OPTIMAL DECISION MAKING FOR ACCELERATING SCIENTIFIC DISCOVERY

A Dissertation

by

HYUN-MYUNG WOO

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Byung-Jun Yoon
Co-Chair of Committee,	Edward R. Dougherty
Committee Members,	Xiaoning Qian
	Krishna R. Narayanan
	Yang Shen
	Theodora Chaspari
Head of Department,	Miroslav M. Begovic

August 2022

Major Subject: Electrical Engineering

ABSTRACT

Scientific discovery is the process of finding answers to scientific inquiries. Scientific discovery forms the basis of scientific/engineering applications as it serves as an operational objective or a means of achieving operational goals. In practice, scientific discovery is realized via (a sequence of) scientific decision-making that involves predicting the potential efficacy of available options and taking action that maximizes the expected utility of interest. Making optimal decisions is particularly important in real-world scientific/engineering applications as, potentially, it accelerates the discovery or even has a profound impact on the success of scientific applications.

In this dissertation, we comprehensively study the optimal decision-making problem for accelerating successful scientific discovery. Starting with a data-driven model that accelerates the optimal decision-making itself for a representative real-world scientific/engineering application, we propose mathematical optimization frameworks for identifying optimal decision-making. Based on the proposed models, we quantitatively analyze how fast this set of models can advance scientific discoveries for the applications.

In the first part, we consider the optimal decision-making problem in the context of optimal experimental design (OED). Identifying the optimal experiment that is expected to maximally reduce system uncertainty has become a critical problem in real-world scientific/engineering applications that involve modeling complex systems. Mean objective cost of uncertainty (MOCU)-based OED has shown that such a goal-driven OED is extremely useful in real-world problems that aim at achieving specific objectives based on complex uncertain systems. However, MOCU-based OED tends to be computationally expensive mainly due to the prediction cost of the potential efficacy of available experiments based on MOCU, which limits its practical applicability. To address this issue, we propose a novel ML scheme that can significantly accelerate MOCU computation and expedite MOCU-based experimental design. We apply the proposed ML-based OED acceleration scheme to design experiments aimed at optimally enhancing the control performance of uncertain Kuramoto oscillator models.

In the second part, we study an optimal decision-making problem for screening campaigns based on high-throughput virtual screening (HTVS) pipeline structures, which frequently arises in various scientific and engineering problems including drug discovery and materials design. We propose a general mathematical framework for optimizing HTVS pipelines that consist of multi-fidelity models. The central idea is to optimally allocate the computational resources to models with varying costs and accuracy to optimize the return-on-computational-investment (ROCI). Based on both simulated as well as real data, we demonstrate that the proposed optimal HTVS framework can significantly accelerate screening virtually without any degradation in terms of accuracy.

In the third part, based on the optimization framework we proposed in the second part of the dissertation, we design an optimal computational campaign (OCC) in the context of rapidly selecting redox-active organic materials for developing novel energy storage devices. Starting from a high-fidelity model that computes the redox potential (RP) of a given material, we show how a set of surrogate models with different accuracy and complexity may be designed to construct a highly accurate and efficient HTVS pipeline. Besides, we further generalize the screening condition and the optimization framework accordingly, which enables the design of computational screening campaigns according to a screening range. We demonstrate that the proposed HTVS pipeline remarkably enhances the overall throughput for a given computational budget.

DEDICATION

To my father, mother, sisters, and grandmother

and

Eun Jung, Juwon, and the little angel.

ACKNOWLEDGMENTS

First of all, I would like to express my deepest gratitude to my advisor, Professor Byung-Jun Yoon, for his invaluable guidance and support throughout my graduate studies. He was the perfect advisor, teacher, researcher, and mentor. He has always inspired me and motivated me to become a better researcher and person. He has always been on my side and guided me to the best path. He really showed me what a person looks like who wishes and tries to resemble His personality. All the things with him-learning from him, doing research with him, and being his Ph.D. student-are (will be) one of the happiest memories of my life.

I would like to thank my co-advisor, Professor Edward R. Dougherty, and committee members, Professor Xiaoning Qian, Professor Krishna R. Narayanan, Professor Yang Shen, and Professor Theodora Chaspari, for giving me excellent guidance, help, and teaching. I also would like to thank Professor Seung Soon Jang, Professor Sung Il Park, Professor Youngjoon Hong, and Professor Bongsuk Kwon for giving me invaluable opportunities for collaboration.

I am very thankful to my master's advisor, Professor Jaekwon Kim. Whenever I feel demotivated, I reminded myself of the moment when I first met you and decided to follow the path you passed through. You shaped my dream of becoming a professor who does research with pure curiosity and spread His love to students. I will move forward until I achieve my dream.

I would like to thank my former officemate, Hyundoo Jeong. He always gave me thoughtful advice regarding my life as a Ph.D. student and many practical bits of help. I could not make it without his kind support. My warmest appreciation goes to my officemate Omar Maddouri. He never hesitated to help me whenever I needed it. I would like to thank my former labmate, Mansuck Kim, for his kind advice and care. I also would like to thank my labmates: Alif Bin Abdul Qayyum, A N M Nafiz Abeer, Narges Zarnaghinaghsh, Pei-Hung Chung, Manasa Gadiyaram, and Mamoon Masud.

I have met many great people in the Korean church at College station. First, I would like to express my appreciation to my spiritual mentor, Pastor Sunyeop Lee. He prayed for me and taught

me many of the essential Christian doctrines to grow right in God. Besides, I would like to thank all the brothers and sisters in our church. I won't list them all because I'm afraid I'll miss some of them by mistake. However, I should say I am thankful to Sungmin Lee and Ji Won Nam for taking care of my wife so much after she gave birth.

I would like to thank my parents, Sung Ho Woo and Jong Soon Shin, for their unconditional love and unceasing support throughout my entire life. They always trusted me with love even when I was almost last in the whole school on my first high school midterm or when I told them I want to transfer from the department of administration to engineering for no reason when I was in my second year of university. I would like to thank my sisters, Ami Woo and Nami Woo, for always being supportive of what I did. I would like to thank my relatives for their encouragement. Especially, I would like to thank my cousin and best friend, Ilhwan Na, for his encouragement and fun conversations whenever I needed a refresh.

I would like to express my deepest thanks and love to my wife, Eun Jung Sim, and son, Juwon Woo, for filling my life with nothing but a full of happiness and love.

Last, but by no means least, I would like to thank God for guiding me in the right direction and having me meet such great brothers and sisters in you. I am nothing without you. Soli Deo Gloria! Amen.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Byung-Jun Yoon, Professor Edward R. Dougherty, Professor Xiaoning Qian, Professor Krishna R. Narayanan, and Professor Yang Shen of the Department of Electrical & Computer Engineering and Professor Theodora Chaspari of the Department of Computer Science & Engineering.

Funding Sources

Graduate study was supported in part by the National Science Foundation (NSF) under Award 1835690 and the Department of Energy (DOE) under Award DE-SC0019303.

NOMENCLATURE

CNN	Convolutional Neural Network
COVID-19	COronaVIrus Disease 2019
CPAT	Coding Potential Assessment Tool
CPC2	Coding Potential Calculator 2
CUDA	Compute Unified Device Architecture
DE	Differential Equation
DL	Deep Learning
DQN	Deep Q Network
DFT	Density Functional Theory
DNN	Deep Neural Network
EA	Electron Affinity
EM	Expectation-Maximization
fcNN	Fully-Connected Neural Network
FHSP-NLO	First-principles High-throughput Screening Pipeline for Non-Linear Optical materials
GAN	Generative Adversarial Network
GCN	Graph Convolutional Network
GPU	Graphics Processing Unit
GRN	Gene Regulatory Network
HOMO	Highest Occupied Molecular Orbital
HTVS	High-Throughput Virtual Screening
IMPECCABLE	Integrated Modeling PipelinE for COVID Cure by Assessing Better LEads
KRR	Kernel Ridge Regression

lncRNA	Long Non-Coding RiboNucleic Acid
LSTM	Long Short Term Memory
LUMO	Lowest Unoccupied Molecular Orbital
MD	Molecular Dynamics
ML	Machine Learning
MLE	Maximum Likelihood Estimation
MLP	MultiLayer Perceptron
MOCU	Mean Objective Cost of Uncertainty
NN	Neural Network
objective-UQ	objective-based Uncertainty Quantification
OCC	Optimal Computational Campaign
OCU	Objective Cost of Uncertainty
ODE	Ordinary Differential Equation
OED	Optimal Experimental Design
PDF	Probability Density Function
PLEK	Predictor of Long non-coding RNAs and mEssenger RNAs based on an improved K-mer scheme
RCT	Radical CyberTools
RNN	Recurrent Neural Network
ROCI	Return-On-Computational-Investment
RP	Redox Potential
SARS-CoV-2	Severe Acute Respiratory Syndrome CoronaVirus 2
SELFIES	SELF-referencIng Embedded Strings
SMILES	Simplified Molecular Input Line Entry System
SVR	Support Vector Regression
TRN	Transcription Regulatory Network

VAE	Variational AutoEncoder
ZINC	ZINC Is Not Commercial

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Scientific discovery-the process of finding answers to scientific inquiries-forms a basis of scientific/engineering applications as it serves as an operational objective or a means of achieving operational goals. Scientific discovery is realized via (a sequence of) scientific decision-making that involves predicting the potential efficacy of available options and taking action that maximizes the expected utility of interest. In real-world scientific/engineering applications, making optimal decisions is particularly important as they are often closely relevant to the allocation of experimental or computational resources, the progress, and even the outcomes of the applications.

The primary focus of the dissertation is the application of scientific frameworks to make the optimal decision for accelerating successful scientific discoveries in the context of real-world scientific/engineering applications. In Chapter 2, we propose a machine learning (ML)-based approach for accelerating decision-making to expedite the identification of the optimal experiment. In Chapter 3, we propose a mathematical framework that identifies the optimal operational policy of high-throughput virtual screening (HTVS) pipelines to accelerate computational screening campaigns. In chapter 4, we design an optimal computational campaign (OCC) for the effective detection of redox-active organic materials. To this aim, we propose an effective strategy to construct an HTVS pipeline based on a high-fidelity model. Besides, we further generalize the screening criterion of screening campaigns and the optimization framework proposed in Chapter 3, accordingly.

The purpose of the remaining part of this chapter is to provide readers with some gentle introductions to the scientific applications and signal processing techniques considered in the dissertation. In Section 1.1, we describe the optimal experimental design (OED) problem. We mathematically define the mean objective cost of uncertainty (MOCU) that forms the basis of MOCU-based OED in Section 1.2. In Section 1.3, we introduce a general computational screening campaign problem and HTVS pipelines. Then, we provide a high-level overview of neural network (NN) and kernel ridge regression (KRR) models in Section 1.4 and Section 1.5, respectively. Finally, we review parametric density estimation algorithms in Section 1.6. For a more extensive treatment

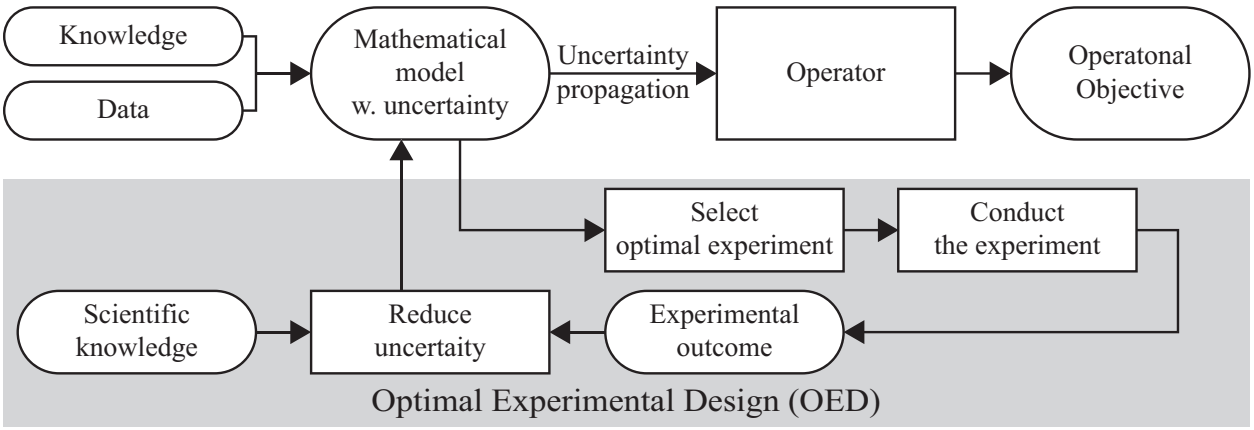


Figure 1.1: The flowchart depicting an optimal experimental design (OED) framework for reducing system uncertainty.

with mathematical derivations of these topics, the reader is referred to the references given at the end of each section.

1.1 Optimal experimental design (OED)

Various real-world scientific and engineering applications often involve the mathematical modeling of complex systems for designing operators of interest. A fundamental challenge in such applications is to construct a model that can generate precise responses with respect to the devised operators. Even with sufficient data, the accurate model construction may not be practically impossible due to the intrinsic complexity of the application of interest, causing substantial uncertainty in the model. The system uncertainty affects the subsequent processes–designing operator and performance evaluation. In such a case, based on scientific knowledge, one may consider OED to reduce system uncertainty, thereby improving the performance of the operators of interest.

Figure 1.1 shows an overview of the OED framework for an application incorporating mathematical system modeling. OED aims to identify the optimal experiment whose outcome is expected to maximally reduce the system uncertainty among all available experiments. The outcome of the selected experiment is used to reduce system uncertainty based on scientific knowledge.

In Chapter 2, we consider an optimal decision-making problem in OED in the context of robust synchronization of the uncertain Kuramoto model. For a comprehensive and detailed introduction

to OED and relevant signal processing techniques under system uncertainty, see [1].

1.2 Objective-based Uncertainty Quantification (objective-UQ)

Objective-UQ is the art of quantifying the system uncertainty that affects the operational objective, such as filtering, classification, estimation, or control. The central idea of Objective-UQ is MOCU which quantifies the system uncertainty based on the expected increase of the operational cost that it induces. MOCU is particularly important in effective experimental design as it provides an effective means of quantifying the expected efficacy of the experiments with respect to the operational objective.

Let $\theta \in \Theta$ be a system parameter vector governed by distribution $f(\theta)$. First, We define a robust operator over the uncertainty class Θ.

$$\psi^* = \arg\min_{\psi \in \Psi} E_\theta \left[\xi_\theta (\psi) \right], \tag{1.1}$$

where Ψ is a set of operators, and $\xi_\theta : \Psi \to [0, \infty)$ is a cost function within a set $\Theta = \{\xi_\theta | \theta \in \Theta\}$.

For a specific model parameter θ, the objective cost of uncertainty (OCU) is a differential cost of using ψ^* instead of the optimal operator ψ_θ for the model θ as follows:

$$U_{\Psi,\Xi,f}(\Theta, \theta) = \xi_\theta (\psi^*) - \xi_\theta (\psi_\theta). \tag{1.2}$$

Finally, MOCU is defined as the expectation of the OCU over $f(\theta)$ as follows:

$$M_{\Psi,\Xi,f}(\Theta) = E_\Theta \left[U_{\Psi,\Xi,f}(\Theta, \theta) \right]. \tag{1.3}$$

In Chapter 2, we utilize the concept of MOCU to quantify the system uncertainty and identify the optimal experiment that is expected to maximally minimize the system uncertainty that affects the operational performance. For a detailed description of MOCU and various applications, see [2].

1.3 High-throughput virtual screening (HTVS) pipeline

There has been a continuous need for efficient computational screening of molecular candidates that possess desired properties in various scientific and engineering problems, including drug

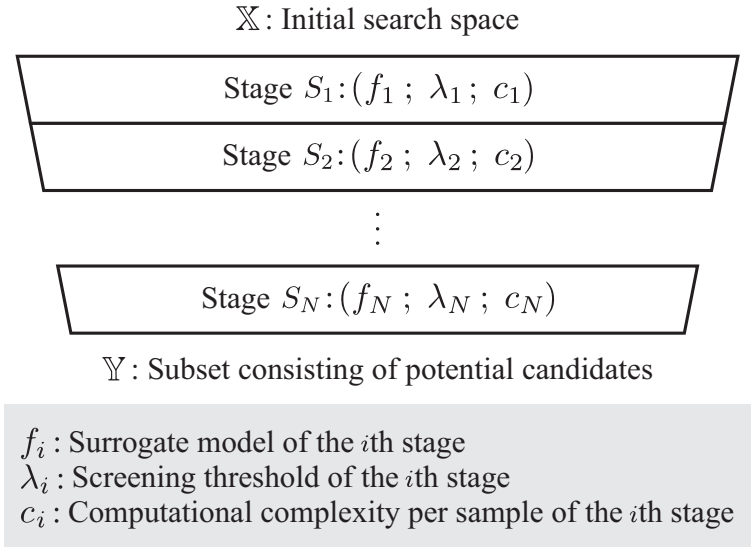


Figure 1.2: Illustration of a high-throughput virtual screening (HTVS) pipeline.

discovery and materials design. Fundamental challenges in such computational screening campaigns are (1) the large size of the search space containing the candidates and (2) the considerable computational cost of high-fidelity property prediction models. To address these issues, constructing and operating an HTVS pipeline that consists of multi-fidelity models is an effective approach to accomplish computational screening tasks.

Figure 1.2 illustrates a typical HTVS pipeline that consists of N stages, each of which is associated with a surrogate model f_i that evaluates the property of the molecules with a different accuracy/fidelity and computational cost c_i. Based on screening policy λ_i, each stage S_i evaluates all the molecular candidates to determine whether the evaluation score appears promising enough to warrant passing it to the next–often more computationally expensive but more accurate–stage without unnecessarily wasting computational resources and time. In this manner, the HTVS pipeline gradually narrows down the number of candidate molecules for investigating those that are promising and more likely to possess the desired property. The most potent candidates that remain at the end of screening may proceed to experimental validation. In Chapter 3, we provide the details on the stringent mathematical description and problem setup.

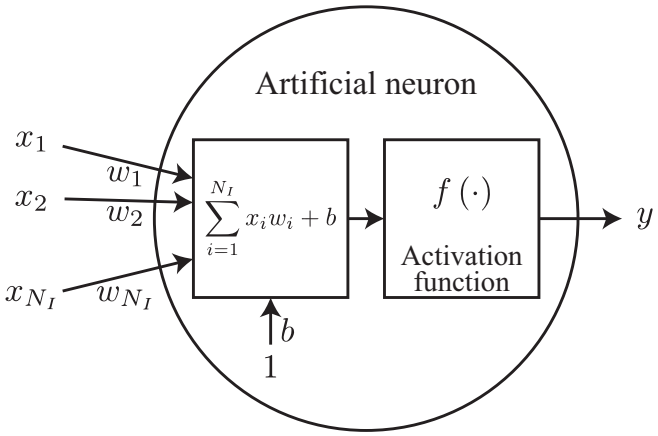


Figure 1.3: Illustration of an artificial neuron that makes up the neural network (NN).

1.4 Neural network (NN) model

(Artificial) NN is a computational model inspired by the human brain systems. A fundamental building block of NN is an artificial neuron shown in Figure 1.3. Formally, for N_I features $x_1, x_2, \ldots, x_{N_I}$, the artificial neuron aggregates weighted features with bias b and computes the response y through non-linear function f called the activation function as follows:

$$y = f\left(\sum_{i=1}^{N_I} x_i w_i + b\right). \tag{1.4}$$

NN is a network of artificial neurons that are connected via weighted directed edges. Figure 1.4 shows a general NN structure that computes N_O system responses of N_I input values. Thanks to the capability of learning sophisticated non-linear relationships between input and response, the NN model has been widely used in various research fields such as climate science [3, 4], medical science [5, 6], finance [7, 8], chemistry [9, 10], and nuclear science [11, 12], just to name a few. Besides, NN itself can serve as a building block for machine learning models. For example, variational autoencoder (VAE) [13], generative adversarial network (GAN) [14], and various types of deep Q networks (DQNs) [15, 16, 17, 18] have NN models as substructures to learn the relationship between the extracted features and responses of interest.

