pi^2\beta^2}{3\alpha\eta^2} + \beta \ln T + \frac{8\beta}{\alpha\eta^2}\sum_{t\in[T]} \frac{c^{(t)}}{t} + \frac{8}{\alpha\eta^2}\sum_{t\in[T]} \big(c^{(t)}\big)^2. \tag{31}$$

Step 4: We now discuss how to select $\zeta^{(t)}$ to guarantee Eq. (30), and give bounds of $\sum_t (c^{(t)})^2$ and $\sum_t \frac{c^{(t)}}{t}$ by the following:

Claim 4. *With the choice of* $\zeta^{(t)} = \begin{cases} T_0^{-2}, & \text{if } t \le T_0; \\ t^{-2}, & \text{if } t > T_0. \end{cases}$

If $c^{(t)} = \frac{4}{3} N \xi^3 \frac{\ln(1/\zeta^{(t)})}{t}$, we have:

$$\begin{cases} \sum_{t\in[T]} \big(c^{(t)}\big)^2 \le \frac{64N^2\xi^6}{9}\left[\frac{\pi^2(\ln T_0)^2}{6} + 2\right], \\ \sum_{t\in[T]} \frac{c^{(t)}}{t} \le \frac{8N\xi^3}{3}\left[\frac{\pi^2 \ln T_0}{6} + 1\right]. \end{cases} \tag{32}$$

If $c^{(t)} = 2\sqrt{N\xi^5 \sum_m \mu_m}\sqrt{\frac{\ln(1/\zeta^{(t)})}{t}}$, we have:

$$\begin{cases} \sum_{t\in[T]} (c^{(t)})^2 \le 4N\xi^5 \sum_m \mu_m \left[(\ln T_0)^2 + (\ln T)^2\right], \\ \sum_{t\in[T]} \frac{c^{(t)}}{t} \le \sqrt{8N\xi^5 \sum_m \mu_m}\Big\{(2+\sqrt{2})\sqrt{\ln T_0} \\ \qquad + \sqrt{2\pi}\,\mathrm{erf}\Big(\sqrt{\ln T/2}\Big)\Big\}. \end{cases} \tag{33}$$

Step 5: With a probability no more than $N\sum_t \zeta^{(t)} \le \frac{2N}{T_0}$, we have $R_T \le \xi^2 \sum_{i\in[N]} \mu_i \sqrt{N}$. Also, with a probability at least $1 - N\sum_t \zeta^{(t)}$, we have the bound of R_T in Claim 3. By plugging bounds of $\sum_t (c^{(t)})^2$ and $\sum_t \frac{c^{(t)}}{t}$ into Claim 3, we can prove Theorem 4.■

IV. Learning to Minimize the Simulation Cost

In this section, we present the design of our SimCos algorithm, which learns $\boldsymbol{w}^*$ and minimizes the simulation cost in an online manner. Then, we present the algorithm regret upper bound and reveal the impact of ξ-similarity on the learning speed. We also provide the key idea to prove the regret upper bound.

A. The Design of SimCos Algorithm

The key idea of our SimCos Algorithm is that at each round of learning: (1) *First develop a linear approximation framework to locate the search direction*; (2) *Then design an estimator to estimate the search direction from simulation samples*; (3) *Finally, use the estimated search direction to select the arm.*

Search direction. One challenge in locating the search direction is that the objective $L_{\mathrm{SimCos}}(\boldsymbol{w})$ takes the pointwise maximum of functions $\ell_n(\boldsymbol{w})$, leading to the nonsmoothness. Another constraint is that Problem 2 implies a step size of $1/t$ in updating $\boldsymbol{w}^{(t)}$, i.e.,

$$\boldsymbol{w}^{(t+1)} = \frac{t\boldsymbol{w}^{(t)} + \boldsymbol{z}}{t+1}, \tag{34}$$

where $\boldsymbol{z} \in \Delta$. Namely, to determine the search direction, we first need to determine $\boldsymbol{z}$. To measure the potential of $\boldsymbol{z}$ in decreasing $L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)})$, we take a *linearization* of $L_{\mathrm{SimCos}}(\boldsymbol{w})$ at $\boldsymbol{w} = \boldsymbol{w}^{(t)}$:

$$\begin{aligned} L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)}; \boldsymbol{z}) &= \max_{n\in[N]} \ell_n(\boldsymbol{w}^{(t)}) + \nabla \ell_n(\boldsymbol{w}^{(t)})^\top (\boldsymbol{w}^{(t+1)} - \boldsymbol{w}^{(t)}) \\ &= \max_{n\in[N]} \ell_n(\boldsymbol{w}^{(t)}) + \nabla \ell_n(\boldsymbol{w}^{(t)})^\top \frac{\boldsymbol{z} - \boldsymbol{w}^{(t)}}{t+1}, \end{aligned} \tag{35}$$

and bound its approximation error in the following lemma:

Lemma 1. $\big|L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)}; \boldsymbol{z}) - L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t+1)})\big| = O\big(\frac{\xi^3}{(t+1)^2}\big)$.

Lemma 1 states that the approximation error of linear approximation decreases at a rate of $1/t^2$. This implies that the linear approximation is asymptotically accurate in approximating the $L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t+1)})$. Hence, given $\boldsymbol{w}^{(t)}$, we consider the *minimizer of* $L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)}; \boldsymbol{z})$ as the search direction. Furthermore, the minimum of $L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)}; \boldsymbol{z})$ can be attained by *the standard direction with steepest decrease*, i.e.,

$$\min_{\boldsymbol{z}\in\Delta} L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)}; \boldsymbol{z}) = \min_{\boldsymbol{z}\in\mathcal{U}} L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)}; \boldsymbol{z}),$$

where $\mathcal{U} \triangleq \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_N\}$ represents the standard basis. This implies that we can reduce the search space from Δ to $\mathcal{U}$, and simplify estimations of the search direction as we proceed to show. We take such steepest decrease direction as the search direction, and denote it by:

Algorithm 2 SimCos MIS-Learning

Input: N, $\boldsymbol{w}=(\frac{1}{N},...,\frac{1}{N})$
for all $t\leq N$ **do**
  Draw $\boldsymbol{x}^{(t)}$ according to the distribution $Q_t(\boldsymbol{x})$ and record history: $Q_t(\boldsymbol{x}^{(t)})$ and $\mathbf{1}_{\mathcal{E}_t}(\boldsymbol{x}^{(t)})$.
for all $t>N$ **do**
  Estimate $\mu_n^{(t-1)}, n\in[N]$ by $\widehat{\mu}_n^{(t-1)}=\frac{1}{t-1}\sum_{s\in[t-1]}\frac{P(\boldsymbol{x}^{(s)})\mathbf{1}_{\mathcal{E}_n}(\boldsymbol{x}^{(s)})}{Q(\boldsymbol{x}^{(s)};\boldsymbol{w}^{(s)})}$.
  For all arms $n\in[N]$, compute $g_n^{(t)}$, i.e., the estimated linear approximation of decreasing progress achieved by taking different arms at round t according to Eq. (38).
  Compute the LCB $\underline{g}_n^{(t)}$, where $\underline{g}_n^{(t)}=g_n^{(t)}-c_n^{(t)}$.
  Select arm $I_t\in\operatorname{argmin}_{n\in[N]}\underline{g}_n^{(t)}$.
  Update $\boldsymbol{w}^{(t)}\leftarrow\boldsymbol{w}^{(t-1)}+\frac{1}{t}(e_{I_t}-\boldsymbol{w}^{(t-1)})$.

$$\boldsymbol{e}_{*_t}=\operatorname{argmin}_{\boldsymbol{z}\in\mathcal{U}}L_{\text{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{z}). \tag{36}$$

Search direction estimation. We consider the following equivalent form of the search direction:

$$\boldsymbol{e}_{*_t}=\operatorname{argmin}_{\boldsymbol{z}\in\mathcal{U}}L_{\text{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{z})-L_{\text{SimCos}}(\boldsymbol{w}^{(t)}). \tag{37}$$

Such form of search direction is useful to estimate the search direction, for the value of $L_{\text{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{z})-L_{\text{SimCos}}(\boldsymbol{w}^{(t)})$ shrinks in t. As we will show later, this property enables us to derive better concentration results for the search direction estimation. As the search direction is in the set $\mathcal{U}$, we only need to estimate $\{L_{\text{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_n)\}_{n=1}^N$ and $L_{\text{SimCos}}(\boldsymbol{w}^{(t)})$ so to locate $\boldsymbol{e}_{*_t}$. Essentially, we need to estimate $\ell_n(\boldsymbol{w}^{(t)})$ and $\nabla\ell_n(\boldsymbol{w}^{(t)})$ from the data samples $\{\boldsymbol{x}^{(s)}\}_{s=1}^{t-1}$. We have similar challenges as in Section III-A, i.e., data samples are not IID. We then address these challenges with a similar method: We estimate $L_{\text{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_n)-L_{\text{SimCos}}(\boldsymbol{w}^{(t)})$ as $g_n^{(t)}$ where $g_n^{(t)}$ is derived as $g_n^{(t)}\triangleq\dot{g}_n^{(t)}-\ddot{g}^{(t)}$ and:

$$\dot{g}_n^{(t)}=\max_{i\in[N]}\tfrac{t+1}{t}\widehat{A}_i\left(\boldsymbol{w}^{(t-1)}\right)-\tfrac{1}{t}\widehat{B}_i\left(\boldsymbol{w}^{(t-1)};n\right)-\tfrac{(\widehat{\mu}_i^{(t-1)})^2}{(\widehat{\mu}_i^{(t-1)}-o_i)^2},$$
$$\ddot{g}^{(t)}=\max_{i\in[N]}\widehat{A}_i\left(\boldsymbol{w}^{(t-1)}\right)-\tfrac{(\widehat{\mu}_i^{(t-1)})^2}{(\widehat{\mu}_i^{(t-1)}-o_i)^2},$$
$$\widehat{B}_i\left(\boldsymbol{w}^{(t)};n\right)=\tfrac{1}{(\widehat{\mu}_i^{(t-1)}-o_i)^2}\tfrac{1}{t}\sum_{s\in[t]}\tfrac{P^2\left(\boldsymbol{x}^{(s)}\right)\mathbf{1}_{\mathcal{E}_i}\left(\boldsymbol{x}^{(s)}\right)}{Q^2\left(\boldsymbol{x}^{(s)};\boldsymbol{w}^{(t)}\right)}\tfrac{Q_n\left(\boldsymbol{x}^{(s)}\right)}{Q\left(\boldsymbol{x}^{(s)};\boldsymbol{w}^{(t)}\right)},$$
$$\widehat{A}_i\left(\boldsymbol{w}^{(t)}\right)=\tfrac{1}{(\widehat{\mu}_i^{(t-1)}-o_i)^2}\cdot\tfrac{1}{t}\sum_{s\in[t]}\tfrac{P^2(\boldsymbol{x}^{(s)})\mathbf{1}_{\mathcal{E}_i}(\boldsymbol{x}^{(s)})}{Q^2\left(\boldsymbol{x}^{(s)};\boldsymbol{w}^{(t)}\right)}. \tag{38}$$

In the following theorem, we prove that the search direction can be estimated asymptotically accurate.

Theorem 5. *Consider the MIS-Learning framework. where at round $t, t\in[T]$ take distribution $Q_{I_t}(\boldsymbol{x})$ to generate $\boldsymbol{x}^{(t)}$. Then*

$$\lim_{t\to\infty}\|\dot{g}_n^{(t)}-L_{\text{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_n)\|=0, \tag{39}$$
$$\lim_{t\to\infty}\|\ddot{g}^{(t)}-L_{\text{SimCos}}(\boldsymbol{w}^{(t)})\|=0. \tag{40}$$

Remark: Similar as Theorem 1, such asymptotic property owns much to the mixture parameter $\boldsymbol{w}^{(t)}$.