We utilize a NN model with one hidden layer as a surrogate classifier in order to accelerate the

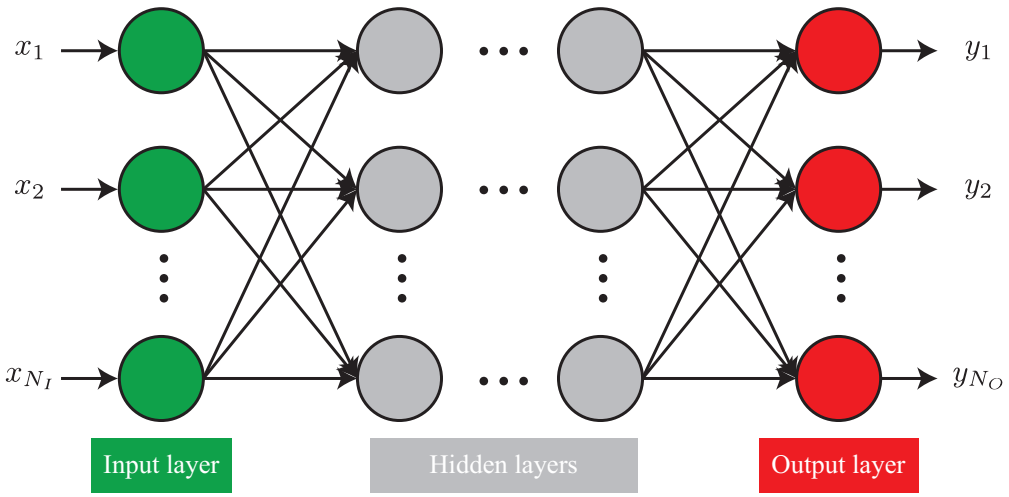


Figure 1.4: Illustration of an NN of artificial neurons.

optimal decision-making in the context of MOCU-based OED in Chapter 2. For a comprehensive description of NN models and their applications, see [19].

1.5 kernel ridge regression (KRR)

KRR is a fundamental ML algorithm based on linear least square with l_2-norm regularization in the kernel space. Formally, for given training data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, the KRR learns $\mathbf{w}$ that minimizes the cost function C_{KRR} of the KRR model with l_2-norm regularization defined as follows:

$$C_{\text{KRR}} = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \mathbf{w}^T\mathbf{\Phi}\left(\mathbf{x}_i\right)\right)^2 + \frac{1}{2}\lambda\left\|\mathbf{w}\right\|_2^2, \tag{1.5}$$

where y_i, $\mathbf{w}$, $\mathbf{\Phi}$, and λ are the response to $\mathbf{x}_i$, linear transformation matrix, kernel function, and regularization parameter, respectively.

In Chapter 4, we use a set of KRR models that learn different feature sets to construct an HTVS pipeline. For an exhaustive introduction to KRR and its applications, see [20].

1.6 Parametric density estimation

Parametric density estimation is a signal processing technique that estimates parameters of an underlying density from samples drawn from the distribution. First, we describe the maximum like-

lihood estimation (MLE) for normal distribution. Then, we discuss the expectation-maximization (EM) algorithm for a Gaussian mixture model that does not possess a closed-form expression of MLE due to the latent variables. Note that, for simplicity, we consider univariate distributions.

We utilize parametric density estimation techniques in order to learn the joint score distribution of the scores predicted via surrogate models in HTVS pipelines in Chapters 3 and 4. For more comprehensive details on mathematical derivations and parametric density estimation for multivariate distributions, see [21].

1.6.1 Maximum likelihood estimation (MLE)

Suppose that $X_1, X_2, \ldots, X_n$ are independent and identically distributed random variables following normal distribution $\mathcal{N}(\mu, \sigma^2)$. The likelihood function $L(\mu, \sigma)$ is defined as follows:

$$L(\mu, \sigma) = f_{X_1, X_2, \ldots, X_N}\left(x_1, x_2, \ldots, x_N; \mu, \sigma^2\right) \tag{1.6}$$

$$= \prod_{i=1}^{N} f_{X_i}\left(x_i; \mu, \sigma^2\right), \tag{1.7}$$

where $f_{X_i}(x_i; \mu, \sigma^2)$ is the probability density distribution (PDF) of normal distribution $\mathcal{N}(\mu, \sigma^2)$.

With realizations $x_1, x_2, \ldots, x_N$ of random variables $X_1, X_2, \ldots, X_N$, the key idea of MLE is to find parameters $\hat{\mu}_{\text{MLE}}$ and $\hat{\sigma}_{\text{MLE}}$ that maximize the log-likelihood function $\ln(L(\mu, \sigma^2))$ as follows:

$$\hat{\mu}_{\text{MLE}} = \arg\max_{\mu} \sum_{i=1}^{N} \ln\left(f_{X_i}\left(x_i; \mu, \sigma^2\right)\right), \tag{1.8}$$

$$\hat{\sigma}_{\text{MLE}} = \arg\max_{\sigma} \sum_{i=1}^{N} \ln\left(f_{X_i}\left(x_i; \mu, \sigma^2\right)\right). \tag{1.9}$$

As the log-likelihood function $\ln(L(\mu, \sigma^2))$ is concave, we can analytically find the ML parameters $\hat{\mu}_{\text{MLE}}$ and $\hat{\sigma}_{\text{MLE}}$ by taking the partial differentiation of $\ln(L(\mu, \sigma^2))$ with respect to μ and

σ and setting them to zero, respectively:

$$\hat{\mu}_{\text{MLE}} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{1.10}$$

$$\hat{\sigma}_{\text{MLE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \left(x_i - \hat{\mu}_{\text{MLE}}\right)^2}. \tag{1.11}$$

1.6.2 Expectation-maximization (EM) algorithm

Assume that $X_1, X_2, \ldots, X_N$ are independent and identically distributed random variables of a Gaussian mixture model with K components, where the joint PDF is defined as follows:

$$f_{X_1,X_2,\ldots,X_N}(x_1, x_2, \ldots, x_N) = \prod_{i=1}^{N}\sum_{k=1}^{K} \pi_k f_{X_i|Z_i}(x_i|z_i = k) \tag{1.12}$$

$$= \prod_{i=1}^{N}\sum_{k=1}^{K} \pi_k f_{X_i}\left(x_i; \mu_k, \sigma_k^2\right), \tag{1.13}$$

where π_k is a mixture weight of the kth component; Z_i is a latent variable representing the mixture component for X_i; and μ_k and σ_k^2 are the mean and variance of the kth component, respectively.

An EM algorithm finds the maximum likelihood estimates of the parameters for the Gaussian mixture model that depends on unobserved latent variable Z. The EM algorithm iteratively alternates an expectation step that computes the posterior distribution of Z_i given X_i, and a maximization step that estimates the optimal parameters based on the expectation. Specifically, the EM algorithm starts with initial parameters $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_K]$, $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_K]$, and $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \ldots, \sigma_K]$, and, based on these parameters, the posterior distribution is computed as follows:

$$P_{Z_i}(k|X_i = x_i) = \frac{\pi_k f_{X_i}(x_i; \mu_k, \sigma_k^2)}{\sum_{k=1}^{K} \pi_k f_{X_i}(x_i; \mu_k, \sigma_k^2)}. \tag{1.14}$$

Then, the EM algorithm updates the parameters based on the following expected complete

log-likelihood with respect to μ_k, σ_k, and, π_k:

$$E_{Z|X}\left[\ln f\left(X, Z; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}\right)\right] = \sum_{i=1}^{N}\sum_{k=1}^{K} P_{Z_i}\left(k|X_i = x_i\right)\left(\ln\left(\pi_k\right) + \ln\left(f_{X_i}\left(x_i; \mu_k, \sigma_k^2\right)\right)\right). \quad (1.15)$$

By alternating these steps, we can find the optimal parameters that maximize a lower bound on the log-likelihood function.

2. ACCELERATION OF BAYESIAN OPTIMAL EXPERIMENTAL DESIGN (OED)

Many real-world engineering applications involve mathematical modeling of complex systems, where the constructed models are used for designing operators–such as controllers, filters, classifiers, and estimators–that can effectively achieve engineering goals of interest. For example, one may be interested in building a network model representing the transcription regulations in micro-organisms that regulate their metabolism [22]. The resulting model may be used to infer the potential impacts of modifications in the transcription regulatory network (TRN) on the metabolism of interest, for example, predicting the metabolic flux changes that result from the deletion of one or more transcription factors. In this example, the engineering goal may be predicting the optimal genetic modification in the TRN that will lead to maximizing the production of a metabolite of interest. In fact, designing optimized strains of micro-organisms for ethanol overproduction [23] is an active area of research due to its implications for efficient bio-energy production.

A fundamental challenge in the aforementioned application as well as many other real-world engineering problems involving complex systems is the difficulty of accurate model construction. While one may have ample training data for model inference, the data size may nevertheless pale in comparison to the complexity of the system being modeled. Prior knowledge, if available, may also aid in improving model construction, but the final model is likely to still have substantial uncertainties. Consequently, a critical question is how one may reliably and optimally achieve the given engineering goals in the presence of model uncertainty. Furthermore, when one has the experimental budget for the acquisition of additional data or relevant knowledge (*e.g.*, via hypothesis testing), how should the experimental campaigns be designed to maximize the expected "return on investment"?

While these are fundamental problems in modern engineering with a long and rich history [24, 25], it has been recently shown that a novel Bayesian paradigm for objective-based uncertainty

quantification (objective-UQ) based on the mean objective cost of uncertainty (MOCU) [2, 26] can effectively address the optimal design of operators and experiments for complex uncertain systems [27, 28, 29, 30, 31, 32, 33, 34]. The core idea underlying the MOCU-based optimal experimental design (OED) is that, when dealing with complex uncertain models, one should quantify the model uncertainty in an objective-based manner and design experiments that can reduce the uncertainty that impacts one's operational goals. By focusing on the uncertainty that matters to the operation to be performed, the experimental budget can be efficiently used for optimizing the operational performance. To date, the efficacy of MOCU-based OED has been demonstrated in various systems, including experimental design for robust intervention in gene regulatory networks (GRNs) [29, 30] and that for robust synchronization of inter-coupled Kuramoto oscillators [27].

One practical challenge that limits the potential applicability of the MOCU-based OED scheme is its high computational cost, as discussed in [27, 29]. The computation of MOCU involves identifying the optimal robust operator for an uncertainty class that consists of all possible models (*e.g.*, models with different parameter values) as well as evaluating expectations based on high-dimensional prior (or posterior) probability distributions. Except for very simple cases, there is no closed-form expression for the optimal robust operator and the expectations have to be evaluated numerically [27]. As a result, the evaluation of MOCU involves costly optimization to find the optimal robust operator as well as extensive sampling of the uncertain model parameters from the uncertainty class to obtain reliable estimates, which may make the cost of MOCU computation formidably high in many applications.

In this chapter, we tackle this issue by adopting a machine learning (ML) approach for an efficient design of the optimal robust operator, thereby significantly accelerating the computation of MOCU as well as the MOCU-based experimental design. To the best of our knowledge, this is the first study that investigates adopting ML to accelerate MOCU-based OED. In order to develop and validate this ML-based OED acceleration scheme, we focus on designing experiments that can enhance the robust control of uncertain Kuramoto models that was investigated recently in [27]. A Kuramoto model [35] consists of a network of interconnected oscillators, whose dynamics are

described by coupled ordinary differential equations (ODEs). The Kuramoto oscillator model has been widely studied in various fields across engineering, physics, chemistry, and biology, due to its capability to model interesting collective behavior (*e.g.*, global/partial synchronization) that emerges in complex networks [36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47]. For example, a *microgrid* system with droop-controlled inverters can be mathematically cast as a Kuramoto model, where the synchronization failure of the model corresponds to a power outage in the microgrid [42, 44, 47, 48, 49, 50, 51]. Another interesting example is the application of the Kuramoto model for studying brain dynamics [38, 39, 45, 46], where the synchronization phenomena may be associated with neurodegenerative diseases [45, 52]. We show that our proposed ML-based OED acceleration scheme can improve the speed of MOCU-based experimental design by $104 \sim 154$ times without degrading the OED performance.

The two major contributions of this chapter are as follows. First, we propose an ML-based scheme for the acceleration of MOCU-based OED, which leads to significant speed improvement without performance degradation. Second, we present a comprehensive analysis of ML-based MOCU estimation and validate its performance in the context of OED.

2.1 Overview of Bayesian OED

In this section, we provide a brief review of the OED strategy for uncertain Kuramoto oscillator models, which we originally proposed in our recent work [27]. We begin the section with an introduction to the Kuramoto model, followed by a brief description of the robust synchronization problem for uncertain Kuramoto models. Given an uncertain Kuramoto model, we describe how the MOCU can be used to quantify the impact of the model uncertainty on the control synchronization performance and how the MOCU-based OED strategy can be used to effectively reduce the uncertainty that matters to the objective at hand–*i.e.*, optimal robust synchronization of the Kuramoto model in the presence of uncertainty.

Consider the Kuramoto model that consists of N interacting oscillators described by the fol-

lowing ODEs:

$$\dot{\theta}_i(t) = \omega_i + \sum_{j=1}^{N} a_{i,j} \sin\left(\theta_j(t) - \theta_i(t)\right), \tag{2.1}$$

for $i = 1, 2, \ldots, N$, where $\theta_i(t)$ is the instantaneous phase of the ith oscillator at time t, ω_i is the natural frequency of the ith oscillator, and $a_{i,j}$ is the coupling strength between the ith and jth oscillators. Kuramoto models have been widely studied to investigate the synchronization phenomena in various biological, chemical, or engineered oscillator systems, whose primary interest is whether the oscillators in a given Kuramoto model will get frequency synchronized as follows:

$$\lim_{t\to\infty} |\dot{\theta}_i(t) - \dot{\theta}_j(t)| = 0, \tag{2.2}$$

for $1 \leq i, j \leq N$. For example, it has been shown that modern smart grid networks referred to as microgrids can be modeled as a network of Kuramoto model oscillators, where the synchronization phenomena of the Kuramoto model are closely tied with the stability of the power grid network [48, 49, 50, 51]. Furthermore, in neuroscience studies, brain network synchronization has been shown to be associated with various neurological disorders, where excessive neuronal activities can be represented as a global synchronization of the Kuramoto model [38, 45, 46, 52, 53]. While conditions for synchronization have been extensively studied for homogeneous Kuramoto models with uniform coupling strength [54, 55, 56], there is yet no closed-form solution that can be used to predict the asymptotic synchronization of a general heterogeneous Kuramoto model based on its parameters.

In a real-world setting, the parameters of the Kuramoto model, which represents a complex network of oscillators, may not be completely known. For example, while it may be relatively easy to accurately estimate the natural frequency of each oscillator, in the absence of interactions with other oscillators, it will be practically challenging to accurately measure the coupling strengths between all oscillators in a large network. This uncertainty gives rise to an uncertainty class of Kuramoto models, which contains all possible Kuramoto models that are consistent with our prior knowledge regarding the true model and/or available observation data. Under this setting, our

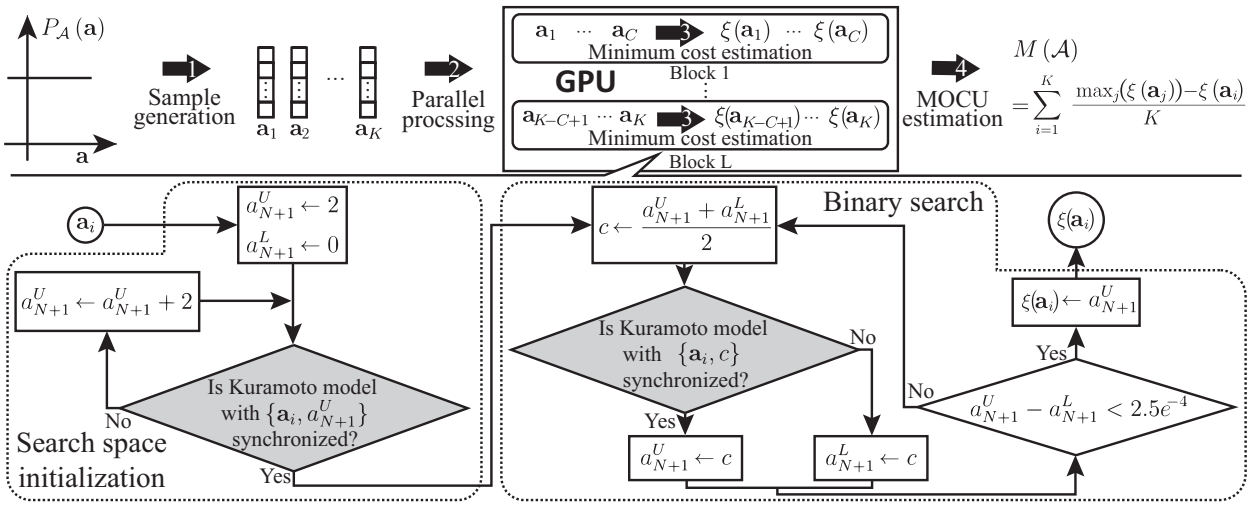


Figure 2.1: Illustration of the original sampling-based mean objective cost of uncertainty (MOCU) computation scheme in [27]. For reliable MOCU estimation, a relatively large sample size K is needed (step 1). The sampling-based estimation scheme takes advantage of graphics processing unit (GPU) programming for massive parallelization of the sampling operation. Specifically, we group the K sample points $\{\mathbf{a}_i\}$, $i = 1, 2, \ldots, K$, into L blocks (step 2), and the GPU processes L sample points in different processing blocks in parallel (step 3). Within block l, based on a sample point $\mathbf{a}_i$ that specifies a Kuramoto model (sampled from the uncertainty class), we find a valid search range $\left[a^L_{N+1}, a^U_{N+1}\right]$ that contains at least one valid solution that leads to global synchronization of the Kuramoto model (left bottom part). In the next phase (right bottom part), we find the solution with the smallest cost $\xi(\mathbf{a}_i)$ through a binary search, reducing the search range by half in every iteration. Finally, we compute the MOCU $M(\mathcal{A})$ of the uncertainty class $\mathcal{A}$ based on the K estimates $\xi(\mathbf{a}_i)$, $i = 1, 2, \ldots, K$, (step 4). © 2021 IEEE.

primary interest is how we can apply robust control to the uncertain Kuramoto model, comprised of a network of oscillators whose natural frequency ω_i is known but their coupling strength $a_{i,j}$ is only known up to a range $a_{i,j} \in \left[a^L_{i,j}, a^U_{i,j}\right]$. We denote the uncertainty class of all possible Kuramoto models as $\mathcal{A}$, which consists of all parameter vector $\mathbf{a} = [a_{1,2}, a_{1,3}, \ldots, a_{N-1,N}]^T \in \mathcal{A}$ that satisfies the given constraints. As in the previous study [27], we assume that prior distribution $P_{\mathcal{A}}(\mathbf{a})$ is uniformly distributed. However, this is not necessary. Non-uniform priors may be assumed, or custom priors may be constructed based on available prior domain knowledge [57, 58].