Arm selection. Based on $g_n^{(t)}, n\in[N]$, we estimate the steepest search direction using the LCB framework and we outline the arm selection in Algorithm 2. Selecting the parameter $c_n^{(t)}$ is closely related to the regret of Algorithm 2.[2] We thus delay the selection in the next subsection, where we analyze the regret.

[2]The derivation of c_n in Algorithm 2 relies on A_i, B_i, and their moments, which is discussed in the next subsection. In the implementation, we take empirical estimations of these values. This will not affect our regret conclusion as its derivation utilizes the upper bounds of A_i, B_i, and their moments.

B. Regret Analysis of SimCos Algorithm

To first decompose the regret, denote the optimal mixture as $\boldsymbol{w}^*$, the optimal search direction as $\boldsymbol{e}_{*_t}$, and the estimated search direction (i.e., the action direction) as $\boldsymbol{e}_{I_t}$. Then we decompose the regret as follows:

$$\begin{aligned}&L_{\text{SimCos}}(\boldsymbol{w}^{(t+1)})-L_{\text{SimCos}}(\boldsymbol{w}^*)\\ &\leq L_{\text{SimCos}}\left(\tfrac{t\boldsymbol{w}^{(t)}+\boldsymbol{e}_{I_t}}{t+1}\right)-L_{\text{SimCos}}\left(\tfrac{t\boldsymbol{w}^{(t)}+\boldsymbol{e}_{*_t}}{t+1}\right) &\text{(R1)}\\ &+L_{\text{SimCos}}\left(\tfrac{t\boldsymbol{w}^{(t)}+\boldsymbol{e}_{*_t}}{t+1}\right)-L_{\text{SimCos}}\left(\tfrac{t\boldsymbol{w}^{(t)}+\boldsymbol{w}^*}{t+1}\right) &\text{(R2)}\\ &+L_{\text{SimCos}}\left(\tfrac{t\boldsymbol{w}^{(t)}+\boldsymbol{w}^*}{t+1}\right)-L_{\text{SimCos}}(\boldsymbol{w}^*). &\text{(R3)}\end{aligned}$$

This decomposition has three parts. Part $R1$ is the *estimation error*, and is essentially governed by the concentration of $g_n^{(t)}$ in estimating $L_{\text{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_n)-L_{\text{SimCos}}(\boldsymbol{w}^{(t)})$. Part $R2+R3$ is the *approximation error*, and is essentially governed by the convexity and smoothness of the objective $L_{\text{SimCos}}(\boldsymbol{w})$.

Next, similar as the SumVar case, we first establish two building blocks: (1) *The strong convexity and smoothness properties of $L_{\text{SimCos}}(\boldsymbol{w})$ and its components*; (2) *The concentration property of $\boldsymbol{g}^{(t)}$ in estimating $L_{\text{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{z})-L_{\text{SimCos}}(\boldsymbol{w}^{(t)})$*. Then we apply these two blocks to bound the regret of Algorithm 2.

Convexity and smoothness of $L_{\text{SimCos}}(\boldsymbol{w})$ and its components. As an immediate consequence of Theorem 2, we can derive the strong convexity and smoothness of $\ell_n(\boldsymbol{w}), n\in[N]$, i.e., the components of $L_{\text{SimCos}}(\boldsymbol{w})$:

Corollary 1. *If $\{\mathcal{E}_n\}_{n=1}^N$ has a ξ-similarity, then $\ell_n(\boldsymbol{w}), n\in[N]$ in Eq. (9) is α_n-strongly convex and β_n-smooth, where*

$$\alpha_n=\tfrac{2\mu_n^2}{\xi^2(\mu_n-o_n)^2}\text{ and }\beta_n=\tfrac{2\xi^3\mu_n}{(\mu_n-o_n)^2}. \tag{41}$$

Such convexity and smoothness of $\ell_n(\boldsymbol{w}), n\in[N]$ guarantee the convexity of $L_{\text{SimCos}}(\boldsymbol{w})$:

Corollary 2. *If $\{\mathcal{E}_n\}_{n=1}^N$ has a ξ-similarity, then $L_{\text{SimCos}}(\boldsymbol{w})$ in Eq. (13) is α'-strongly convex, where*

$$\alpha'\triangleq\min_{n\in[N]}\alpha_n=\min_{n\in[N]}\tfrac{2\mu_n^2}{\xi^2(\mu_n-o_n)^2}. \tag{42}$$

Remark: Corollary 1 and 2 quantify the impact of ξ-similarity on the strong convexity and smoothness of $L_{\text{SimCos}}(\boldsymbol{w})$ and its components. Also note that the tight approximation mentioned in Lemma 1 is guaranteed by the strong convexity and smoothness of $\ell_n(\boldsymbol{w}), n\in[N]$.

Concentration property of $g_n^{(t)}$. In the following theorem, we characterize how well the estimator $g_n^{(t)}$ concentrates around the $L_{\text{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_n)-L_{\text{SimCos}}(\boldsymbol{w}^{(t)})$.

Theorem 6. *Assume $\boldsymbol{x}\sim Q(\boldsymbol{x};\boldsymbol{w}^{(t-1)})$ for both $\mathbb{E}$ and $\mathbb{V}$. For any random variable $X(\boldsymbol{x})$ define $\widetilde{X}(\boldsymbol{x})\triangleq X(\boldsymbol{x})-\mathbb{E}X(\boldsymbol{x})$ and $\varphi(X(\boldsymbol{x}))\triangleq\frac{2\ln(8/\zeta^{(t)})}{3t}\max\widetilde{X}(\boldsymbol{x})+\sqrt{\frac{2\ln(8/\zeta^{(t)})}{t}\mathbb{V}\widetilde{X}(\boldsymbol{x})}$. Suppose $\zeta^{(t)}$ and $\epsilon^{(t)}$ satisfy:*

$$\begin{cases}\zeta^{(t)}=T_0^{-2},\ \epsilon_n^{(t)}=\frac{C_1}{t+1}, & \text{if } t\leq T_0;\\ \zeta^{(t)}=t^{-2},\ \epsilon_n^{(t)}=\max_{k\in[N]}\frac{1}{t+1}\left(a_k^{(t)}+b_{k,n}^{(t)}\right), & \text{if } t>T_0;\end{cases}$$

where $A_i(\boldsymbol{x};\boldsymbol{w}^{(t)})=\frac{1}{(\mu_i-o_i)^2}\frac{P^2(\boldsymbol{x})\mathbf{1}_{\mathcal{E}_i}(\boldsymbol{x})}{Q^2\left(\boldsymbol{x};\boldsymbol{w}^{(t)}\right)}$,

$$B_i(\boldsymbol{x};\boldsymbol{w}^{(t)};n)=\tfrac{1}{(\mu_i-o_i)^2}\tfrac{P^2(\boldsymbol{x})\mathbf{1}_{\mathcal{E}_i}(\boldsymbol{x})Q_n(\boldsymbol{x})}{Q^3\left(\boldsymbol{x};\boldsymbol{w}^{(t)}\right)},$$
$$a_i^{(t)}=\varphi\left(A_i(\boldsymbol{x};\boldsymbol{w}^{(t)})\right),\quad b_{i,n}^{(t)}=\varphi\left(B_i(\boldsymbol{x};\boldsymbol{w}^{(t)};n)\right),$$
$$C_1=\max_{k\in[N]}\tfrac{2\xi^2\mu_k}{(\mu_k-o_k)^2}+\max_{k\in[N]}\tfrac{2\xi^3}{(\mu_k-o_k)^2}.$$

Then, it holds that

$$\mathbb{P}\Big[g_n^{(t)}-(L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_n)-L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)}))\leq\epsilon_n^{(t)}\Big]\leq\zeta^{(t)},$$
$$\mathbb{P}\Big[g_n^{(t)}-(L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_n)-L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)}))\geq-\epsilon_n^{(t)}\Big]\leq\zeta^{(t)}.$$

Remark: We need to point out that $a_i^{(t)}+b_{i,n}^{(t)}=O\Big(\sqrt{\xi^3\frac{\ln(8/\zeta^{(t)})}{t}}\Big)$ and $C_1=O(\xi^3)$. Therefore, Theorem 6 reveals the impact of ξ-similarity on the concentration of estimation.

Regret upper bound. With the regret decomposition and above two building blocks, we now select the parameter of Algorithm 2 and prove its regret upper bound. Due to page limit, we present a sketch proof in the next subsection.

Theorem 7 (Regret upper bound of SimCos algorithm)**.** *Suppose $\{\mathcal{E}_n\}_{n=1}^N$ has a "ξ-similarity". For MIS-Learning problem with cost measure L_{SimCos} in Eq. (13), after T steps of the SimCos algorithm, the choice of $c_n^{(t)}=\epsilon_n^{(t)}$ and*

$$\zeta^{(t)}=\begin{cases} T_0^{-2}, & \text{if } t\leq T_0;\\ t^{-2}, & \text{if } t>T_0,\end{cases}$$

holds the following:

$$\mathbb{E}[R_T]\leq O(\xi^3)\frac{1}{T}+O(\beta'+\xi^3)\frac{\ln T}{T} +O(\xi^3)\frac{(\ln T)^2}{T}+O(\xi^{5/2})\sqrt{\frac{\ln T}{T}}. \tag{43}$$

Remark: Theorem 7 shows that the regret upper bound is proportional to the ξ-similarity. It also reveals that a small ξ implies a fast convergence to the optimal mixture.

C. Proof Sketch

Here we state the sketch proof for Theorem 7, in which we discuss the regret upper bound of the SimCos algorithm.

Still, let $\boldsymbol{w}^*$ be the optimal mixture:

$$\boldsymbol{w}^*=\operatorname{argmin}_{\boldsymbol{w}\in\Delta} L_{\mathrm{SimCos}}(\boldsymbol{w}). \tag{44}$$

Recall that $\boldsymbol{e}_{*_t}$, i.e., the steepest decent direction of the linearization $L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{z})$, is reorganized in Eq. (37). We estimate $L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_n)-L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)})$ as $g_n^{(t)}$ and so estimate $\boldsymbol{e}_{*_t}$ as $\boldsymbol{e}_{I_t}$ where $I_t=\operatorname{argmin}_{n\in[N]} g_n^{(t)}$. The proof of Theorem 7 can be broken down into the following steps.

Step 1: We first partition the regret as:

$$L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t+1)})-L_{\mathrm{SimCos}}(\boldsymbol{w}^*)\leq R1+R2+R3, \tag{45}$$

where part $R1$, $R2$ and $R3$ are given in Section IV-B. In the following, we will bound each part of the regret.

Step 2: By the definition of linearization $L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{z})$, and the strongly convex and smooth properties of $\ell_n(\boldsymbol{w})$ given in Corollary 1, we have:

$$R1\leq L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_{I_t})-L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_{*_t}) +\frac{\beta'}{2}\Big\|\frac{\boldsymbol{e}_{I_t}-\boldsymbol{w}^{(t)}}{t+1}\Big\|_2^2-\frac{\alpha'}{2}\Big\|\frac{\boldsymbol{e}_{*_t}-\boldsymbol{w}^{(t)}}{t+1}\Big\|_2^2, \tag{46}$$

where $\alpha'=\min_{n\in[N]}\alpha_n$, $\beta'=\max_{n\in[N]}\beta_n$. To bound $R1$, we then look at $c_n^{(t)}$, i.e., the confidence bound when estimating the $L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_n)-L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)})$ with $g_n^{(t)}$, which affects the accuracy of estimating $\boldsymbol{e}_{*_t}$ with $\boldsymbol{e}_{I_t}$ when $n=I_t$. The relationship between $L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_{I_t})-L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_{*_t})$ and $c_{I_t}^{(t)}$ can be revealed by the following claim:

Claim 5. *Assume $c_n^{(t)}$ satisfies*

$$\mathbb{P}\Big[g_n^{(t)}-\Big(L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_{I_t})-L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_{*_t})\Big)\leq c_n^{(t)}\Big]\leq\zeta^{(t)}, \tag{47}$$

$$\mathbb{P}\Big[g_n^{(t)}-\Big(L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_{I_t})-L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_{*_t})\Big)\geq -c_n^{(t)}\Big]\leq\zeta^{(t)}. \tag{48}$$