Suppose that we are interested in synchronizing an uncertain Kuramoto model that consists of N interacting oscillators, whose interaction strengths are only known up to a range, via external control. We adopt the synchronization method proposed in [27] that introduces an additional oscil-

lator as a global "synchronizer" to the original model. Let the natural frequency of this $(N+1)$th oscillator be $\omega_{N+1} = \frac{1}{N}\Sigma_{i=1}^{N}\omega_i$, and we assume that this control oscillator interacts with all oscillators in the original model with a uniform coupling strength $a_{i,N+1} = a_{N+1}, \forall i$, which is a control parameter. The addition of the control oscillator augments the Kuramoto model as follows:

$$\dot{\theta}_i(t) = \omega_i + \sum_{j=1}^{N} a_{i,j}\sin(\theta_j(t) - \theta_i(t)) + a_{N+1}\sin(\theta_{N+1}(t) - \theta_i(t)), \tag{2.3}$$

for $i = 1, 2, \ldots, N+1$. As the increase of the coupling strength a_{N+1} will in practice lead to an increase of the control cost, our control objective is to find a minimum a_{N+1} that guarantees the asymptotic frequency synchronization of the Kuramoto model despite the uncertainty. If we had complete knowledge about the coupling strength $\mathbf{a}$, we would be able to find the optimal (minimum) coupling strength $a_{N+1} = \xi(\mathbf{a})$ that ensures synchronization by gradually increasing the value of a_{N+1} from 0 until synchronization is achieved. A more efficient approach will be to perform a binary search as illustrated in Figure 2.1 (see the blow-up figure at the bottom). In the presence of uncertainty, we have to ensure that the control oscillator will be able to achieve synchronization for any $\mathbf{a} \in \mathcal{A}$. For this reason, we have chosen $a_{N+1} = \xi^*(\mathcal{A})$ as follows:

$$\xi^*(\mathcal{A}) = \max_{\mathbf{a}\in\mathcal{A}} \xi(\mathbf{a}), \tag{2.4}$$

which is the smallest a_{N+1} that guarantees global synchronization of the uncertain Kuramoto oscillators.

Given an uncertain Kuramoto model, the expected impact of this model uncertainty on the operational goal–in this case, the global frequency synchronization of the Kuramoto oscillators–can be quantified by the MOCU [2]. For a given uncertainty class $\mathcal{A}$, MOCU $M(\mathcal{A})$ can be computed by:

$$M(\mathcal{A}) = E_{\mathcal{A}}\left[\xi^*(\mathcal{A}) - \xi(\mathbf{a})\right], \tag{2.5}$$

where $\xi^*(\mathcal{A})$ is the cost of the optimal robust control and $\xi(\mathbf{a})$ is the cost of the optimal control

for the specific model configured with a specific parameter set $\mathbf{a}$. As shown in Equation (2.5), MOCU $M(\mathcal{A})$ quantifies the expected cost increase for applying the optimal robust control (which is inevitable to maintain robust control performance in the presence of uncertainty) instead of the model-specific optimal control (which cannot be applied in practice as the true model is unknown). In this study, the optimal robust interaction strength (cost) $\xi^*(\mathcal{A})$, ensuring that the uncertain Kuramoto model is synchronized by the added control oscillator while keeping the control cost minimum, is given by Equation (2.4).

In general, there is no closed-form expression of Equation (2.5), as a result of which the MOCU $M(\mathcal{A})$ for uncertainty class $\mathcal{A}$ computation requires a numerical approximation. One practical way to compute the MOCU $M(\mathcal{A})$ is to take a sampling-based approach to approximate it through the empirical expectation of the differential cost based on samples drawn from the distribution $P_{\mathcal{A}}(\mathbf{a})$.

Figure 2.1 illustrates the sampling-based MOCU computation process. First, we draw K sample points $\{\mathbf{a}_i\}$, $i = 1, 2, ..., K$, from $P_{\mathcal{A}}(\mathbf{a})$. Then, for each sample point $\mathbf{a}_i$, which is a potential true model parameter in the uncertainty class $\mathcal{A}$, we estimate the minimum coupling strength $\xi(\mathbf{a}_i)$ of the control oscillator that assures the asymptotic frequency synchronization of the Kuramoto model under control. To this aim, we consider a binary search to find the minimum coupling strength $\xi(\mathbf{a}_i)$ efficiently, as depicted in the dotted box at the bottom of Figure 2.1. Specifically, we start with a broad search space that contains at least one coupling strength synchronizing the system. At each iteration, we solve the ODEs of the Kuramoto model augmented with the control oscillator whose coupling strength a_{N+1} is set to the median value c of the current search space: $a_{N+1} \leftarrow c = \left(a_{N+1}^U + a_{N+1}^L\right)/2$. If the system under control is synchronized, we update the upper bound of the search space to the median value: $a_{N+1}^U \leftarrow c$. Otherwise, we set the lower bound of the search space to the median value: $a_{N+1}^L \leftarrow c$. The binary search continues until we find the minimum coupling strength $\xi(\mathbf{a}_i)$, for the given sample point $\mathbf{a}_i$, which is within a specified tolerance level (set to 2.5×10^{-4} in this study). Based on the K sample points, we can obtain the

MOCU $M(\mathcal{A})$ as follows:

$$M(\mathcal{A}) = \frac{1}{K}\sum_{i=1}^{K}\left(\max_{j}\left(\xi\left(\mathbf{a}_j\right)\right) - \xi\left(\mathbf{a}_i\right)\right). \tag{2.6}$$

Note that the accuracy of this numerical approximation of MOCU is dependent on the sample size K. In general, a larger K generally leads to a more accurate MOCU estimation. However, at the same time, the computational cost increases as the sample size increases. We can reduce the computational time for numerical MOCU computation by exploiting parallelism. For example, estimating the optimal cost $\xi(\mathbf{a}_i)$ of a sample point $\mathbf{a}_i$ is an independent process to those of the other samples $\mathbf{a}_j$, $j \neq i$, which can be processed in a parallel manner with powerful parallel processors. In fact, the sampling-based MOCU computation in [27] takes advantage of GPU (Graphics Processing Unit) programming with CUDA (Compute Unified Device Architecture), in which 200 sample points are processed in parallel at a given time–*i.e.*, $L = 200$. However, for each sample point $\mathbf{a}_i$, the estimation of the minimum cost $\xi(\mathbf{a}_i)$ via binary search (step 3 in Figure 2.1) is a highly sequential process–which involves repeatedly solving the ODEs of the corresponding Kuramoto model and verifying whether or not the model is globally synchronized (*i.e.*, not amenable to parallelization).

The significance of objective-UQ using MOCU is that it enables the design of experiments that focus on reducing the model uncertainty that matters. More specifically, as MOCU quantifies the expected cost increase (relevant to our operational goal) due to model uncertainty, it can be used to quantify the expected impact of a potential experiment on reducing the model uncertainty that affects the operational performance, hence how effective the experiment will be in reducing the operational cost.

The MOCU-based OED strategy for uncertain Kuramoto models has been recently proposed in [27]. In this study, a realistic experimental design space was considered, where an experiment corresponds to selecting a pair (i, j) of oscillators and observing whether they get spontaneously synchronized in isolation of other oscillators and in the absence of external control. The experi-

mental outcome was a binary value–either synchronized or non-synchronized–based on which the uncertainty of the coupling strength $a_{i,j} \in [a_{i,j}^L, a_{i,j}^U]$ can be reduced. Theorem 1 in [27] reproduced below gives us the necessary and sufficient condition for an oscillator pair to be frequency synchronized (see Appendix A for the proof):

Theorem 1. *Consider the Kuramoto model of two-oscillators:*

$$\begin{aligned} \dot{\theta}_1(t) &= \omega_1 + 0.5a\sin(\theta_2(t) - \theta_1(t)), \\ \dot{\theta}_2(t) &= \omega_2 + 0.5a\sin(\theta_1(t) - \theta_2(t)), \end{aligned} \tag{2.7}$$

with the initial angles $\theta_1(0), \theta_2(0) \in [0, 2\pi)$. *Then, for any solutions* $\theta_1(t)$ *and* $\theta_2(t)$ *to* (2.7)*, there holds* $|\dot{\theta}_1(t) - \dot{\theta}_2(t)| \to 0$ *as* $t \to \infty$ *if and only if* $|\omega_1 - \omega_2| \leq a$. ■

According to Theorem 1, the Kuramoto oscillator pair (i, j) becomes frequency synchronized $\lim_{t\to\infty}|\dot{\theta}_i(t) - \dot{\theta}_j(t)| = 0$ if and only if $\frac{|\omega_i - \omega_j|}{2} \leq a_{i,j}$. As a result, if the two oscillators are observed to be synchronized, we can increase the lower bound $a_{i,j}^L$ to $\max\left(a_{i,j}^L, |\omega_i - \omega_j|/2\right)$. Otherwise, we can decrease the upper bound $a_{i,j}^U$ to $\min\left(a_{i,j}^U, |\omega_i - \omega_j|/2\right)$. Since the experimental outcome is unknown in advance, we need to consider both possible outcomes to quantify the expected impact of a given experiment on reducing the objective uncertainty. To formalize this, let $O_{i,j}$ be a binary random variable representing the outcome of the pairwise synchronization experiment for the oscillator pair (i, j). Then, the expected remaining MOCU $R(i, j)$ is given by:

$$\begin{aligned} R(i,j) &= E_{O_{i,j}}[M(\mathcal{A}|O_{i,j})] \\ &= \sum_{o\in\{0,1\}} P(O_{i,j} = o)\, M(\mathcal{A}|O_{i,j} = o), \end{aligned} \tag{2.8}$$

where $M(\mathcal{A}|O_{i,j})$ is the conditional MOCU given $O_{i,j}$. The conditional MOCU $M(\mathcal{A}|O_{i,j} = o)$ given an experimental outcome $O_{i,j} = o$ can be computed by reducing the uncertainty class as previously described and numerically computing the MOCU of this reduced uncertainty class.

The probability $P\left(O_{i,j}=o\right)$ can be derived in a straightforward manner as follows:

$$P\left(O_{i,j}=1\right)=\frac{a_{i,j}^{U}-\hat{a}_{i,j}}{a_{i,j}^{U}-a_{i,j}^{L}}, \tag{2.9}$$

$$P\left(O_{i,j}=0\right)=\frac{\hat{a}_{i,j}-a_{i,j}^{l}}{a_{i,j}^{U}-a_{i,j}^{L}}, \tag{2.10}$$

where, $\hat{a}_{i,j}=\min\left(\max\left(\frac{1}{2}\left|\omega_i-\omega_j\right|, a_{i,j}^{L}\right), a_{i,j}^{U}\right)$. The $R\left(i,j\right)$ in Equation (2.8) quantifies the MOCU that is expected to remain after performing the pairwise synchronization experiment for the pair (i,j).

So, how should we prioritize the potential $\binom{N}{2}$ experiments? Naturally, the optimal choice will be to choose the experiment with the smallest $R\left(i,j\right)$ as follows.

$$\left(i^{*},j^{*}\right)=\underset{(i,j)\in\mathcal{E}}{\arg\min}\, R\left(i,j\right). \tag{2.11}$$

Experiment (i^{*},j^{*}) is expected to most effectively reduce the objective uncertainty among all potential experiments. In practice, rather than performing a single best experiment, we may perform a sequence of experiments prioritized by Equation (2.11). In theory, $R\left(i,j\right)$ needs to be re-estimated after performing the predicted optimal experiment and observing its outcome, as it changes the uncertainty class, hence the expected remaining MOCU for the potential subsequent experiments. However, empirically, $R\left(i,j\right)$ computed based on the original uncertainty class $\mathcal{A}$ is a robust indicator of the efficacy of the potential experiments.

The overall computational complexity for predicting the optimal experiment is as follows:

$$O(TKN^{4}L^{-1}\log\epsilon), \tag{2.12}$$

where T is the time duration for solving the ODEs using the Runge-Kutta method (to check for asymptotic global frequency synchronization among the Kuramoto model oscillators), K is the sample size for numerical computation of MOCU, N is the number of oscillators in the Kuramoto model, L is the number of parallel processing blocks in GPU, and ϵ is the tolerance level for the

binary search (set to $\epsilon = 2.5 \times 10^{-4}$ in this study). Note that the complexity for computing MOCU is $O(TKN^2L^{-1}\log\epsilon)$, where predicting the optimal experiment involves computing MOCU $2 \cdot \binom{N}{2}$ times to calculate $R(i,j)$ given by Equation (2.8) for all oscillators pairs. As we can see in Equation (2.12), the computational cost for the sampling-based OED sharply increases as the size N of the Kuramoto model increases, which limits the practical applicability of the OED scheme for large models. For example, when $T = 5$, $K = 20,480$, and $L = 128$, respectively, identifying the optimal experiment (i^*, j^*) for the uncertain Kuramoto model operating on five oscillators required 650 seconds on average. However, it took $3,171$ seconds to determine the optimal experiment (i^*, j^*) for the uncertain Kuramoto model with seven oscillators.

2.2 Methods

We propose an ML approach for accelerating the quantification of the objective system uncertainty. As we discussed in the previous section, in real-world applications that typically involve the control consisting of highly non-linear sequential operations, the effective computational complexity is critically dependent on the computational complexity of the control rather than the number of samples. The proposed approach learns a surrogate model for (part of) the operations of the control for estimation of the control cost for a system, thereby reducing the effective computational complexity that cannot be further reduced by parallelism. Recently, there has been an increasing number of studies investigating the application of deep learning (DL) methods to scientific computation, including approximating and solving differential equations (DEs) (*e.g.*, see [59, 60, 61] and references therein). However, it is worth noting that the primary focus of our current study does not lie in solving ODE systems via deep network models but in the accelerated design of optimal experiments based on the objective-UQ via the concept of MOCU. Rather than aiming at a fast solution of DEs, our goal is to efficiently design experiments that can most effectively reduce model uncertainty, thereby optimally enhancing the control performance of uncertain Kuramoto oscillator models.

Estimating the MOCU of the uncertain Kuramoto model based on the sampling approach involves a binary search for each sample $\mathbf{a}_i$, where at each iteration solving the corresponding ODEs

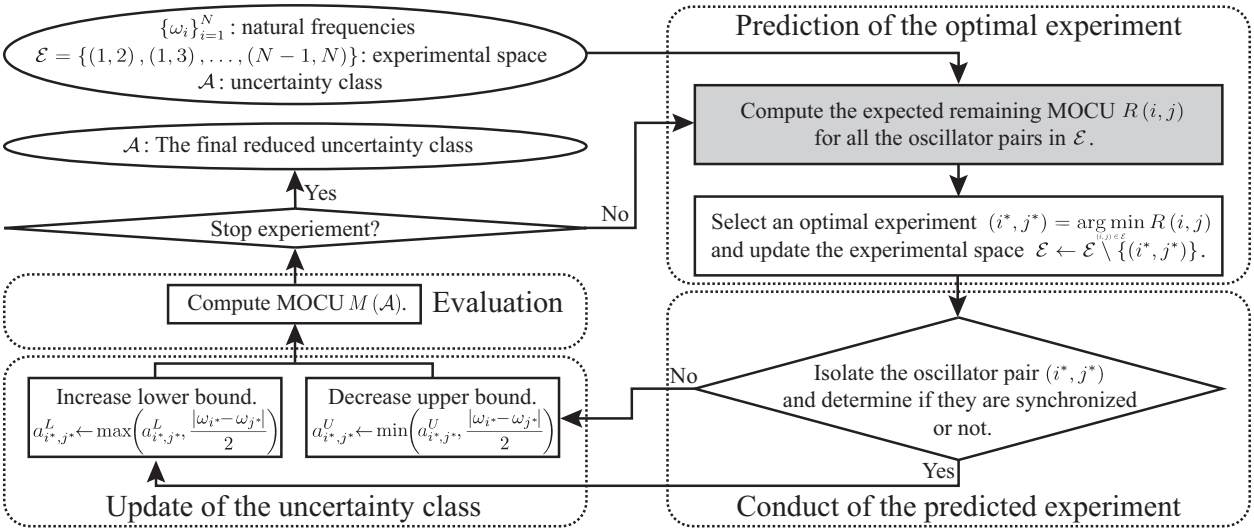


Figure 2.2: Illustration of the MOCU-based OED loop. First, we compute the expected remaining MOCU $R(i,j)$ for all possible experiments (i,j) in the experimental design space $\mathcal{E}$ based on the current uncertainty class $\mathcal{A}$. Next, we identify the optimal experiment (i^*,j^*) that has the smallest expected remaining MOCU such that $(i^*,j^*) = \arg\min_{(i,j)\in\mathcal{E}} R(i,j)$. In the second phase (right bottom), we conduct the selected experiment (i^*,j^*) and remove the performed experiment from the experimental space $\mathcal{E}$. Specifically, in this experiment, we isolate the selected oscillator pair (i^*,j^*) and determine whether or not they get synchronized without external control. Based on the experimental outcome, we update the uncertainty class accordingly [27]. Finally, we evaluate the actual efficacy of the conducted experiment by computing the MOCU of the updated uncertainty class $\mathcal{A}$. We iterate this experimental loop until the experimental space becomes empty (*i.e.*, there are no more experiments left to be performed). © 2021 IEEE.

and determining if the system under control is synchronized or not. From a broad perspective, at each iteration, these operations, the gray box in Figure 2.1, are nothing but a binary classification problem. Hence, if we collect enough samples to build an accurate classifier, we replace such a process with the binary classifier, which is computationally efficient. In this study, we considered a fully-connected neural network (fcNN) with only one hidden layer, possibly the simplest ML structure that we can think of.

The proposed approach on the MOCU-based OED framework is realized by replacing part of the operations of control with the trained model for the estimation of the expected remaining MOCU $R(i,j)$ highlighted in gray in Figure 2.2. Hence we focus on the difference in quantifying the expected remaining MOCU $R(i,j)$ between the proposed ML-based approach and the original

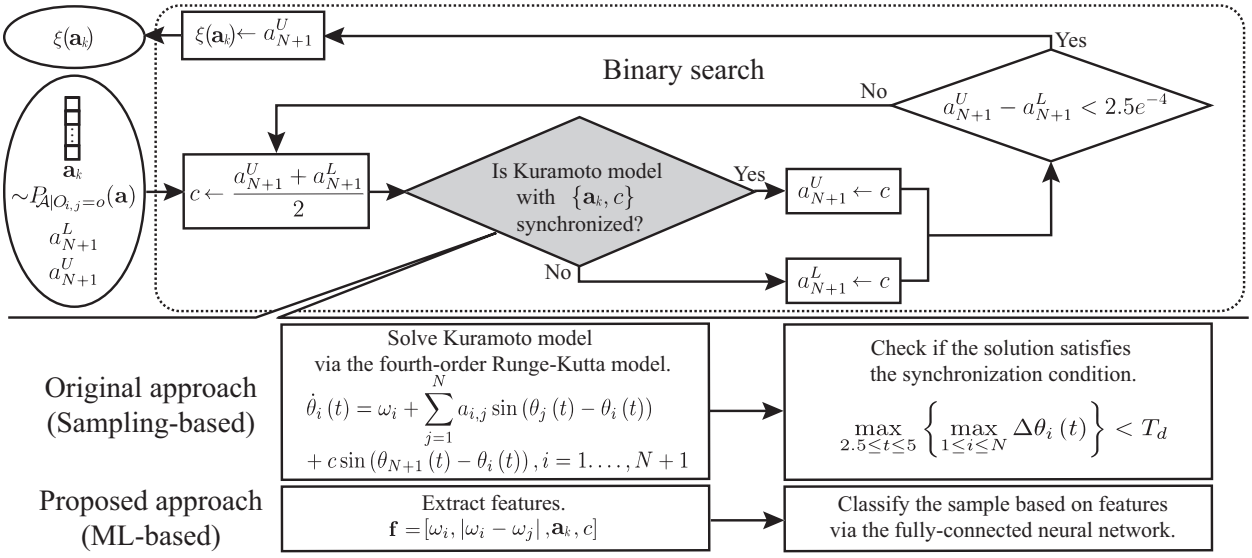


Figure 2.3: Comparison between the original sampling-based estimation scheme adopted in [27] and the proposed machine learning (ML)-based estimation scheme. The proposed scheme obviates the need for repeatedly solving the coupled ordinary differential equations (ODEs) within the binary search routine to find the optimal robust coupling strength illustrated in Figure 2.1. This significantly enhances the computational efficiency of MOCU estimation. © 2021 IEEE.

approach that manually determines the synchronization of the Kuramoto model. To compute the expected remaining MOCU $R(i,j)$ we first need to estimate conditional MOCU $M(\mathcal{A}|O_{i,j} = o)$ given the experimental outcome $O_{i,j} = o \in \{0, 1\}$ (*i.e.*, synchronized or not) as derived in Equation (2.8). Specifically, we compute the control cost $\xi(\mathbf{a}_i)$ of all samples $\mathbf{a}_i$, $i = 1, 2, \ldots, K$, drawn from the posterior uncertainty class distribution $P_{\mathcal{A}|O_{i,j}=o}(\mathbf{a})$ updated according to the experimental outcome $O_{i,j} = o$ as shown in Figure 2.2. Figure 2.3 shows the difference in estimating the control cost $\xi(\mathbf{a}_i)$ of sample $\mathbf{a}_k$ between the proposed approach and the original approach. Both approaches find a numerical solution through the binary search that is a sequential process. At each iteration, the coupling strength a_{N+1} of the control oscillator is set to the midpoint $c \leftarrow \left(a^U_{N+1} + a^L_{N+1}\right)/2$ of the search space. The original approach solves the Kuramoto model determined by the sample $\mathbf{a}_k$ and midpoint c and determines if the solutions are synchronized or not according to the criterion (Equation (2.2)). On the other hand, the proposed ML-based approach extracts features based on the natural frequencies ω_i, sample $\mathbf{a}_k$, and midpoint c and classifies the

feature vector. The search space is then halved according to the outcome. Note that the computational complexity of the original approach is critically dependent on the time precision and simulation time. Less time precision and shorter simulation time can reduce the overall computational complexity, but such parameters significantly affect the estimation accuracy of the MOCU. On the other hand, MOCU-based OED with the proposed approach is free from such a trade-off at the inference phase as the features are independent of the parameters.

2.3 Results and discussion

In this section, we demonstrate the efficacy of the proposed ML approach in accelerating the speed of objective-UQ, resulting in a very efficient OED. As described in the previous section, we considered the OED for the Kuramoto model under uncertainty, where one's operational objective is to ensure synchronization of the model by adding an oscillator for control. For validation, we considered two experimental setups based on uncertain Kuramoto models with five and seven oscillators, respectively. As a reference ODE solver, we used the fourth-order Runge-Kutta method to solve the Kuramoto model sampled at the sampling frequency f_s of 160Hz for five seconds. To determine whether the Kuramoto model is synchronized or not, we used the following criterion:

$$\max_{2.5\leq t\leq 5}\left(\max_{1\leq i\leq N}\Delta\theta_i\left(t\right)\right) < T_d, \tag{2.13}$$

where $\Delta\theta_i\left(t\right) \triangleq \theta_i\left(t+\left(1/f_s\right)\right) - \theta_i\left(t\right)$, $\theta_i\left(t\right)$ is the instantaneous phase of the ith oscillator, and T_d is a threshold of tolerance. We set T_d to 0.001. To estimate the MOCU of a given uncertainty class, we randomly drew $20,480$ sample points from the uncertainty class (*i.e.*, $K = 20,480$). We used a *Lambda workstation* equipped with *Intel i*9-9960*X*, 128GB memory, and *GeForce RTX* 2080 *Ti* for the simulations.

At the core of the proposed method lies a binary classifier that accurately classifies the global frequency synchronization of the model when a control oscillator is introduced. To train an accurate classifier, we used an fcNN model with one hidden layer. In that regard, it is essential to extract representative features from the parameters that define the Kuramoto model, such as the

number of oscillators, natural frequencies, initial phases, or coupling strength values between oscillators. Inspired by Theorem 1, which gives us the necessary and sufficient condition for pairwise frequency synchronization of Kuramoto oscillators, we used the natural frequencies, the absolute difference between the frequencies, and the corresponding coupling strength values as features. More specifically, given a parameter set that fully determines the Kuramoto model operating on $N+1$ oscillators, we first sort all the natural frequencies in descending order and rearrange the coupling strength accordingly. Then, we construct the corresponding feature set that consists of the sorted natural frequencies, the absolute difference of the natural frequencies of all oscillator pairs, and their coupling strengths. Note that this arrangement makes the feature set highly structured but does not affect the characteristics of the Kuramoto model. To accurately label a given sample point (the feature set of a given Kuramoto model), we used the fourth-order Runge-Kutta method with a much longer simulation time T of 400 seconds to determine whether the model reaches global frequency synchronization or not. Besides, we rigorously determined the synchronization of the Kuramoto model based on more stringent criteria. For the labeling purpose, we consider that a Kuramoto model is synchronized if both of the following two conditions are satisfied: First, frequencies of all oscillators rounded to the sixth decimal place are equal for the last 20 $(T * 0.95)$ seconds. Second, the sum of absolute change in the coherence value $r\,(t)$ of the order parameter $r\,(t)\,e^{j\psi(t)} = \frac{1}{N}\sum_{i=1}^{N} e^{j\theta_i(t)}$ is less than 10^{-6} for the last 20 seconds. Note that if the results for the two conditions differ, we excluded the sample point from the training dataset.

First, in order to validate the efficacy of the proposed method that incorporates ML-based predictions into MOCU estimation, we directly compared the MOCU values from the ML-based approach and the sampling-based approach.

As a first experimental scenario, we considered an uncertain Kuramoto model that consists of five oscillators that do not get spontaneously synchronized in the absence of external control. In this experiment, we adopted the identical experimental setup in the previous work [27] for direct comparison. Specifically, we assumed that the five oscillators have the natural frequencies of -2.50, -0.6667, 1.1667, 2.0, and 5.8333, respectively. The natural frequency of the additional

(*i.e.*, 6th) control oscillator was set to the average frequency of the five oscillators ($\omega_6 = 1.1667$). Besides, we set the initial phase of all the oscillators to zero. Finally, we used the uncertainty class defined as follows:

$$\mathbf{a}^U = \begin{bmatrix} 1.0541 & 0.6325 & 0.7762 & 1.4375 & 1.0542 & 0.6900 & 1.6819 & 0.4791 & 2.6833 & 2.2041 \end{bmatrix}^T, \tag{2.14}$$

$$\mathbf{a}^L = \begin{bmatrix} 0.7791 & 0.4675 & 0.5737 & 1.0625 & 0.7792 & 0.5100 & 1.2431 & 0.3541 & 1.9833 & 1.6291 \end{bmatrix}^T. \tag{2.15}$$

To train the classifier, we generated $40,000$ sample points (a set of $20,000$ parameter values that result in synchronization and another set of $20,000$ parameter values that do not) from a multivariate uniform distribution whose support completely covers the range of the parameters in the uncertainty class at hand. Specifically, a parameter set has six real-values from the uniform distribution with a range of $(-2\pi, 2\pi)$ as natural frequencies of the six oscillators ω_i, $i = 1, 2, \ldots, 6$, and ten coupling strength values $a_{i,j}$, $1 \leq i < j \leq 6$, between oscillators ranging from $0.25\,|\omega_i - \omega_j|$ to $2.35\,|\omega_i - \omega_j|$. To build the classifier, we sorted the six natural frequencies in descending order and rearranged the coupling strength values accordingly. Then, we extracted the following features: the sorted natural frequencies, the absolute difference of the natural frequencies of all oscillator pairs, and their coupling strengths. Finally, we trained an fcNN model with a single hidden layer, whose width is three times the number of features, until the model is capable of classifying all the $40,000$ sample points in the training dataset perfectly. We validated the trained model in terms of its asymptotic classification accuracy by assessing the accuracy as a function of the training data size. This result is shown in Figure B.1 in Appendix B.

We started with the original uncertainty class defined in Equations (2.14) and (2.15) and estimated the expected remaining MOCU of random oscillator pairs through both approaches one hundred times while randomly changing the true model (assumed to be unknown). Figure 2.4 is a scatter plot that shows the comparison between the expected remaining MOCU values computed by different methods. As shown in Figure 2.4, the expected remaining MOCU values computed by the proposed ML-based method and the original sampling-based method display a strong linear relationship. The Pearson's correlation coefficient was 0.9849 with a p-value of 1.90×10^{-76}.

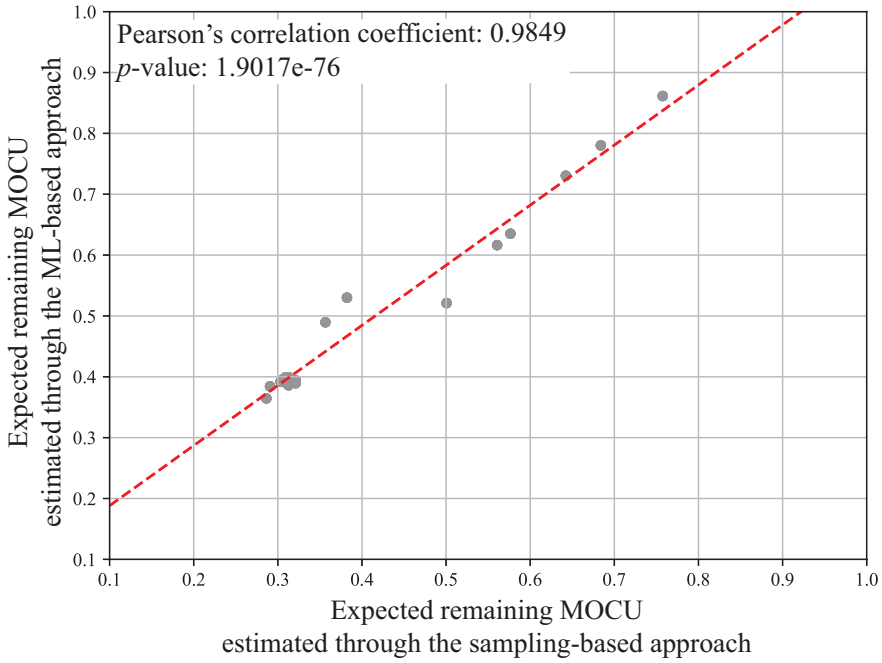


Figure 2.4: The scatter plot shows the expected remaining MOCU values for the uncertain five-oscillator Kuramoto model estimated using the proposed ML-based approach and the original sampling-based approach in [27]. As shown, the estimated values were highly correlated to each other. © 2021 IEEE.

This plot shows that the ML-based computational scheme has the potential to effectively replace the costly sampling-based scheme without affecting the MOCU-based OED performance, as it will likely not affect the ranking of potential experiments. In terms of computational cost, the ML-based approach was able to compute the expected remaining MOCU in 0.1110 seconds (on average) for a given uncertainty class, while it took 818.7 seconds (on average) for the sampling-based approach. These results clearly show the advantages of the proposed approach in efficiently quantifying the objective uncertainty.

To examine the computational cost increase and scalability for larger models, we next considered an uncertainty class of Kuramoto models with seven oscillators. This increases the time for solving the ODEs and the number of possible experiments also increases from $\binom{5}{2} = 10$ to $\binom{7}{2} = 21$. Here we set the natural frequency of the oscillators to -3.4600, -1.9611, -0.6754, -0.3806, -0.3675, 6.1161, and 8.3287, respectively. We assumed that the natural frequency

of the control oscillator (*i.e.*, 8th oscillator) is the average frequency of the seven oscillators ($\omega_8 = 1.0857$). We considered the uncertainty class shown below:

$$\mathbf{a}^U = \begin{bmatrix} 0.848 & 0.988 & 1.446 & 1.607 & 3.820 & 0.915 & 0.400 \\ 0.850 & 0.419 & 4.162 & 1.090 & 0.122 & 0.039 & 2.124 \\ 0.872 & 0.007 & 2.737 & 1.804 & 1.360 & 0.744 & 1.174 \end{bmatrix}^T, \tag{2.16}$$

$$\mathbf{a}^L = \begin{bmatrix} 0.073 & 0.172 & 0.153 & 0.054 & 0.501 & 0.463 & 0.043 \\ 0.015 & 0.096 & 0.501 & 0.103 & 0.007 & 0.009 & 0.139 \\ 0.408 & 0.000 & 0.131 & 0.119 & 0.300 & 0.286 & 0.131 \end{bmatrix}^T. \tag{2.17}$$

As in the previous experiment for the Kuramoto model with five oscillators, we set the initial phase of all oscillators to zero.

As the size of the parameter set is much greater for this Kuramoto model, we generated the training data in a more tailored way. Rather than generating the sample points (*i.e.*, Kuramoto model parameter sets) with random natural frequencies within a specific range as we did for the five oscillator model, we fixed the natural frequencies to -3.4600, -1.9611, -0.6754, -0.3806, -0.3675, 6.1161, 8.3287, and 1.0857 in this example. For the coupling strength values, we drew them from the uniform distribution for the uncertainty class, whose support is defined in (2.16) and (2.17). In this manner, we collected $50,000$ sample points per label according to the same criteria we used for the five oscillator case. Then, we extracted the feature values as described previously for the five oscillator case and trained the classifier using an fcNN with a single hidden layer, whose width is four times the number of features. Figure B.1 in Appendix B shows that this model quickly learns the classification boundary, where the classification accuracy rapidly converges to 100% as the size of the training data increases.

As before, we started with the original uncertainty class defined in Equations (2.16) and (2.17) and computed the expected remaining MOCU of random oscillator pairs using the ML-based method and the sampling-based method. We repeated this until we collected a hundred expected remaining MOCU values per method. Figure 2.5 shows the scatter plot that compares the expected

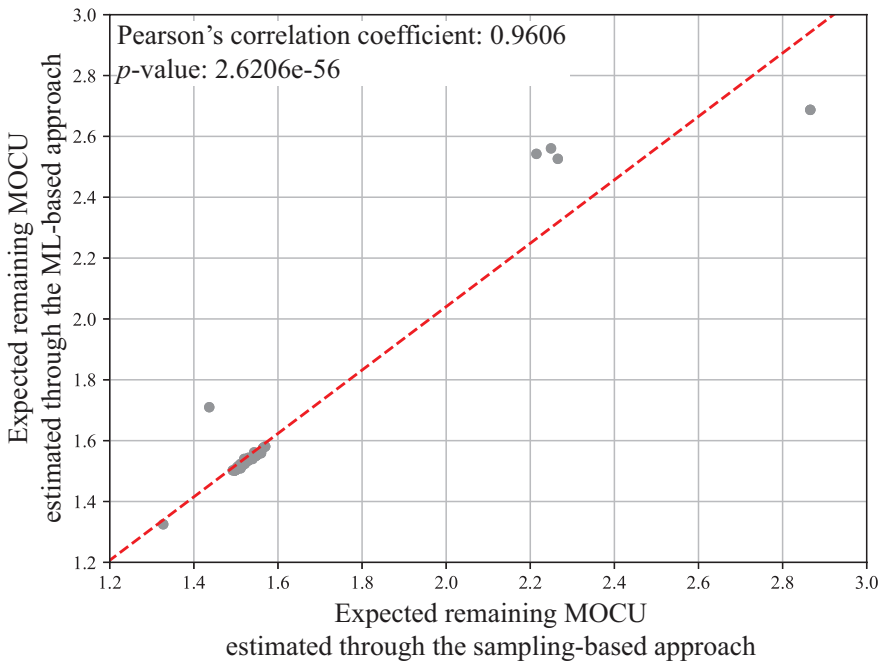


Figure 2.5: The scatter plot shows the expected remaining MOCU values for the uncertain seven-oscillator Kuramoto model estimated using the proposed ML-based approach and the original sampling-based approach in [27]. As before, the estimated values showed a high correlation. © 2021 IEEE.

remaining MOCU values computed by the two methods. Again, we can see that there is a strong linear relationship between the computed values. The Person's correlation coefficient was 0.9606 with a p-value of 2.62×10^{-56}. In terms of computational cost, it took 0.6953 seconds (on average) for the ML-based method to compute the expected remaining MOCU, which was still less than a second although the experimental design space has grown from $\frac{(5\times4)}{2} = 10$ experiments to $\frac{(7\times6)}{2} = 21$. It took the sampling-based approach $3,684.9$ seconds (on average) to compute the expected remaining MOCU values, which shows that our proposed method makes the computation $5,298$ times faster at practically identical accuracy. These results clearly show the advantages of the proposed ML-based approach in quantifying the objective model uncertainty.

Next, we compared the OED performance of the proposed ML-based method against three existing approaches:

- **Sampling-based approach**: the original approach proposed in [27] based on the MOCU

framework, where a fourth-order Runge-Kutta method is to solve the Kuramoto model to determine synchronization.

- **Entropy-based approach**: the experiment is chosen for the oscillator pair whose coupling strength value has the largest entropy to reduce this uncertainty.
- **Random approach**: the experiment is randomly selected from the experimental design space.

For the MOCU-based OED schemes (*i.e.*, ML-based and sampling-based computations), we considered the following OED strategies. In the first approach (marked as *iterative* in the figures), we re-estimated the expected remaining MOCU for the remaining experiments in each iteration, after performing the predicted optimal experiment and updating the uncertainty class based on the observed experimental outcome. In the second approach, we estimated the expected remaining MOCU only based on the initial uncertainty class and prioritized all experiments based on this result. While this approach is theoretically suboptimal, it significantly reduced the overall computational cost and empirically showed comparable performance to the iterative scheme, as we will show in this section. Note that we reused the fcNN models trained for the MOCU value comparisons.

We conducted OED simulations for the same five-oscillator Kuramoto model considered in the previous study [27] for direct comparison. The true (unknown) model $\mathbf{a}$ was assumed to be as follows:

$$\mathbf{a} = \begin{bmatrix} 0.9166 & 0.55 & 0.675 & 1.25 & 0.9167 & 0.6 & 1.4625 & 0.4166 & 2.3333 & 1.9166 \end{bmatrix}^T. \tag{2.18}$$

Figure 2.6 shows the experimental design performance of the different algorithms, where the objective uncertainty (quantified by MOCU) is shown as a function of the number of experimental updates (iterations). As shown in Figure 2.6, the proposed ML-based approach with iterative re-estimation (red dotted line with asterisks) showed the nearly identical performance to sampling-based methods (both iterative and non-iterative schemes, shown in yellow lines). All three schemes

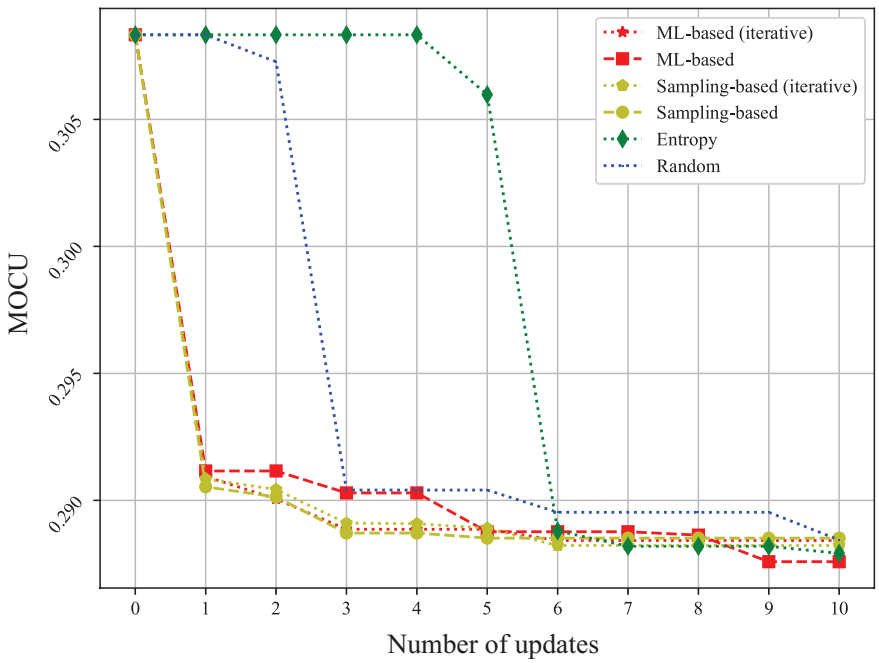


Figure 2.6: Performance comparison of various experimental design strategies for the uncertain five-oscillator Kuramoto model considered in [27]. The results showed that the three MOCU-based OED schemes perform similarly, regardless of how MOCU was estimated. The MOCU-based schemes clearly outperformed other schemes as reported in [27]. © 2021 IEEE.

reached the near minimum MOCU within only three experimental updates. The non-iterative ML-based scheme (red dashed line with squares) also identified the first optimal experiment accurately and showed comparable performance in the later updates with the other three MOCU-based OED schemes. All four MOCU-based OED schemes (both ML-based and sampling-based) significantly outperformed the entropy-based and random approaches, resulting in much sharper uncertainty reduction within fewer experimental updates.