Then with a probability at least $1-2\zeta^{(t)}$,

$$L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_{I_t})-L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_{*_t})\leq 2c_{I_{t+1}}^{(t)}.$$

By the above discussion, we bound R_1 by the following:

$$R1\leq 2c_{I_t}^{(t)}+\frac{\beta'}{2}\Big\|\frac{\boldsymbol{e}_{I_t}-\boldsymbol{w}^{(t)}}{t+1}\Big\|_2^2-\frac{\alpha'}{2}\Big\|\frac{\boldsymbol{e}_{*_t}-\boldsymbol{w}^{(t)}}{t+1}\Big\|_2^2. \tag{49}$$

Step 3: By the definition of $L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{z})$ and the optimality of $\boldsymbol{e}_{*_t}$, we show that $R2$ is upper bounded by:

$$R2\leq-\frac{\alpha'}{2}\Big\|\frac{\boldsymbol{w}^*-\boldsymbol{w}^{(t)}}{t+1}\Big\|_2^2+\frac{\beta'}{2}\Big\|\frac{\boldsymbol{e}_{*_t}-\boldsymbol{w}^{(t)}}{t+1}\Big\|_2^2. \tag{50}$$

Step 4: By the strong convexity and smoothness properties of $\ell_n(\boldsymbol{w})$, we show that $R3$ is upper bounded by:

$$R3\leq\frac{t}{t+1}\Big[L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)})-L_{\mathrm{SimCos}}(\boldsymbol{w}^*)\Big]+\frac{\beta'-\alpha'}{2}\Big\|\frac{\boldsymbol{w}^*-\boldsymbol{w}^{(t)}}{t+1}\Big\|_2^2. \tag{51}$$

Step 5: Combine the upper bounds of part R_1, R_2 and $R3$, we show that:

$$R_T\leq\frac{2}{T}\sum_{t\in[T-1]}c_{I_t}^{(t)}+3(\beta'-\alpha')\frac{\ln(T/2)}{T}. \tag{52}$$

Step 6: Next, we focus on bounding $c_n^{(t)}$, which measures the accuracy in estimating $L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)};\boldsymbol{e}_n)-L_{\mathrm{SimCos}}(\boldsymbol{w}^{(t)})$ with $g_n^{(t)}$. By Theorem 6, we have the upper bounds of $c_n^{(t)}$ under different conditions, which depend on C_1, $a_i^{(t)}$ and $b_{i,n}^{(t)}$. The bound of C_1 is given by Theorem 6. Then $a_i^{(t)}$ and $b_{i,n}^{(t)}$ are bounded by:

$$a_i^{(t)}=\varphi\big(A_i(\boldsymbol{x};\boldsymbol{w}^{(t)})\big) \leq\frac{2\xi^2}{3(\mu_i-o_i)^2}\frac{\ln(8/\zeta^{(t)})}{t}+\frac{\sqrt{2\xi^3\mu_i}}{(\mu_i-o_i)^2}\sqrt{\frac{\ln(8/\zeta^{(t)})}{t}}; \tag{53}$$

$$b_{i,n}^{(t)}=\varphi\big(B_i(\boldsymbol{x};\boldsymbol{w}^{(t)};n)\big) \leq\frac{2\xi^3}{3(\mu_i-o_i)^2}\frac{\ln(8/\zeta^{(t)})}{t}+\frac{\sqrt{2\xi^5\mu_i}}{(\mu_i-o_i)^2}\sqrt{\frac{\ln(8/\zeta^{(t)})}{t}}. \tag{54}$$

Combining Theorem 6, and Eq. (52), (53) and (54), we finish the proof of Theorem 7. ∎

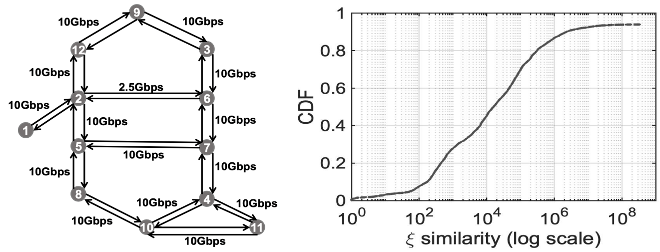
V. Applications

In this section, we demonstrate the efficiency of our MIS-Learning framework in evaluating risks for a set of rare threats. We take the Abilene backbone network [8], [9] as an example to show how our method works in detail. And we aim at evaluating the impact of *network link failures* on the occurrences of interested events $\mathcal{E}_n, n\in[N]$, which are specified as the *non-satisfaction of bandwidth demands for traffic flows n, $n\in[N]$*. The numerical results show that our SumVar and SimCos algorithms reduce the associated cost measures by 37.8% and 61.6% respectively, compared with the uniform mixture IS.

A. Problem Description

As depicted in Fig. 2(a), the network contains 12 nodes and 30 links. Each link fails with a probability of 0.01 and link failure occurrences are indicated by $\boldsymbol{x}$. The topology and traffic matrices are collected from [10]. There are 132 competing flows, and their bandwidth demands are extracted from [5]. The flow routing follows the shortest path policy. The capacity allocation follows the max-min fairness policy, which is also adopted by Google's B4 backbone network [11].

For each interested event $\mathcal{E}_n$ with the occurrence probability of μ_n, we take the *customized pure IS distribution in* [5] as the efficient IS distribution $Q_n(\boldsymbol{x})$ of $\mathcal{E}_n$. To accurately estimate $\{\mu_n\}$



(a) The Abilene network. (b) CDF of ξ similarity.

Fig. 2: The network topology and ξ-similarity information.

for a set of events $\{\mathcal{E}_n\}$, authors in [5] consider the MIS solution with a *uniform mixture* $\boldsymbol{w}=(\frac{1}{N},...,\frac{1}{N})$. In the following, we apply our MIS-Learning framework to learn a more efficient mixture $\boldsymbol{w}^*$ which minimizes the cost measure $L(\boldsymbol{\sigma}(\boldsymbol{w}))$.