Figure 2.7 compares the overall computational cost between the ML-based OED schemes and the sampling-based OED schemes. The entropy-based approach and the random approach are not shown, as their computational cost is fixed and negligible. As we can see in Figure 2.7, the proposed ML-based OED approaches, marked as red, showed significantly lower time complexity compared to the sampling-based OED approaches. Note that the ML-based methods (red dotted lines) were significantly faster compared to the sampling-based methods, despite maintaining

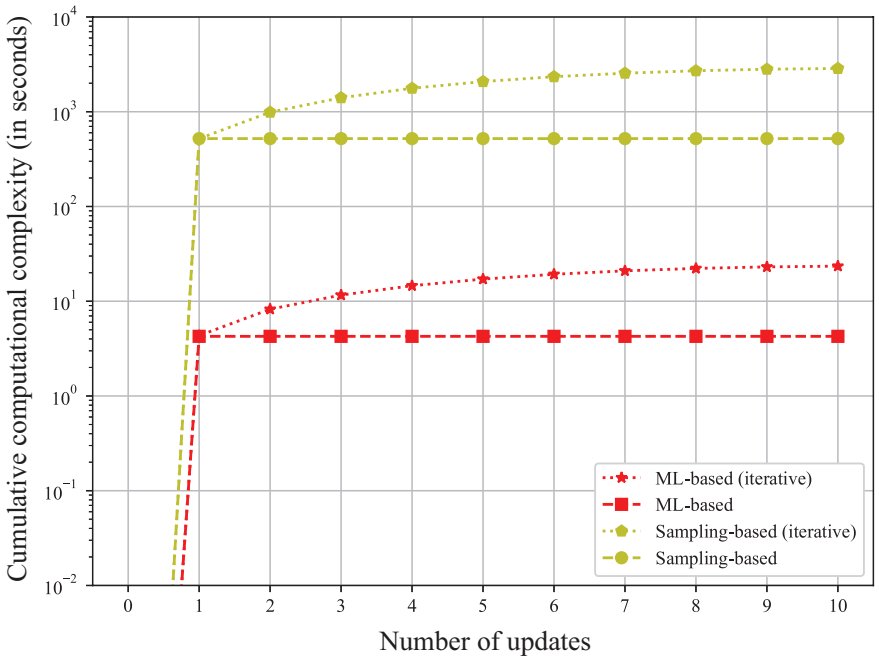


Figure 2.7: Cumulative computational cost (in seconds) for identifying the optimal experiment. As shown, the proposed ML-based estimation clearly outperformed the original sampling-based estimation [27] in terms of efficiency, where their costs differed by two orders of magnitude. For both ML-based/sampling-based schemes, iterative estimation required further computations, as the uncertainty class is updated after each experiment, based on which the remaining expected MOCU values are assessed again. © 2021 IEEE.

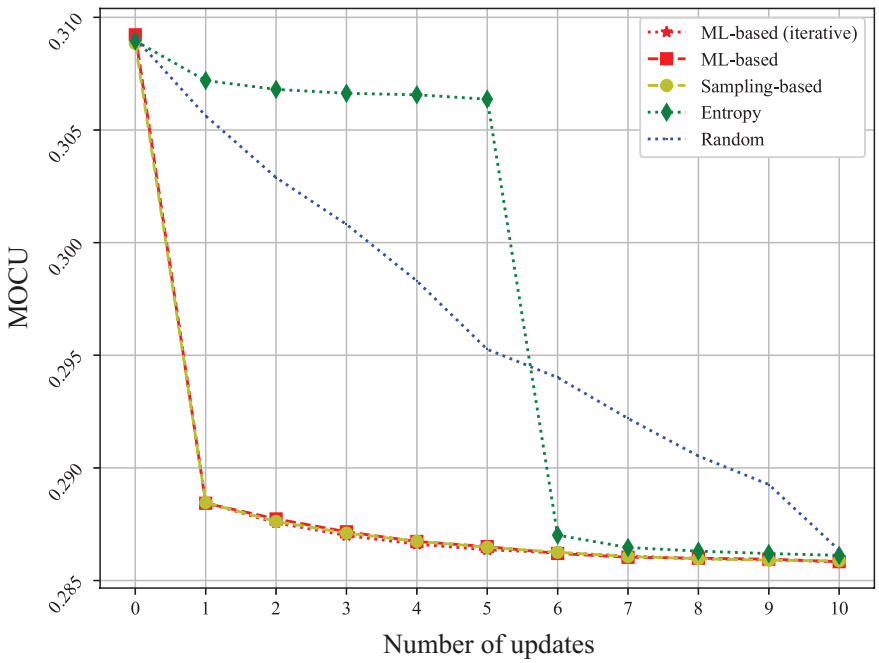


Figure 2.8: Average performance of various experimental design strategies for uncertain Kuramoto models with five oscillators. The experiments were repeated one hundred times by sampling potential true models from the uncertainty class. As shown, all three MOCU-based methods led to the best performance. Random selection resulted in linear uncertainty reduction as expected. © 2021 IEEE.

equivalent OED performance. We did not include the time for training the fcNN model as the model training only needs to be performed once before the beginning of OED. In fact, the trained model can be reused for different uncertainty classes and *any* true model therein. Besides, the training time is negligible thanks to the shallow structure of the fcNN model considered in this study. Specifically, the model learned the training dataset for making a decision for the uncertain Kuramoto model with five oscillators within 90 seconds.

Next, we repeated the experiment based on one hundred different true models randomly drawn from the uncertainty class (*i.e.*, different coupling strength values were drawn from the prior distribution of the uncertainty class). The results of these large-scale experiments are shown in Figure 2.8 and Figure 2.9. Note that we excluded the iterative sampling-based OED method due to its excessive requirement of computational time. As shown in these figures, the proposed ML-based method without iterative re-estimation of the expected remaining MOCU showed identical

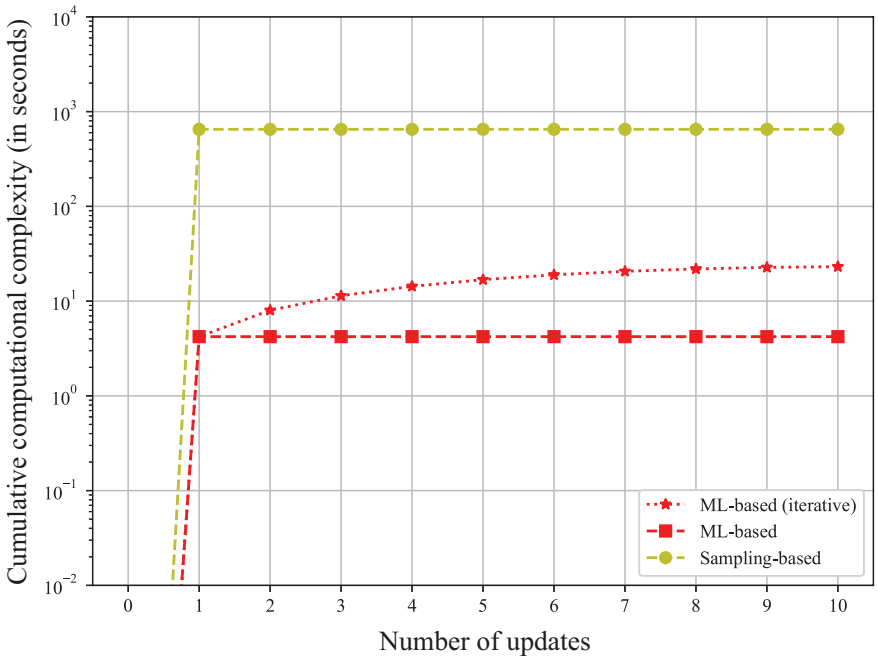


Figure 2.9: Average cumulative computational cost (in seconds) for identifying the optimal experiment based on different experimental design strategies for uncertain Kuramoto models with five oscillators. © 2021 IEEE.

performance to other best performers. Random experimental selection (blue dotted line) yielded a linearly decreasing MOCU curve, as we would expect on average. The entropy-based method showed similar performance as before (see Figure 2.6). Computational cost in Figure 2.9 shows a similar trend as before (see Figure 2.7). As before, the time for training the ML model is not included in this plot. Furthermore, Figure C.1 in Appendix C shows the RainCloud plot [62] that depicts the instantaneous performance of the different methods measured in terms of the remaining uncertainty (measured by MOCU) after performing the first experiment selected by the respective methods. As we can see from Figure C.1, all three MOCU-based OED schemes consistently yield the best overall performance.

We compared experimental sequences identified by the ML-based methods and the sampling-based approach to further investigate if the proposed ML approach can practically replace the sampling-based method for prioritizing the experiments in the experimental design space. The vertical axis corresponds to the number of intersecting experiments in the first k experiments pre-

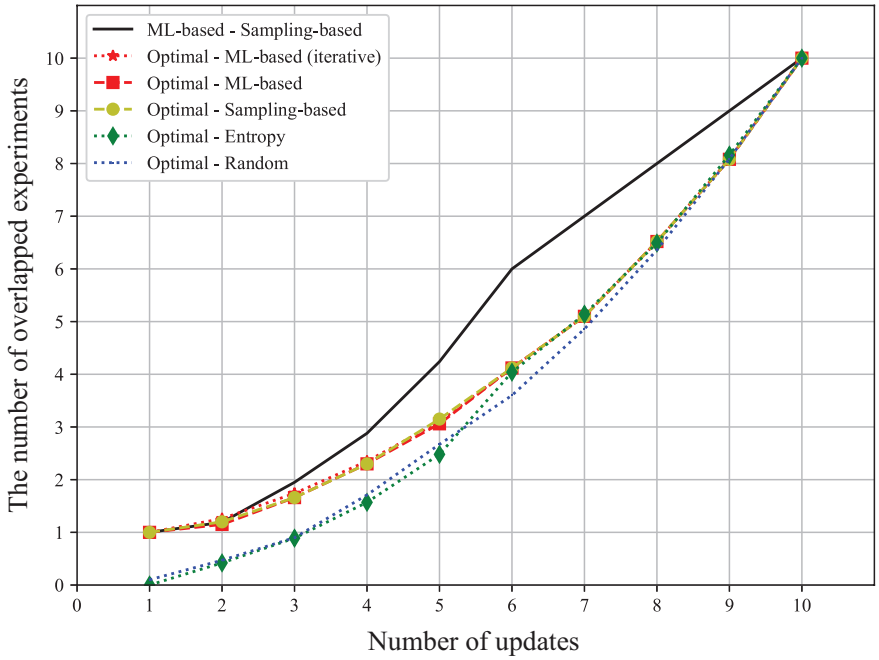


Figure 2.10: Comparison between the optimal sequence of experiments predicted by different OED strategies for uncertain Kuramoto models with five oscillators. The y-axis shows the number of common experiments within the first k experiments predicted by two different methods. © 2021 IEEE.

dicted by two different methods. If two methods predict the identical experimental sequence, the resulting curve will be a straight line (with unit slope). For example, the black line in Figure 2.10 compares the ML-based method and the sampling-based method. From Figure 2.10, we can see that the proposed ML-based method (without re-estimation) always identified the same first experiment as the sampling-based method in all one hundred evaluations. By comparing the true optimal experimental sequence (*i.e.*, predicted by an "oracle") and the sequences predicted by the ML-based method, we can see that the first optimal experiment was always accurately predicted. In fact, results in Figure 2.8 show that the first experiment leads to the most significant drop in model uncertainty, and all MOCU-based OED schemes (both ML-based and sampling-based) accurately predict this critical experiment. We also note that the entropy-based/random approaches tended to mispredict the best first experiment, resulting in a substantial performance gap when compared to the MOCU-based approaches. Figure 2.10 also shows that the predicted experimental sequences

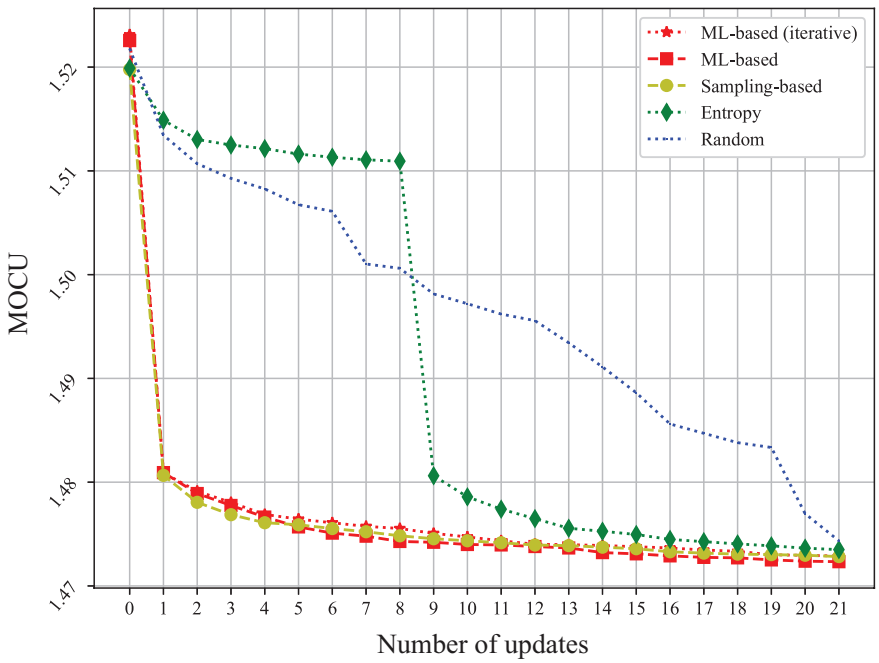


Figure 2.11: Average performance of various experimental design strategies for uncertain Kuramoto models with seven oscillators based on one hundred experiments. All MOCU-based methods led to the best performance, and random selection resulted in linear uncertainty reduction. © 2021 IEEE.

diverge in later iterations. However, this did not impact the OED performance on average, as later experiments did not reduce the model uncertainty as significantly as the earlier experiments.

We also repeated the experiments for uncertain Kuramoto models that consist of seven oscillators. As before, true (unknown) models were randomly sampled from the uncertainty class one hundred times to evaluate average performance.

Figure 2.11 shows the OED performance assessment results for the various experimental design methods based on the seven-oscillator Kuramoto model. As we can see from Figure 2.11, the performance trends were very similar to those seen in Figure 2.8 for the Kuramoto model with five oscillators. The proposed ML-based methods again accurately identified the first optimal experiment that maximally reduces MOCU on average. All four MOCU-based OED schemes (both ML-based and sampling-based), regardless of whether or not the remaining expected MOCU values were re-estimated after each experimental update, showed almost identical performance on average.

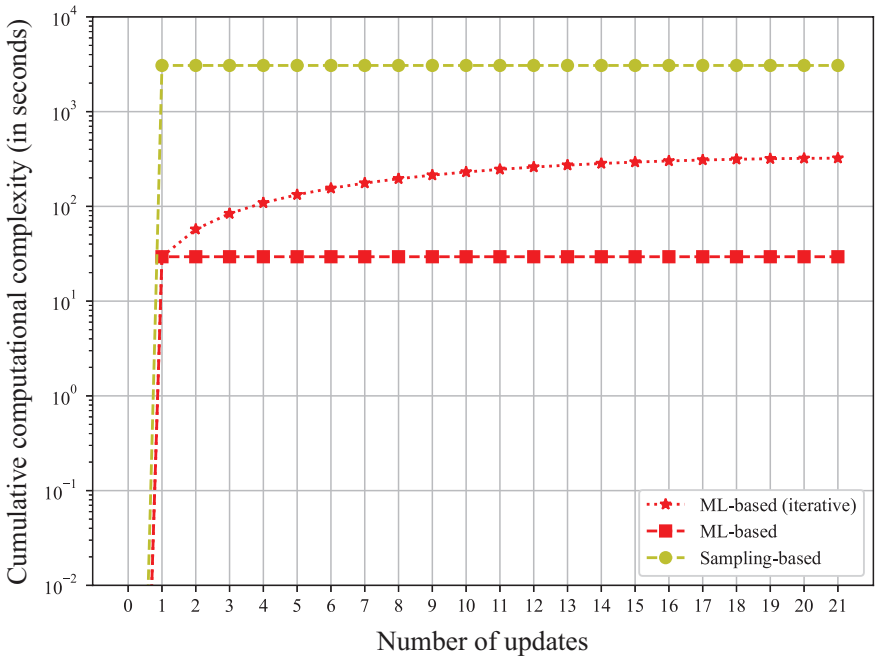


Figure 2.12: Average cumulative computational cost (in seconds) for identifying the optimal experiment based on different experimental design strategies for uncertain Kuramoto models with seven oscillators. © 2021 IEEE.

Figure C.2 in Appendix C compares the performance of different methods, where we measured the MOCU that remains after performing the first experiment selected by each method. The results are again shown for one hundred evaluations based on different true models. As shown in Figure C.2, the efficacy of the first experiment varies depending on the underlying true model, which is expected. As before, the results in Figure C.2 clearly show that the proposed ML-based OED scheme can effectively replicate the performance of the original sampling-based approach [27], the primary goal of this study. The computational time is shown in Figure 2.12, which clearly shows that the ML-based scheme (especially, the non-iterative scheme) is significantly faster compared to the original sampling-based approach. As before, the plot only shows the time for OED and does not include the training time for the ML model. Furthermore, even the ML-based method with the iterative update was considerably faster than the sampling-based that does not iteratively re-estimate the expected remaining MOCU.

Finally, we compared the experimental sequences identified by the different methods, including

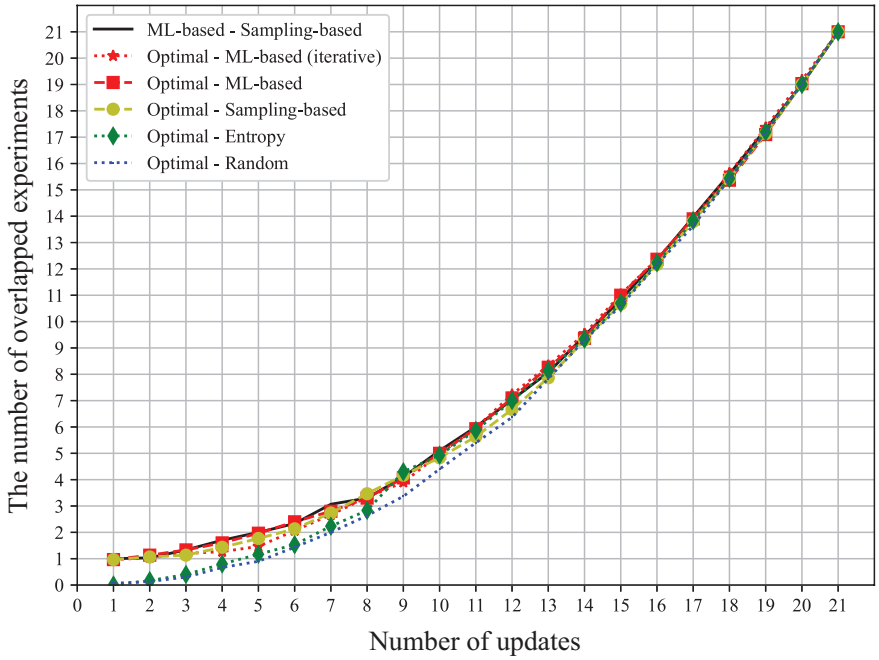


Figure 2.13: Comparison between the optimal sequence of experiments predicted by different OED strategies for uncertain Kuramoto model with seven oscillators. The y-axis shows the number of common experiments within the first k experiments predicted by two different methods. © 2021 IEEE.

the true optimal experimental selection (*i.e.*, predicted by an "oracle"). Note that due to the excessive computational cost of the optimal experimental selection (as it requires exhaustive search), we identified the optimal experimental sequences only for the first thirty evaluations based on randomly sampled true models from the uncertainty class. For this reason, Figure 2.13 shows the comparison results based on the first thirty experimental sequences (out of one hundred). As shown in the figure, both the ML-based and the sampling-based methods were able to accurately identify the first optimal experiment. The predicted sequences tended to diverge in later iterations. However, considering the simulation results shown in Figure 2.11, it is likely that this was because many experiments in later updates did not significantly reduce the objective uncertainty, once the best experiment has been performed in the earlier iterations (especially, the first iteration). Also, we can see that the entropy-based and the random selection approaches tended to miss the best experiment, which resulted in a significant degradation in the overall experimental design perfor-

mance. These comprehensive simulation results clearly showed that our proposed ML-based OED approach effectively quantifies the objective model uncertainty at a small fraction of the computational cost of the sampling-based method, thereby remarkably accelerating the OED process while maintaining excellent performance.