We first derive ξ-similarities between any two interested events $\mathcal{E}_{n_1}$ and $\mathcal{E}_{n_2}$, $n_1, n_2 \in [N]$. The cumulative probability distribution (CDF) of the *pairwise* ξ-similarity is provided in Fig. 2(b). By setting the upper thresholds of the pairwise ξ-similarity, we can partition $\{\mathcal{E}_n\}_{n=1}^N$ into different subsets, on which we apply our MIS-Learning method to find an efficient mixture $\boldsymbol{w}$ to estimate their occurrence probabilities simultaneously. We set the upper bounds of the pointwise ξ-similarity as $\xi \leq 100$, $\xi \leq 200$, $\xi \leq 300$ and $\xi \in [1000, 5000]$, and obtain the corresponding event subsets $\{\mathcal{E}_n\}_{n=1}^{N'}$ with set sizes of $N'=16$, $N'=19$, $N'=30$ and $N'=5$.

B. Numerical Results

Minimizing the sum of variances. We start with the SumVar MIS-Learning with $L(\boldsymbol{\sigma}(\boldsymbol{w})) \triangleq L_{\text{SumVar}}(\boldsymbol{w})$. For each event subset $\{\mathcal{E}_n\}_{n=1}^{N'}$ with the corresponding ξ-similarity threshold, we run the SumVar MIS-Learning for 80,000 runs and plot the cost measure $L_{\text{SumVar}}(\boldsymbol{w})$ of each round in Fig. 3. We then compare the result with the uniform mixture proposed in [5]. Fig. 3(a), 3(b) and 3(c) illustrate the reduction of $L_{\text{SumVar}}(\boldsymbol{w})$ achieved by the SumVar MIS-Learning with a small ξ-similarity, and 3(d) illustrates the performance of the SumVar MIS-Learning with a large ξ-similarity. The SumVar MIS-Learning with Algorithm 1 reduces the cost measure by 25.1%, 23.6%, 26.4% and 37.8% when $\xi \leq 100$, $\xi \leq 200$, $\xi \leq 300$ and $\xi \in [1000, 5000]$.

Minimizing the simulation cost. We then consider the SimCos MIS-Learning with $L(\boldsymbol{\sigma}(\boldsymbol{w})) \triangleq L_{\text{SimCos}}(\boldsymbol{w})$. For each event subset $\{\mathcal{E}_n\}_{n=1}^{N'}$ with the corresponding ξ-similarity threshold, we run the SimCos MIS-Learning for 80,000 runs and plot the cost measure $L_{\text{SimCos}}(\boldsymbol{w})$ of each round in Fig. 3, and compare with the uniform mixture. Fig. 3(e), 3(f) and 3(g) show the reduction of $L_{\text{SimCos}}(\boldsymbol{w})$ achieved by the SimCos MIS-Learning with a small ξ-similarity, while 3(h) shows the performance of SimCos MIS-Learning with a large ξ-similarity. The SimCos MIS-Learning with Algorithm 2 reduces the cost measure by 35.7%, 55.1%, 39.9% and 61.6% when $\xi \leq 100$, $\xi \leq 200$, $\xi \leq 300$ and $\xi \in$[1000,5000].

Impact of ξ-similarity on the convergence rate. We take a detailed look at the convergence of cost measure in Fig. 4, and compare convergence rates of the large ξ case (i.e., $\xi \in [1000,$ $5000)$ and the small ξ case (i.e., $\xi \leq 300$). For the SumVar MIS-Learning with Algorithm 1, Theorem 4 implies that the regret $L_{\text{SumVar}}(\boldsymbol{w}) - L_{\text{SumVar}}(\boldsymbol{w}^*)$ first decreases at a fast rate in Eq. (21) and then at a slow rate in Eq. (22). Theorem 4 also reveals that a smaller ξ implies a longer fast rate period. As shown in Fig. 4, with a small ξ, $L_{\text{SumVar}}(\boldsymbol{w})$ decreases first at a fast rate and then at a slow rate; with a large ξ, the fast rate period vanishes. For SimCos MIS-Learning with Algorithm 2, Theorem 7 states that the regret decreases first at a fast rate of $O(1/T)$ and then at a slow rate of $Q(\sqrt{\ln T/T})$ in Eq. (43). As shown in Fig. 4, with a small ξ, $L_{\text{SimCos}}(\boldsymbol{w})$ decreases first at a fast rate and then at a slow rate; with a large ξ, the short fast rate period vanishes.

VI. Related work

A. MIS-Learning vs. IS and MIS

Comprehensive reviews on the rare event simulation are given in [12], [13]. These works are mainly IS based and focus on the single rare event estimation: they estimate the probability of rare event $\mathcal{E}_n$ by simulating the system under an alternative distribution $Q_n(\boldsymbol{x})$ and then unbias the results [14]. Given many rare events to estimate, as each $Q_n(\boldsymbol{x})$ is merely *customized* for $\mathcal{E}_n$ and may not work efficiently for other events, IS needs to "*sequentially*" estimate the occurrence of each $\mathcal{E}_n$ with its corresponding pure importance distribution $Q_n(\boldsymbol{x})$.

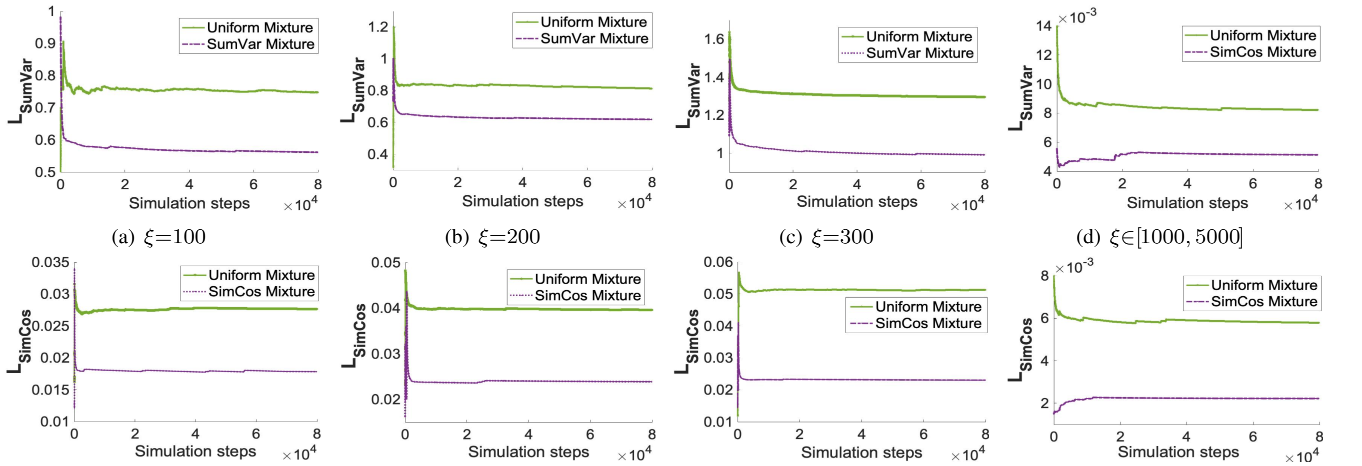
To efficiently estimate multiple rare events, authors in [5] propose using MIS to cooperate multiple $Q_n(\boldsymbol{x})$. However, most MIS based works take a *uniform mixture* [5] or *heuristic mixture strategies without theoretical guarantees* [15]. Some works [16], [17] consider computing the optimal mixture via standard convex optimization methods. However, they require that at each iteration, the variances (i.e., their cost measure) should either *be computed analytically* [17] or *be estimated accurately from sufficient samples* [16], which is unrealizable or computational expensive for the curse of dimensionality.

Our work aims to efficiently learn the optimal mixture working for estimations of many rare events, with a zero cost on extra samples. We reveal that *not all rare events can be efficiently estimated at the same time*, and we introduce the ξ-similarity to partition events into subsets with smaller ξ values, which can be efficiently estimated via MIS at the same time.

B. MIS-Learning vs. Stochastic Optimization

The MIS-Learning can be viewed as *the stochastic optimization (SO) problem over the simplex*: to minimize the objective function $L(\boldsymbol{\sigma}(\boldsymbol{w}))$, we choose at each round an action I_t, which affects the variable $\boldsymbol{w}$ and provides observations on $L(\boldsymbol{\sigma}(\boldsymbol{w}))$.

In the common case where objectives are *smooth*, i.e., $L(\boldsymbol{\sigma}(\boldsymbol{w})) \triangleq L_{\text{SumVar}}(\boldsymbol{w})$, *iterative gradient-based methods*, such as the gradient descent (GD) and stochastic gradient descent (SGD) [18], are popular optimization tools. Yet in our setting, *neither the gradient* $\nabla L_{\text{SumVar}}(\boldsymbol{w})$ *nor its components can be computed exactly* and so estimations are required. To accurately estimate $\nabla L_{\text{SumVar}}(\boldsymbol{w}^{(t)})$ and meanwhile guarantee a good convergence speed, SGD needs to generate sufficient simulation samples from $Q(\boldsymbol{x}; \boldsymbol{w}^{(t)})$ at each learning round t, making the learning cost unaffordable.



(a) ξ=100 (b) ξ=200 (c) ξ=300 (d) $\xi\in[1000, 5000]$

(e) ξ=100 (f) ξ=200 (g) ξ=300 (h) $\xi\in[1000, 5000]$

Fig. 3: The reduction of cost measure $L_{\mathrm{SumVar}}(\boldsymbol{x})$ (or $L_{\mathrm{SimCos}}(\boldsymbol{x})$) achieved by MIS-learning, compared with the uniform mixture. (a)-(d) show the SumVar case and (e)-(h) show the SimCos case; (a)-(c), (e)-(g) show the small ξ case, and (d), (f) show the large ξ case.

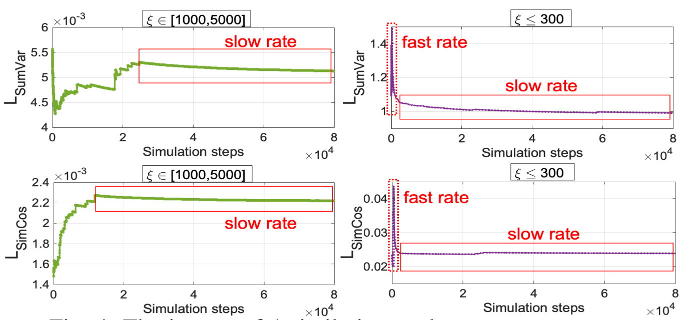


Fig. 4: The impact of ξ-similarity on the convergence rate

When objectives are *non-smooth*, i.e., a pointwise maximum function $L(\boldsymbol{\sigma}(\boldsymbol{w}))\triangleq L_{\mathrm{SimCos}}(\boldsymbol{w})$ with smooth components, *gradient mapping based methods*[19] guarantee an exponential regret convergence. Yet in our setting, it faces the same problem of *expensive gradient (or its components) estimation*. A more challenging point is *the constrained* $\boldsymbol{w}^{(t)}$ *updating*: the updating of $\boldsymbol{w}^{(t)}$ has a fixed step size of $1/t$ and constrained moving directions, i.e., $\boldsymbol{w}^{(t)}$=$\boldsymbol{w}^{(t-1)}+\frac{1}{t}(\boldsymbol{e}_{I_t}-\boldsymbol{w}^{(t-1)})$.

Our method solves these challenges and reduce the gradient (or its components) estimation cost by generating only one sample $\boldsymbol{x}$ from one of $\{Q_n(\boldsymbol{x})\}_{n=1}^N$ at each round. Hence, it has a "*zero cost on extra samples*". Besides, with SO, estimations of rare events $\{\mathcal{E}_n\}_{n=1}^N$ are performed only after deriving a proper $\boldsymbol{w}$. In other words, samples generated while optimizing $\boldsymbol{w}$ cannot be used for estimating $\{\mathcal{E}_n\}_{n=1}^N$. As a contrast, our method estimates $\{\mathcal{E}_n\}_{n=1}^N$ and learns the optimal mixture $\boldsymbol{w}^*$ at the same time. Thus, it also has a "*zero learning cost*".