2.4 Concluding remarks

In this chapter, we proposed an ML approach that can significantly accelerate the objective-based quantification of model uncertainty via MOCU. A major bottleneck in applying MOCU for designing/prioritizing optimal experiments that can optimally reduce the uncertainty in models that represent real-world complex uncertain systems has been the high computational cost for accurately estimating MOCU. The proposed approach effectively addresses this issue in the context of OED for uncertain Kuramoto models by replacing the computational costly DE solver with an ML model, which remarkably speeds up the process of predicting the optimal controller (*i.e.*, the oscillator that guarantees global frequency synchronization at minimum cost). The trained ML model predicts the asymptotic behavior of a given Kuramoto model, namely, whether all oscillators in the model will be eventually frequency synchronized or not.

The comprehensive simulation results clearly demonstrate that the ML-based MOCU calculations are highly correlated with those computed by the sampling-based scheme originally proposed in [27]. Furthermore, the OED performance of the ML-based scheme is practically equivalent to that of the original sampling-based OED scheme. However, despite achieving equivalent OED performance, our proposed ML-based OED scheme accelerates the experimental design process by at least two orders of magnitude, resulting in significant computational gains. The remarkably enhanced computational efficiency enables more reliable MOCU calculation by further increasing the sample size (*i.e.*, K) as needed. Furthermore, it allows us to iteratively recompute the remaining MOCU $R\,(i,j)$ after performing the predicted optimal experiment at each experimental update (see Figure 2.9 and Figure 2.12), which can–in theory–lead to a more accurate prediction of the optimal experiment, although the actual gain will depend on the underlying model uncertainty. Such iterative update is practically infeasible for the original sampling-based OED scheme without

resorting to HPC (high-performance computing).

Our ML-based MOCU estimation and OED approach remarkably enhance the computational efficiency by refraining from repeatedly solving the DEs for the uncertain Kuramoto models for the sake of finding the optimal robust operator (which is required in the original sampling-based approach) but instead adopting ML for decision-making. However, as training the ML model requires the generation of sufficient training data, which also requires solving the coupled ODEs for different Kuramoto models in the uncertainty class, it will be interesting to compare the proposed ML-based approach with the sampling-based approach from the perspective of "data efficiency". For this purpose, we quantitatively compare the proposed approach with the sampling-based approach in terms of data requirements. For the uncertain Kuramoto model with five oscillators, we trained the ML model (an fcNN with a single hidden layer) with $40,000$ labeled sample points. Each sample point corresponds to the Kuramoto model with a different parameter, and labeling the sample point (*i.e.*, synchronized vs non-synchronized) requires solving the corresponding ODEs. The trained model is used throughout the entire experimental design process without the need for generating additional sample points. On the other hand, the sampling-based method requires generating approximately 2.2×10^7 labeled sample points (*i.e.*, by solving the DEs for different Kuramoto model parameters). Similarly, for the uncertain Kuramoto model with seven oscillators, we trained an fcNN model based on $100,000$ labeled sample points, and the trained model is used throughout the experimental design process. In comparison, the sampling-based approach requires the generation of around 9.4×10^7 labeled sample points. These comparisons clearly show that our proposed ML-based OED acceleration scheme not only improves the computational efficiency but also drastically improves the data efficiency.

3. PERFORMANCE OPTIMIZATION OF HIGH-THROUGHPUT VIRTUAL SCREENING (HTVS) PIPELINE

In various real-world scientific and engineering applications, the need for screening a large set of molecular candidates to identify a small subset of molecules that satisfy certain criteria or possess targeted properties arises fairly frequently. For example, since the Coronavirus disease 2019 (COVID-19) outbreak, there have been significant concurrent efforts among various groups of scientists to identify or develop drugs that can provide a potential cure for this extremely infectious disease. One such notable effort is IMPECCABLE (Integrated Modeling PipelinE for COVID Cure by Assessing Better LEads) [63] whose operational objective is to optimize the number of promising ligands that potentially lead to the successful discovery of drug molecules. To this aim, IMPECCABLE utilized deep learning-based surrogates for predicting docking scores and multi-scale biophysics-based computational models for computing docking poses of compounds. Built on the strength of massive parallelism on exascale computing platforms combined with RADICAL-Cybertools (RCT) managing heterogeneous workflows, IMPECCABLE identified promising leads targeted at COVID-19.

Considering that severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), the virus that can potentially lead to COVID-19, is known to rapidly mutate itself to create more infectious and deadlier variants [64], such drug screening process to identify anti-viral drug candidates that are effective against a specific variant may have to be repeated as new variants emerge. However, when one considers the huge search space of potential molecules–*e.g.*, ZINC (Zinc Is Not Commercial) 15 [65] contains about 230 million commercially available compounds. There are, however, about 10^{12} compounds that can be considered for drug design in chemical space theoretically–and the astronomical amount of computation that was devoted to the screening of drug candidates in [63] to screen 10^{11} candidates, this is without question a Herculean task that requires enormous resources and one that cannot be routinely repeated.

While different in scale and complexity, high-throughput virtual screening (HTVS) pipelines

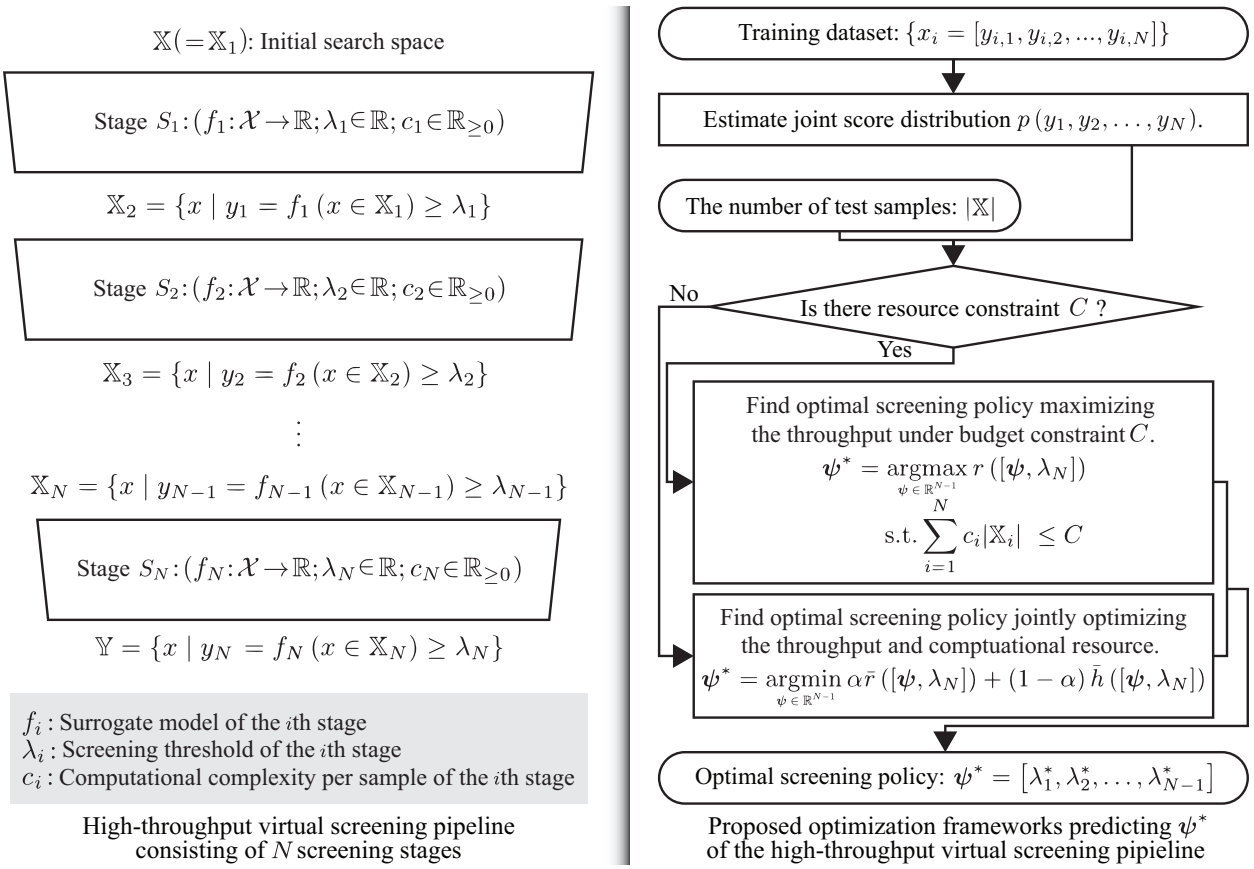


Figure 3.1: Illustration of a general high-throughput virtual screening (HTVS) pipeline (left) that consists of N stages (surrogate models) for rapid and reliable identification of a set $\mathbb{Y}$ of candidate molecules that likely possess the desired properties from a huge original set $\mathbb{X}$ containing all candidates. Stage S_i evaluates all the molecules $x \in \mathbb{X}_i$, which passed the previous stage S_{i-1}, via a surrogate model f_i. S_i passes the sample x to the next stage S_{i+1} if $f_i(x) \geq \lambda_i$. Otherwise, it discards the molecule. The proposed optimization framework shown on the right side predicts optimal screening policy $\boldsymbol{\psi}^* = \left[\lambda_1^*, \lambda_2^*, \ldots, \lambda_{N-1}^*\right]$ that yields the maximal throughput according to the screening campaign scenarios.

have been widely utilized in various fields, including biology [66, 67, 68, 69, 70, 71], chemistry [63, 72, 73, 74, 75, 76, 77], engineering [78], and materials science [79, 80]. However, the construction of such HTVS pipelines and the strategies for operating them heavily rely on expert intuition, often resulting in heuristic methods with reasonable yet sub-optimal screening performance. It remains a fundamental challenge to optimally construct and operate such screening pipelines to sift potential molecular candidates from an enormous search space in an efficient yet accurate manner.

In general, typical HTVS pipelines consist of multiple stages, each of which is associated with a surrogate model that evaluates the property of the molecules with a different accuracy/fidelity and computational cost. This is illustrated on the left side of Figure 3.1. At each stage in the pipeline, the molecular candidate is evaluated to determine whether the evaluation result appears promising enough to warrant passing it to the next–often more computationally expensive but more accurate–stage without unnecessarily wasting computational resources and time. In this way, the HTVS pipeline narrows down the number of candidate molecules, while sensibly allocating the available resources for investigating those that are promising and more likely to possess the desired property. The most promising candidates that remain at the end of screening may proceed to experimental validation, which is often more laborious, costly, and time-consuming. For example, in [70], an HTVS pipeline based on multi-fidelity surrogate models combined with an experimental platform successfully selected and reported a novel non-covalent inhibitor, MCULE-5948770040. The reported inhibitor has been identified by screening over 6.5 million molecules, and it has been shown to inhibit the SARS-Cov-2 main protease. HTVS pipelines have been also widely used for materials screening. For example, FHSP-NLO (First-principles High-throughput Screening Pipeline for Non-Linear Optical materials) [80] consisting of several computational predictors, based on density functional theory (DFT) calculations as well as data transformation and extraction methods, successfully identified deep-ultraviolet non-linear optical crystals that were reported in previous studies [81, 82, 83, 84, 85, 86, 87].

Although previous studies have demonstrated the advantages of constructing an HTVS pipeline for rapid screening of huge set of molecules to narrow down the most promising molecular candidates that are likely to possess the desired properties, the problem of *optimal decision-making* in such screening pipelines has not been extensively investigated to date. For example, how should one decide whether or not to pass a molecular candidate at hand to the next stage, given the outcome of the current stage? More specifically, in the HTVS example shown on the left side of Figure 3.1, how do we optimally determine the screening threshold of each stage for a given HTVS structure? Furthermore, if we were to modify the HTVS structure or construct it from scratch

by interconnecting multi-fidelity surrogate models, what would be the optimal structure of such HTVS that maximizes the throughput and accuracy? This requires selecting the optimal subset of the available multi-fidelity models, arranging them in the optimal order, and then exploring the interrelations among their predictive outcomes to make optimal operational decisions for the constructed HTVS.

In this chapter, we present a computational framework that can answer the aforementioned questions and applied to the optimization of HTVS pipelines involving that consist of multiple surrogate models with different costs and fidelity. The key idea is to estimate the joint probability distribution of predictive scores that result from the different stages constituting the HTVS pipeline, based on which we optimize the screening threshold values. We consider two optimization scenarios. First, we consider the case where the total computational budget is fixed and the goal is to maximize the throughput within the given budget. Second, we consider the case where we aim to jointly maximize the throughput of the HTVS pipeline while minimizing the overall computational cost required for screening. We demonstrate the performance of the proposed HTVS pipeline optimization framework based on both *simulated data* as well as *real data*. In the simulated example, the joint distribution of the predictive scores from the multi-fidelity models at different stages is assumed to be known, based on which we extensively evaluate the performance of the proposed approach under various scenarios. As a second example, we consider the problem of screening for long non-coding ribonucleic acids (lncRNAs). In this example, we first construct an HTVS pipeline by interconnecting existing lncRNA prediction algorithms with varying costs and accuracy and apply our proposed framework for performance optimization. Both examples clearly demonstrate the advantages of our proposed scheme, which leads to a substantial reduction of the total computational cost at virtually no degradation in overall prediction accuracy. Furthermore, we show that the proposed framework enables one to make an informed decision to balance the trade-off between speed and accuracy, where one could trade accuracy for higher efficiency, and vice versa.

3.1 Overview of HTVS pipeline

We assume that an HTVS pipeline consists of N screening stages $S_i : (f_i : \mathcal{X} \to \mathbb{R}; \lambda_i; c_i)$, $i = 1, 2, \ldots, N$, connected in series as shown in Figure 3.1 (left), where $f_i : \mathcal{X} \to \mathbb{R}$ is a surrogate model for predicting the property of interest for a given molecule and λ_i is the screening threshold. The average computational cost per sample for f_i associated with the ith stage S_i is denoted by c_i. At each stage S_i, the corresponding surrogate model f_i is used to evaluate the property of all molecules $x \in \mathbb{X}_i$ that passed the previous screening stage S_{i-1}, where $\mathbb{X}_i$ is given by:

$$\mathbb{X}_i = \{x \mid x \in \mathbb{X}_{i-1} \text{ and } f_{i-1}(x) \geq \lambda_{i-1}\}. \tag{3.1}$$

By definition, we have $\mathbb{X}_i \triangleq \mathbb{X}$, which contains the entire set of molecules to be screened. At stage S_i, every molecule $x \in \mathbb{X}_i$ whose property score $y_i = f_i(x)$ is below the threshold λ_i is discarded such that only the remaining molecules $x \in \mathbb{X}_{i+1}$ that meet or exceed this threshold are passed on to the next stage S_{i+1}. We assume that all molecules in $\mathbb{X}_i$ at each stage S_i are batch-processed to select the set of molecules $\mathbb{X}_{i+1}$ that will be passed to the subsequent stage S_{i+1}, as it is often done in practice [88, 89, 90].

Although every stage S_i in the screening pipeline performs a down-selection of the molecules by assessing their molecular property based on the surrogate model $f_i(x)$ and comparing it against the threshold λ_i, we assume only the threshold values $\lambda_1, \cdots, \lambda_{i-1}$ of the first $N-1$ stages will need to be determined while the threshold λ_N for the last screening stage S_N is predetermined. This reflects how such screening pipelines are utilized in real-world scenarios. For example, in the IMPECCABLE pipeline [63], as well as in many other computational drug discovery pipelines, potentially effective lead compounds that pass the earlier stages based on efficient but less accurate models will be assessed using computationally expensive yet highly accurate molecular dynamics (MD) simulations to evaluate the binding affinity against the target. Only the molecules whose binding affinity estimated by the MD simulations exceeds a reasonably high threshold set by domain experts may be further assessed experimentally, in order not to unnecessarily waste the avail-

able resources, considering that such experimental validation is typically costly, time-consuming, and labor-intensive. Similarly, in a materials screening pipeline, the last screening stage may involve expensive calculations based on DFT, a quantum mechanical modeling scheme that is widely used for predicting material properties [91, 92, 93, 94, 95, 96, 97, 98, 99].

Our primary goal is to design the optimal screening policy $\boldsymbol{\psi}^* = \left[\lambda_1^*, \lambda_2^*, \ldots, \lambda_{N-1}^*\right]$ that leads to the optimal operation of the HTVS pipeline. We consider two different scenarios. In the first scenario, we assume that the total computational budget for screening the candidate molecules is fixed, where the design goal would then be to identify the optimal screening policy that maximizes the screening throughput, namely, the percentage (or number) of potential molecules that meet or exceed the qualification in the last stage S_N (*i.e.*, $f_N(x) \geq \lambda_N$). In the second scenario, we consider the case when the computational budget is not fixed and where the goal is to design the optimal policy that simultaneously maximizes the throughput while minimizing the overall computational cost.

3.2 Methods

Figure 3.1 (right) shows a flowchart summarizing the proposed approach for identifying the optimal screening policy $\boldsymbol{\psi}^* = \left[\lambda_1^*, \lambda_2^*, \ldots, \lambda_{N-1}^*\right]$ for the optimal operation of a given HTVS pipeline under the two screening scenarios described above. First, we estimate the joint distribution $p\left(y_1, y_2, \ldots, y_N\right)$ of the predictive scores from the N stages based on the available training data. In case the probability density function (PDF) $p\left(y_1, y_2, \ldots, y_N\right)$ is known *a priori*, this PDF estimation step will not be required. Given $p\left(y_1, y_2, \ldots, y_N\right)$, we can predict the optimal screening policy $\boldsymbol{\psi}^* = \left[\lambda_1^*, \lambda_2^*, \ldots, \lambda_{N-1}^*\right]$ that leads to the optimal operational performance of the HTVS pipeline. Specifically, in case the total computational budget C is fixed, we find the optimal policy $\boldsymbol{\psi}^* = \left[\lambda_1^*, \lambda_2^*, \ldots, \lambda_{N-1}^*\right]$ that maximizes the screening throughput of the pipeline–*i.e.*, the proportion of molecules that pass the last (and the most stringent/accurate) screening stage that meet the condition $f_N\left(x\right) \geq \lambda_N$–under the budget constraint C. Otherwise, we predict optimal screening policy $\boldsymbol{\psi}^* = \left[\lambda_1^*, \lambda_2^*, \ldots, \lambda_{N-1}^*\right]$ that jointly optimizes the throughput and computational resource based on a weighted objective function that balances the throughput and the computa-

tional cost. In this case, the balancing weight α can be used to trade throughput for computational efficiency, or vice versa. We note that the training dataset is only used for estimating the PDF $p(y_1, y_2, \ldots, y_N)$ and not (directly) for finding the optimal screening policy. In fact, the optimal policy $\boldsymbol{\psi}^* = \left[\lambda_1^*, \lambda_2^*, \ldots, \lambda_{N-1}^*\right]$ is determined by a function of up to three parameters: the joint score distribution $p(y_1, y_2, \ldots, y_N)$, $|\mathbb{X}|$ (the number of potential molecules to be screened), and the total computational budget C (in the first screening scenario, where the computational budget is assumed to be limited).