C. MIS-Learning vs. MAB Optimization

The MIS-Learning is also similar to the MAB optimization [20], [21], where at each round t, we pick an action $\boldsymbol{e}_{I_t}$ and observe information on the loss function L. The major difference is that these works consider a *cumulative regret* $\frac{1}{T}\sum_{t\in[T]}L(\boldsymbol{e}_{I_t})$ but we focus on the *global loss* $L(\frac{1}{T}\sum_{t\in[T]}\boldsymbol{e}_{I_t})$.

Problems related to the MAB optimization with the global loss have been studied in [6], [22]–[24], where they consider minimizing a known loss $L(\boldsymbol{w}^{(t)\top}V)$ with an unknown matrix V. This differs from our setting where L is unknown and cannot be computed analytically. [22], [24] consider a stochastic setting and achieve a convergence rate of $O(\sqrt{1/T})$. The work in [23] considers an adversarial setting, but there are cases that their regrets cannot converge to zero. Our SumVar case is similar to [6], which considers the global loss $L(\frac{1}{T}\sum_{t\in[T]}\boldsymbol{e}_{I_t})$ and focuses on the strongly-convex and smooth loss function L. They consider $L(\boldsymbol{w})\triangleq\sum_{n\in[N]}\sigma_n^2/w_n$ with the unknown but fixed σ_n^2, $n\in[N]$. Yet, in our setting, $\sigma_n^2, n\in[N]$ also depend on $\boldsymbol{w}$.

VII. Conclusion

This paper develops a MAB OL framework to address the high simulation cost limitation of IS in dealing with *a set of rare events*. Our framework consists of a mixture importance sampling optimization problem (MISOP) and two OL algorithms. MISOP aims to select the optimal mixture attaining various tradeoffs, which are quantified by our cost measures. We first show that the objective function of MISOP is computationally expensive to evaluate. Then we extend MISOP to an OL setting to efficiently optimize the objective function without incurring any extra learning cost. Our SumVar and SimCos algorithms learn to minimize the *sum of variances* and *simulation cost* with regrets of $(\ln T)^2/T$ and $\sqrt{\ln T/T}$ respectively, where T is the number of samples. When applying to a realistic network, our methods reduce the cost measure value by 61.6% compared with the uniform mixture.

Acknowledgment

The research of John C.S. Lui is supported in part by the GRF 14201819. The research of Hong Xie is supported by National Nature Science Foundation of China (61902042). Hong Xie is the corresponding author.

References

[1] F. Cadini, G. L. Agliardi, and E. Zio, "Estimation of rare event probabilities in power transmission networks subject to cascading failures," *Reliability Engineering & System Safety*, 2017.

[2] L. Mikeev, W. Sandmann, and V. Wolf, "Efficient calculation of rare event probabilities in markovian queueing networks," in *Proc. ICST VALUETOOLS*, 2011.
[3] P. W. Glynn and D. L. Iglehart, "Importance sampling for stochastic simulations," *Manag. Sci.*, 1989.
[4] A. Rai, R. C. Valenzuela *et al.*, "Approximate zero-variance importance sampling for static network reliability estimation with node failures and application to rail systems," in *Proc. IEEE WSC*, 2016.
[5] T. Liu and J. C. S. Lui, "FAVE: a fast and efficient network flow AVailability estimation method with bounded relative error," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Apr. 2019.
[6] Q. Berthet and V. Perchet, "Fast rates for bandit optimization with upper-confidence frank-wolfe," in *Advances in Neural Information Processing Systems*, 2017, pp. 2225–2234.
[7] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval research logistics quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.
[8] "Abilene," https://www.internet2.edu/.
[9] W. Jiang, R. Zhang-Shen *et al.*, "Cooperative content distribution and traffic engineering in an isp network," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1. ACM, 2009, pp. 239–250.
[10] Y. Zhang, "Six months of Abilene traffic matrices," http://www.cs.utexas.edu/ yzhang/, 2014.
[11] S. Jain, A. Kumar *et al.*, "B4: Experience with a globally-deployed software defined wan," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 3–14.
[12] J. Bucklew, *Introduction to rare event simulation*. Springer Science & Business Media, 2013.
[13] G. Rubino and B. Tuffin, *Rare event simulation using Monte Carlo methods*. John Wiley & Sons, 2009.
[14] J. Frank, S. Mannor, and D. Precup, "Reinforcement learning in the presence of rare events," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 336–343.
[15] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of sram designs in the presence of rare failure events," in *ACM/IEEE Design Automation Conference*, 2006.
[16] J.-M. CORNUET, J.-M. MARIN *et al.*, "Adaptive multiple importance sampling," *Scandinavian Journal of Statistics*, 2012.
[17] H. Y. He and A. B. Owen, "Optimal mixture weights in multiple importance sampling," *arXiv preprint arXiv:1411.3954*, 2014.
[18] G. Papamakarios, "Comparison of modern stochastic optimization algorithms," 2014.
[19] Y. Nesterov, "Introductory lectures on convex programming volume i: Basic course," *Lecture notes*, 1998.
[20] E. Hazan, "The convex optimization approach to regret minimization," *Optimization for machine learning*, 2012.
[21] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.
[22] S. Agrawal, N. R. Devanur, and L. Li, "An efficient algorithm for contextual bandits with knapsacks, and an extension to concave objectives," in *Conference on Learning Theory*, 2016, pp. 4–18.
[23] A. Rakhlin, K. Sridharan, and A. Tewari, "Online learning: Beyond regret," 2011.
[24] S. Agrawal and N. R. Devanur, "Bandits with concave rewards and convex knapsacks," in *Proceedings of the fifteenth ACM conference on Economics and computation*. ACM, 2014, pp. 989–1006.