As shown in Figure 3.1, the proposed optimization framework that identifies the optimal screening policy $\boldsymbol{\psi}^*$ takes a two-phase approach. In the first phase, we estimate the joint score distribution $p(y_1, y_2, \ldots, y_N)$. Based on the estimated score distribution, we find the optimal screening policy $\boldsymbol{\psi}^*$ that maximizes the screening performance. To ensure good screening performance, accurate estimation of the joint score distribution $p(y_1, y_2, \ldots, y_N)$ is crucial. In this study, we perform a spectral estimation under the assumption that the joint score distribution follows a multivariate Gaussian mixture model and estimate the parameters via the expectation-maximization (EM) scheme [100].

A formal objective is to determine the screening threshold λ_i at stage S_i, $i = 1, 2, \ldots, N-1$, such that the total number of the detected potential candidates in the set $\mathbb{Y}$, where the candidates meet the target criteria based on the score in the last stage S_N and the pre-specified screening threshold λ_N, is maximized under a given computational budget C. The relationship among the predictive scores from all stages $S_1, S_2, \ldots, S_N$ is captured by their joint score distribution $p(y_1, y_2, \ldots, y_N)$. Based on this joint score distribution, we define the following reward function $r(\boldsymbol{\lambda})$ according to policy $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_N]$ of the stages S_i, $i = 1, 2, \ldots, N$, as follows:

$$r(\boldsymbol{\lambda}) = \int_{[\lambda_N, \lambda_{N-1}, \ldots, \lambda_1]}^{\infty} \cdots \int p(y_1, y_2, \ldots, y_N)\, dy_1 dy_2 \cdots dy_N. \tag{3.2}$$

We can find the optimal screening policy $\boldsymbol{\psi}^* = \left[\lambda_1^*, \lambda_2^*, \ldots, \lambda_{N-1}^*\right]$ to be applied to the first $N-1$ stages (S_i, $i = 1, 2, \ldots, N-1$) that maximizes the reward $|\mathbb{Y}|$ by solving the constrained

optimization problem shown below:

$$\begin{aligned} \boldsymbol{\psi}^* = \underset{\boldsymbol{\psi}\in\mathbb{R}^{N-1}}{\arg\max} \quad & r\left([\boldsymbol{\psi}, \lambda_N]\right) \\ \text{s.t.} \quad & \sum_{i=1}^{N} c_i |\mathbb{X}_i| \ \leq C, \end{aligned} \tag{3.3}$$

where $|\mathbb{X}_i|$ is the number of molecules that passed the previous stages from S_1 to S_{i-1}. Formally, $|\mathbb{X}_i|$ is defined as:

$$|\mathbb{X}_i| = |\mathbb{X}| \int_{[\lambda_{i-1},\lambda_{i-2},\ldots,\lambda_1]}^{\infty} \cdots \int p_{1:i-1}\left(y_1, y_2, \ldots, y_{i-1}\right)\, dy_1 dy_2 \cdots dy_{i-1}, \tag{3.4}$$

where $p_{1:i-1}$ denotes the marginal score distribution for $y_1, \cdots, y_{i-1}$, which can be obtained by marginalizing $p(\cdot)$ over y_i to y_N.

In many real-world screening problems, including drug or material screening, the total computational budget for screening may not be fixed, and one may want to jointly optimize for both screening throughput as well as computational efficiency of screening. In such scenarios, we can formulate a joint optimization problem to find the best screening policy that strikes the optimal balance between throughput and efficiency:

$$\boldsymbol{\psi}^* = \underset{\boldsymbol{\psi}\in\mathbb{R}^{N-1}}{\arg\min}\, \alpha \bar{r}\left([\boldsymbol{\psi}, \lambda_N]\right) + (1-\alpha)\, \bar{h}\left([\boldsymbol{\psi}, \lambda_N]\right). \tag{3.5}$$

The weight parameter $\alpha \in [0, 1]$ determines the relative importance between the relative reward function $\bar{r}\left([\boldsymbol{\psi}, \lambda_N]\right)$, and the normalized total cost function $\bar{h}\left([\boldsymbol{\psi}, \lambda_N]\right)$ defined as follows:

$$\bar{r}\left([\boldsymbol{\psi}, \lambda_N]\right) = \frac{r\left([-\boldsymbol{\infty}, \lambda_N]\right) - r\left([\boldsymbol{\psi}, \lambda_N]\right)}{r\left([-\boldsymbol{\infty}, \lambda_N]\right)} \tag{3.6}$$

$$= \frac{\int_{\lambda_N}^{\infty} p_N\left(y_N\right)\, dy_N - r\left([\boldsymbol{\psi}, \lambda_N]\right)}{\int_{\lambda_N}^{\infty} p_N\left(y_N\right)\, dy_N}, \tag{3.7}$$

$$\bar{h}\left([\boldsymbol{\psi}, \lambda_N]\right) = \frac{1}{N|\mathbb{X}| \max_i c_i} \sum_{i=1}^{N} c_i |\mathbb{X}_i|. \tag{3.8}$$

Note that p_N is the marginal score distribution for y_N, which is obtained by marginalizing $p(\cdot)$ over y_1 to y_{N-1}.

3.3 Results and discussion

In this section, we validate the proposed optimization framework based on both synthetic and real data. First, we evaluated the performance of our optimization framework based on a four-stage HTVS pipeline, where the joint probability distribution of the predictive scores is assumed to be known. Next, we constructed an HTVS pipeline for lncRNAs by interconnecting existing lncRNA prediction algorithms with different prediction accuracy and computational complexity. In this example, the joint distribution of the predictive scores from the different algorithms at different stages was learned from training data, based on which the proposed HTVS optimal framework was used to identify the optimal screening policy.

We optimized the screening policy–for both optimization problems defined in Eq. (3.3) and Eq. (3.5)–using the differential evolution optimizer [101] in the *Scipy Python* package (version $1.7.0$). We performed all simulations on *Ubuntu* (version $20.04.2$ LTS) installed on *Oracle VM VirtualBox* (version $6.1.22$) that runs on a workstation equipped with *Intel* $i7 - 8809G$ CPU and 32GB RAM.

For comprehensive performance analysis of the proposed HTVS pipeline optimization framework, we considered a synthetic HTVS pipeline with $N = 4$ stages, where the joint PDF of the predictive scores from all stages is assumed to be known. We varied the correlation levels between the scores from neighboring stages to investigate the overall impact on the performance of the optimized HTVS pipeline.

Specifically, we assumed that the computational cost for screening a single molecule is 1 at stage S_1, 10 at S_2, 100 at S_3, and $1,000$ at S_N. As the per-molecule screening cost was fairly different across stages, the given setting for the synthetic HTVS pipeline allowed us to clearly see the impact and significance of optimal decision-making on the overall throughput and accuracy of

the screening pipeline.

We considered the case when we have complete knowledge of the joint score distribution $p(y_1, y_2, y_3, y_4)$. The score distribution was assumed to be a multivariate uni-modal Gaussian distribution $\mathcal{G}(\mathbf{0}, \mathbf{\Sigma}(\rho))$, where the covariance matrix $\mathbf{\Sigma}(\rho)$ is a Toeplitz matrix defined as follows:

$$\mathbf{\Sigma}(\rho) = \begin{bmatrix} 1 & \rho & \rho - 0.1 & \rho - 0.2 \\ \rho & 1 & \rho & \rho - 0.1 \\ \rho - 0.1 & \rho & 1 & \rho \\ \rho - 0.2 & \rho - 0.1 & \rho & 1 \end{bmatrix}, \tag{3.9}$$

where ρ is the correlation between neighboring stages S_i and S_{i+1} for $i = 1, 2, 3$. We assumed that the score correlation is lower between stages that are further apart, which is typically the case in real screening pipelines that consist of multi-fidelity models.

The primary objective of the HTVS pipeline was to maximize the number of potential candidates that satisfy the final screening criterion (*i.e.*, $f_4(x) \geq \lambda_4$) based on the highest fidelity model at stage S_4 while minimizing the total computational cost induced by the entire screening pipeline. The total number of all candidate molecules in the initial set $\mathbb{X}$ was assumed to be 10^5. We assumed that we are given $\lambda_4 = 3.0902$ as prior information set by a domain expert, which results in 100 molecules (among 10^5 in $\mathbb{X}$) that satisfy the final screening criterion. We validated the proposed HTVS optimization framework for two cases: first, for $\rho = 0.8$, where the neighboring stages yield scores that are highly correlated, and next, for $\rho = 0.5$ where the correlation is relatively low. Performance analysis results based on various other covariance matrices can be found in Appendix D.

Figure 3.2 shows the performance evaluation results for different HTVS pipeline structures optimized via the proposed framework under a fixed computational resource budget. The total number of the desirable candidates detected by the pipeline is shown as a function of the available computational budget for two cases: (A) HTVS pipeline that consists of highly-correlated stages (*i.e.*, $\rho = 0.8$) and (B) HTVS pipeline comprised of stages with lower correlation (*i.e.*, $\rho = 0.5$).

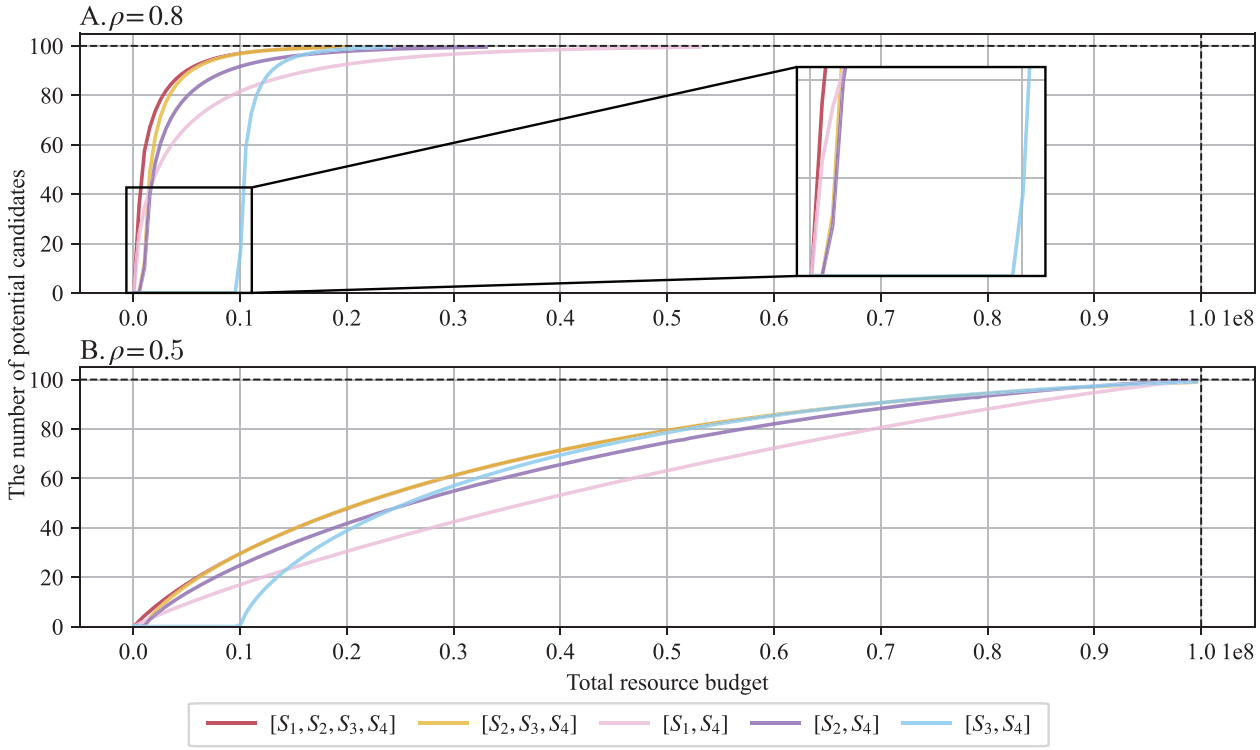


Figure 3.2: Performance assessment of the optimized HTVS pipelines. The number of candidate molecules that meet the desired screening criterion is shown as a function of the available computational budget. Results are shown for the case when the stages are highly correlated (A, $\rho = 0.8$) as well as when they have relatively low correlation (B. $\rho = 0.5$). Performance of the best performing 4-stage pipeline and the best performing 3-stage pipeline is shown. For comparison, we also show the performance of all 2-stage pipelines. Note that only the best-performing configurations are shown for $N \geq 3$.

The black horizontal and vertical dashed lines depict the total number of true candidates that meet the screening criterion (100 in this simulation) and the total computational budget required when screening all molecules in $\mathbb{X}$ only based on the last stage S_4 (*i.e.*, the highest-fidelity and most computationally expensive model), respectively. Figure 3.2 shows the performance of the best-performing $N = 4$ stage pipeline and that of the best-performing $N = 3$ pipeline. Additionally, the performance of all $N = 2$ stage pipelines is shown for comparison.

First, as shown in Figure 3.2.A, the performance curves of the pipelines consisting of only two stages (shown in pink, purple, and light blue lines) demonstrate how each of the lower-fidelity stages S_1–S_3 improves the screening performance when combined with the highest-fidelity stage S_4 and performance-optimized by our proposed framework. As we can observe in Figure 3.2.A, the correlation between the lower-fidelity/lower-complexity stage S_i, $i = 1, 2, \ldots, N-1$, at the beginning of the HTVS pipeline and highest-fidelity/highest-complexity stage S_N at the end of the pipeline had a significant impact on the slope of the performance curve. For example, in the two-stage pipeline $[S_3, S_4]$, where the two stages are highly correlated to each other, we could observe the steepest performance improvement as the available computational budget increased. On the other hand, for the two-stage pipeline $[S_1, S_4]$ which consists of less correlated stages, the performance improvement was more moderate in comparison as the available computational budget increased. Note that the minimum required computational budget to screen all candidates was larger for the pipeline $[S_3, S_4]$ compared to that for $[S_1, S_4]$, which was due to the assumption that all candidates are batch-processed at each stage. For example, with the minimum budget needed by pipeline $[S_3, S_4]$ to screen all candidates, the other pipelines $[S_1, S_4]$ and $[S_2, S_4]$ were capable of completing the screening and detecting more than 80% of the desirable candidates. Nevertheless, the detection performance improved with the increasing computational budget for all two-stage pipelines.

It is important to note that we can in fact simultaneously attain the advantage of using a lower-complexity stage (*e.g.*, $[S_1, S_4]$) that allows a "quick-start" with a small budget as well as the merit of using a higher-complexity stage (*e.g.*, $[S_3, S_4]$) for rapid performance improvement with the

budget increase by constructing a multi-stage HTVS pipeline and optimally allocating the computational resources according to our proposed optimization framework. This can be clearly seen in the performance curve for the four-stage pipeline $[S_1, S_2, S_3, S_4]$ (shown in the red solid line). The optimized four-stage pipeline consistently outperformed all other pipelines across all budget levels. Specifically, the optimized pipeline $[S_1, S_2, S_3, S_4]$ quickly evaluated all the molecular candidates in $\mathbb{X}$ through the most efficient stage S_1 and sharply improved the screening performance through the utilization of more complex yet also more accurate subsequent stages in the HTVS pipeline in a resource-optimized manner. For example, the optimized four-stage pipeline detected 97% of the desirable candidates that meet the target criterion at only 10% of the total computational cost that would be required if one used only the last stage (which we refer to as the "original cost"). To detect 99% of the desired candidates, the optimized four-stage pipeline $[S_1, S_2, S_3, S_4]$ would need only about 14% of the original cost.

Among all three-stage pipelines (*i.e.*, $N = 3$), pipeline $[S_2, S_3, S_4]$ yielded the best performance when performance-optimized using our proposed optimization framework (orange solid line in Figure 3.2.A). As we can see in Figure 3.2.A, the screening performance sharply increased as the available computational budget increased, thanks to the high correlation between S_4 and the prior stages S_2 and S_3. However, due to the higher computational complexity of S_2 compared to that of S_1, the optimized pipeline $[S_2, S_3, S_4]$ required a higher minimum computational budget for screening all candidate molecules compared to the minimum budget needed by a pipeline that begins with S_1. Despite this fact, when the first stage S_2 in this three-stage HTVS pipeline was replaced by the more efficient S_1, our simulation results (see Figure D.41 in Appendix D) showed that the screening performance improved relatively moderately as the budget increased. Empirically, when all stages are relatively highly correlated to each other, the best strategy for constructing the HTVS pipeline appears to place the stages in increasing order of complexity and optimally allocate the computational resources to maximize the return-on-computational-investment (ROCI). In fact, this observation is fairly intuitive and also in agreement with how screening pipelines are typically constructed in real-world applications.

Configuration	High correlation ($\rho = 0.8$)				Low correlation ($\rho = 0.5$)			
	Potential candidates	Total cost	Effective cost	Comp. savings	Potential candidates	Total cost	Effective cost	Comp. savings
$[S_4]$	100	$100,000,000$	$1,000,000$	0%	100	$100,000,000$	$1,000,000$	0%
$[S_1, S_4]$	94	$22,372,654$	$238,007$	76.20%	89	$56,551,129$	$635,406$	36.46%
$[S_2, S_4]$	96	$15,511,702$	$161,580$	83.84%	90	$43,620,751$	$484,675$	51.53%
$[S_3, S_4]$	98	$18,152,330$	$185,228$	81.48%	92	$41,522,035$	$451,326$	54.87%
$[S_1, S_2, S_4]$	97	$17,890,176$	$184,435$	81.56%	94	$53,340,817$	$567,456$	43.25%
$[S_1, S_3, S_4]$	98	$14,451,644$	$147,466$	85.25%	94	$47,550,232$	$505,854$	49.41%
$[S_2, S_1, S_4]$	97	$18,291,054$	$188,568$	81.14%	94	$53,513,582$	$569,293$	43.07%
$[S_2, S_3, S_4]$	98	$13,089,779$	$133,569$	86.64%	94	$44,534,328$	$473,769$	52.62%
$[S_3, S_1, S_4]$	99	$19,505,326$	$197,023$	80.30%	94	$48,708,112$	$518,171$	48.18%
$[S_3, S_2, S_4]$	99	$19,522,312$	$197,195$	80.28%	94	$47,966,605$	$510,283$	48.97%
$[S_1, S_2, S_3, S_4]$	99	$14,147,264$	$142,902$	85.71%	96	$50,336,621$	$524,340$	47.57%
$[S_1, S_3, S_2, S_4]$	99	$15,939,108$	$161,001$	83.90%	96	$52,704,450$	$549,005$	45.10%
$[S_2, S_1, S_3, S_4]$	99	$14,348,794$	$144,937$	85.51%	96	$50,366,503$	$524,651$	47.53%
$[S_2, S_3, S_1, S_4]$	99	$14,335,230$	$144,800$	85.52%	96	$50,411,458$	$525,119$	47.49%
$[S_3, S_1, S_2, S_4]$	99	$20,560,571$	$207,682$	79.23%	96	$53,249,970$	$554,687$	44.53%
$[S_3, S_2, S_1, S_4]$	99	$20,560,299$	$207,680$	79.23%	96	$53,215,674$	$554,330$	44.57%

Table 3.1: Performance comparison of various high-throughput virtual screening (HTVS) pipeline structures jointly optimized via the proposed framework ($\alpha = 0.5$).

Figure 3.2.B shows the performance evaluation results of the HTVS pipelines, where the screening stages were moderately correlated to each other ($\rho = 0.5$). Results are shown for different pipeline configurations, where the screening policy was optimized using the proposed framework to maximize the ROCI. Overall, the performance trends were nearly identical to those shown in Figure 3.2.A, although the overall performance was lower compared to the high correlation scenario ($\rho = 0.8$) as expected. While the screening performance of the optimized HTVS pipeline was not as good as the high-correlation scenario, the multi-stage HTVS pipeline with the optimized screening policy still provided a much better trade-off between the computational cost for screening and the detection performance. For example, if we were to use only the highest-fidelity model in S_4 for screening, the only way to trade accuracy for reduced resource requirements would be to randomly sample the candidate molecules from $\mathbb{X}$ and screen the selected candidates. The performance curve in this case would be a straight line connecting $(0, 0)$ and $(10^8, 100)$, below most of the performance curves for the optimized pipeline approach shown in Figure 3.2.B. As in the previous case ($\rho = 0.8$), the best pipeline configuration was to interconnect all four stages, where the stages were connected to each other in increasing order of complexity.

Table 3.1 shows the performance of the various HTVS pipeline configurations, where the

approach	High correlation ($\rho = 0.8$)				Low correlation ($\rho = 0.5$)			
	Potential candidates	Total cost	Effective cost	Comp. savings	Potential candidates	Total cost	Effective cost	Comp. savings
Proposed ($\alpha = 0.75$)	100	$19,727,704$	$197,277$	80.27%	99	$71,836,915$	$725,625$	27.44%
Proposed ($\alpha = 0.5$)	99	$14,147,264$	$142,902$	85.71%	96	$50,336,621$	$524,340$	47.57%
Proposed ($\alpha = 0.25$)	96	$10,926,901$	$113,822$	88.62%	86	$28,563,886$	$332,138$	66.79%
Baseline ($R_s = 0.75$)	100	$48,966,384$	$489,664$	51.03%	93	$48,662,387$	$523,251$	47.67%
Baseline ($R_s = 0.5$)	98	$15,599,934$	$159,183$	84.08%	69	$15,600,165$	$226,089$	77.39%
Baseline ($R_s = 0.25$)	78	$2,537,498$	$32,532$	96.75%	28	$2,537,516$	$90,626$	90.94%
Baseline ($R_s = 0.1$)	26	$400,000$	$15,385$	98.46%	6	$400,008$	$66,668$	93.33%

Table 3.2: Performance comparison between the proposed pipeline $[S_1, S_2, S_3, S_4]$ jointly optimized for throughput and computational efficiency (with various α) and the baseline pipeline (with different screening ratio R_s) in terms of the total number of detected potential candidates after screening and the computational cost induced.

screening policy was jointly optimized for both throughput and computational efficiency. The joint optimization problem is formally defined in Eq. (3.5), and α was set to 0.5 in these simulations. As a reference, the first row (configuration $[S_4]$) shows the performance of solely relying on the last stage S_4 for screening the molecules without utilizing a multi-stage pipeline. The effective cost is defined as the total computational cost divided by the total number of molecules detected by the screening pipeline that satisfy the target criterion (*i.e.*, average computational cost per detected candidate molecule). The computational savings of a given pipeline configuration is calculated by comparing its effective cost to that of the reference configuration (*i.e.*, $[S_4]$). As we can see in Table 3.1, our proposed HTVS pipeline optimization framework was able to significantly improve the overall screening performance across all pipeline configurations in a highly robust manner. For example, for $\rho = 0.8$, the optimized pipelines consistently led to computational savings ranging from 76.20% to 86.64% compared to the reference, while detecting $94 \sim 99\%$ of the desired candidates that meet the target criterion. Although the overall efficiency of the HTVS pipelines slightly decreased when the neighboring stages were less correlated ($\rho = 0.5$), the pipelines were nevertheless effective in saving computational resources. As shown in Table 3.1, the optimized HTVS pipelines detected $89\% \sim 96\%$ of all desired candidate molecules with computational savings ranging between 36.46% and 54.87%.

For further evaluation of the proposed framework, we performed additional experiments based on the four-stage pipeline $[S_1, S_2, S_3, S_4]$. In this experiment, we first investigated the impact of

α on the screening performance. Next, we compared the performance of the optimal screening policy with the performance of a baseline policy that mimics a typical screening scenario in real-world applications (*e.g.*, see [63]). The baseline policy selects the top $R_s\%$ candidate molecules at each stage and passes them to the next stage while discarding the rest. This baseline screening policy is agnostic of the joint score distribution of the multiple stages in the HTVS and aims to reduce the overall computational cost by passing only the top candidates to subsequent stages that are more costly. Similar strategies are in fact often adopted in practice due to their simplicity. In our simulations, we assumed the proportion R_s is uniform across the screening stages. The performance evaluation results are summarized in Table 3.2. When the neighboring stages were highly correlated ($\rho = 0.8$), the optimized pipelines detected 100, 99, and 96 candidate molecules at a total cost of $19,727,704$; $14,147,264$; and $10,926,901$, respectively. Interestingly, when α was reduced from 0.75 to 0.25 (*i.e.*, trading accuracy for higher efficiency), the number of detected candidate molecules decreased only by 4 (*i.e.*, from 100 to 96), while leading to an additional computational savings of 8 percentage points (*i.e.*, from 80.27% to 88.62%). On the other hand, the performance of the baseline screening policy was highly unpredictable and very sensitive to the choice of R_s. For example, although the baseline with $R_s = 0.75$ found all the potential candidates, the effective cost of the baseline was significantly higher than that of the proposed optimized pipeline with $[\alpha = 0.75]$. For $R_s = 0.5$, the baseline detected 98 potential candidates (out of 100) with a total cost of $15,599,934$, which was higher than the total cost of the optimized pipeline that detected 99 potential candidates. The baseline pipelines with $R_s = 0.1$ and 0.25 selected 26% and 78% of the potential candidates, respectively. Considering that the primary goal of such a pipeline is to detect the largest number of potential candidates in a computationally efficient manner, these results clearly showed that this baseline screening scheme that mimics conventional screening pipelines results in unreliable and suboptimal performance even when the neighboring stages were highly correlated to each other. While the baseline may lead to reasonably good performance for certain R_s, it is important to note that we cannot determine the optimal R_s in advance as the approach is agnostic to the relationships between different stages. As a result, the application of

this baseline screening pipeline may significantly degrade the screening performance in practice. When the correlation between the neighboring stages was relatively low ($\rho = 0.5$), the overall performance of the proposed pipeline degraded as expected. In this case, the pipeline jointly optimized for screening accuracy as well as efficiency with α set to 0.75, 0.5, and 0.25 detected 99, 96, and 86 potential candidates with the computational cost of $71,836,915$, $50,336,621$, and $28,563,886$, respectively. As in the high correlation case, the performance of the baseline scheme significantly varied and was sensitive to the choice of R_s.

To demonstrate the efficacy of the proposed optimization framework in a real-world application, we considered an optimal computational screening campaign for the identification of lncRNAs. Given a large number of RNA transcripts, the goal is to efficiently and accurately detect lncRNA transcripts through an HTVS pipeline. In recent years, interests in lncRNAs have been constantly increasing in relevant research communities, as there is growing evidence that lncRNAs and their roles in various biological processes are closely associated with the development of complex and often hard-to-treat diseases including Alzheimer's diseases [102, 103, 104], cardiovascular diseases [105, 106], as well as several types of cancer [107, 108, 109, 110]. RNA sequencing techniques are nowadays routinely used to investigate the main functional molecules and their molecular interactions responsible for the initiation, progression, and manifestation of such complex diseases. Consequently, the accurate detection of lncRNA transcripts from a potentially huge number of sequenced RNA transcripts is a fundamental step in studying lncRNA-disease association. While several lncRNA prediction algorithms have been developed so far [111, 112, 113, 114], each of which with its own pros and cons, no HTVS pipeline has been proposed to date for fast and reliable screening of lncRNAs.

First, we collected the nucleotide sequences of *Homo sapiens* RNA transcripts from GENCODE v38 (May 5, 2021) [115], which consists of $48,752$ lncRNA sequences and $106,143$ protein-coding sequences. We filtered out sequences that contain any unknown nucleotides (other than A, U, C, or G) and sequences whose length exceeds $3,000$nt. This resulted in $45,216$ lncRNA sequences and $79,030$ protein-coding sequences. Next, we applied a clustering algorithm CD-

Algorithm	Accuracy	Sensitivity	Specificity	Time per RNA (ms)
CPC2 [113]	0.7154	0.5760	0.9493	2.5265
CPAT [111]	0.8217	0.6861	0.9817	2.7336
PLEK [112]	0.7050	0.5666	0.9478	83.1765
LncFinder [114]	0.8329	0.7062	0.9678	2,495.6231

Table 3.3: Performance of the four individual long non-coding ribonucleic acids (lncRNAs) prediction algorithms that constitute the lncRNA HTVS pipeline. The average accuracy, sensitivity, specificity, and processing time (per RNA transcript) are shown.

hit [116] to lncRNAs and protein-coding RNAs, respectively, to remove redundant sequences. We finally obtained a set of $104,733$ RNA transcripts, consisting of $39,785$ lncRNA sequences and $64,948$ protein-coding sequences.

For the construction of the lncRNA screening pipeline, we selected four state-of-the-art lncRNA prediction algorithms that have been shown to achieve good prediction performance: CPC2 (Coding Potential Calculator 2) [113], CPAT (Coding Potential Assessment Tool) [111], PLEK (Predictor of LncRNAs and mEssenger RNAs based on an improved k-mer scheme) [112], and LncFinder [114].

Table 3.3 summarizes the performance of the individual algorithm based on the GENCODE dataset, preprocessed as described previously. We assessed the accuracy, sensitivity, and specificity of the respective lncRNA prediction algorithms. For algorithm CPAT, which yields confidence scores between 0 and 1 rather than a binary output, we set the decision boundary to 0.5 for lncRNA classification. As shown in Table 3.3, LncFinder achieved the accuracy, sensitivity, and specificity of 0.8329, 0.7062, and 0.9678, respectively, outperforming all other algorithms in terms of accuracy and sensitivity. However, LncFinder also had the highest computational cost among the compared algorithm, where processing an RNA transcript required $2,495.6231$ milliseconds on average. CPAT was the second-best performer among the four in terms of accuracy and sensitivity. Furthermore, CPAT also achieved the highest specificity. CPC2 and PLEK were less accurate compared to LncFinder and CPAT in terms of accuracy, sensitivity, and specificity. Despite their high computational efficiency, both CPC2 and CPAT also outperformed PLEK based on overall

Figure 3.3: One of the optimal structures of the HTVS pipeline for selecting long non-coding ribonucleic acids (lncRNAs).

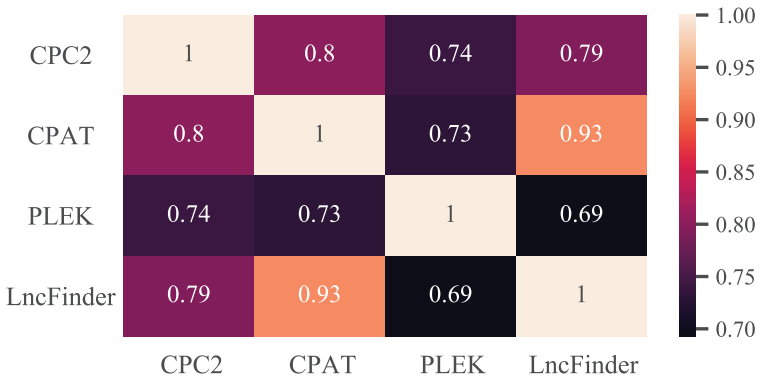


Figure 3.4: The heat map showing the Pearson's correlation coefficient between different stages. CPAT had the highest correlation to LncFinder. While PLEK was computationally more complex compared to CPAT, it showed a relatively lower correlation to LncFinder.

accuracy.

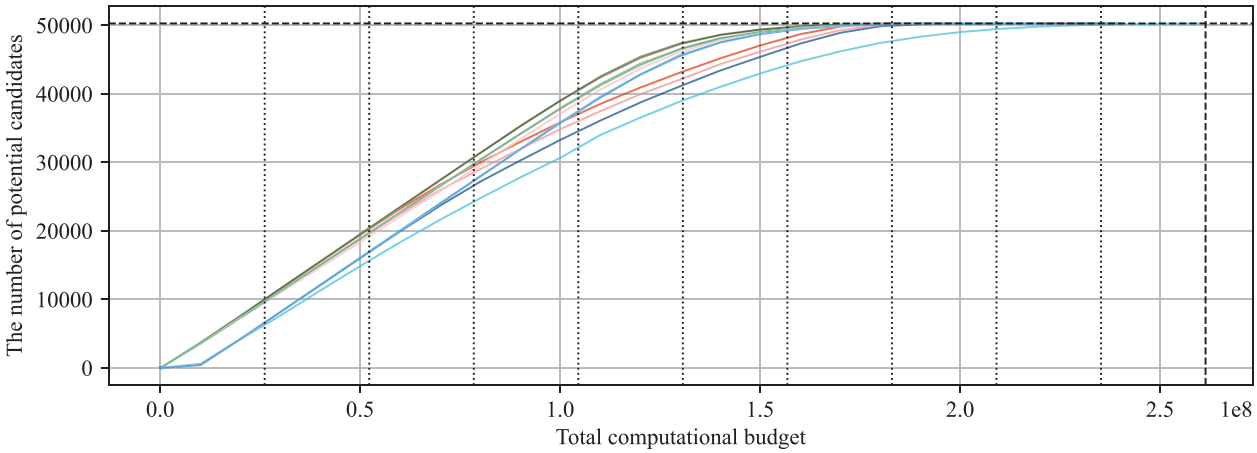
As we previously observed from the performance assessment results based on the synthetic pipeline, the efficacy of the optimized HTVS pipeline is critically dependent on the correlation between the stages constituting the pipeline. The proposed HTVS optimization framework aims to exploit the correlation structure across different screening stages to find the optimal screening policy that strikes the optimal balance between the screening throughput and the computational cost of screening. Here we placed LncFinder–the most accurate and the most computationally costly algorithm among the four–in the final stage. In the first three stages in the HTVS pipeline, we placed CPC2, CPAT, and PLEK, in the order of increasing computational complexity. The resulting HTVS pipeline structure is depicted in Figure 3.3. After constructing the screening pipeline, we computed the Pearson's correlation coefficient between the predictive output scores obtained from different algorithms. As shown in Figure 3.4, CPAT showed the highest correlation with LncFinder in the last stage (with a correlation coefficient of 0.93), the highest among the first three stages in

the screening pipeline.

To apply our proposed HTVS optimization framework, we first estimated the joint probability distribution $p(y_1, y_2, y_3, y_4)$ of the predictive scores generated by the four different lncRNA prediction algorithms–CPC2 (y_1), CPAT (y_2), PLEK (y_3), and LncFinder (y_4)–via the EM algorithm [100]. For training, 4% of the preprocessed GENCODE data was used. Note that all the computational lncRNA identification algorithms considered in this study output protein-coding probabilities, hence a higher output value corresponds to a higher probability for a given transcript to be protein-coding. Since our goal was to identify the lncRNAs, we multiplied the output scores generated by the algorithms by -1 such that higher values represent higher chances to be lncRNA transcripts. The screening threshold for the LncFinder in the last stage of the HTVS pipeline was set to $\lambda_4 = 0.2$, which leads to the optimal overall performance of LncFinder with a good balance between sensitivity and specificity.

Figure 3.5 shows the performance of the optimized lncRNA HTVS pipeline for various pipeline structures with the different numbers of stages and ordering. The black horizontal dashed line indicates the total number of potential candidates (*i.e.*, the total number of functional lncRNAs in the test set) and the black vertical dashed line shows the total computational cost (referred to as the "original cost" as before) that would be needed for screening all candidates based on the last stage LncFinder alone, without using the HTVS pipeline. Black vertical dotted lines are located at intervals of $1/10$ of this original cost. Underneath each dotted line, the number of potential candidates (*i.e.*, true functional lncRNAs) detected by each optimized HTVS pipeline is shown (see the columns in the table aligned with the dotted lines in the plot).

As before, we assumed that the candidates are batch-processed at each stage. As a result, for a given pipeline structure, the computational cost of the first stage determined the minimum computational resources needed to start screening. The correlation between the neighboring stages was closely related to the slope of the corresponding performance curve, which is a phenomenon that we already noticed before based on synthetic pipelines. For example, at 10% of the original cost, the pipelines starting with PLEK (*i.e.*, S_3) showed the worst performance among the tested



		10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	100 %
	$[S_1, S_4]$	9,973	20,280	29,452	37,071	43,250	48,210	50,220	50,265	50,266	50,266
	$[S_2, S_4]$	10,020	20,418	30,749	40,615	47,400	49,719	50,244	50,266	50,266	50,266
	$[S_3, S_4]$	6,218	15,643	24,246	32,032	39,017	44,202	47,696	49,447	50,091	50,254
	$[S_1, S_2, S_4]$	10,033	20,364	30,692	40,537	47,365	49,713	50,244	50,266	50,266	50,266
	$[S_1, S_3, S_4]$	9,563	19,570	28,678	36,164	42,218	47,364	50,125	50,252	50,266	50,266
	$[S_2, S_1, S_4]$	10,025	20,397	30,765	40,695	47,466	49,717	50,244	50,266	50,266	50,266
	$[S_2, S_3, S_4]$	9,688	19,753	29,760	39,431	46,680	49,467	50,198	50,264	50,266	50,266
	$[S_3, S_1, S_4]$	6,530	16,866	26,590	34,663	41,332	46,705	49,974	50,254	50,265	50,266
	$[S_3, S_2, S_4]$	6,607	16,997	27,335	37,464	45,742	49,239	50,162	50,264	50,266	50,266
	$[S_1, S_2, S_3, S_4]$	9,647	19,711	29,728	39,335	46,647	49,456	50,194	50,264	50,266	50,266
	$[S_1, S_3, S_2, S_4]$	9,518	19,184	28,983	38,729	46,271	49,347	50,160	50,264	50,266	50,266
	$[S_2\ S_1, S_3\ S_4]$	9,692	19,741	29,768	39,393	46,758	49,464	50,197	50,264	50,266	50,266
	$[S_2, S_3, S_1, S_4]$	9,677	19,734	29,730	39,393	46,745	49,460	50,197	50,264	50,266	50,266
	$[S_3, S_1, S_2, S_4]$	6,448	16,937	27,276	37,429	45,697	49,222	50,158	50,264	50,266	50,266
	$[S_3, S_2, S_1, S_4]$	6,601	16,964	27,329	37,429	45,709	49,230	50,159	50,264	50,266	50,266

Figure 3.5: Performance evaluation of the optimized lncRNA HTVS pipeline. The number of potential candidates (*i.e.*, lncRNAs) detected by the HTVS pipeline is shown under various computational budget constraints (x-axis). Various different pipeline structures were tested, where the results show that the proposed optimization framework leads to efficient and reliable performance regardless of the structure used.

pipelines in terms of the throughput. Specifically, $[S_3, S_4]$, $[S_3, S_1, S_4]$, $[S_3, S_2, S_4]$, $[S_3, S_1, S_2, S_4]$, and $[S_3, S_2, S_1, S_4]$ detected $6,218$; $6,530$; $6,607$; $6,448$; and $6,601$ lncRNAs, respectively. On the other hand, pipelines starting with either CPC2 or CPAT (*i.e.*, S_1 or S_2) detected $9,518$ to $10,033$ lncRNAs at the same cost. In addition, pipelines $[S_2, S_4]$, $[S_1, S_2, S_4]$, $[S_2, S_1, S_4]$, $[S_2, S_3, S_4]$, $[S_3, S_2, S_4]$, $[S_1, S_2, S_3, S_4]$, $[S_1, S_3, S_2, S_4]$, $[S_2, S_1, S_3, S_4]$, $[S_2, S_3, S_1, S_4]$, $[S_3, S_1, S_2, S_4]$, and $[S_3, S_2, S_1, S_4]$ including the second stage associated with CPAT that is highly correlated to the last stage LncFinder showed the steepest performance improvement. As a result, all HTVS pipelines that include CPAT were able to identify nearly all true lncRNAs (*i.e.*, $45,697$ to $47,466$) at only 50% of the original cost, regardless of at which stage CPAT was placed in the pipeline.

While the structure of the HTVS pipeline impacts the overall screening performance, Figure 3.5 shows that our proposed optimization framework alleviated the performance dependency on the underlying structure by optimally exploiting the relationships across different stages. For example, although the optimized pipeline $[S_1, S_2, S_4]$ outperformed the optimized pipeline $[S_1, S_2, S_3, S_4]$, which additionally included PLEK (*i.e.*, S_3), the performance gap was not very significant. The maximum difference between the two pipeline structures in terms of the detected lncRNAs was $1,202$ when the computational budget was set at 40% of the original cost. However, when considering that PLEK (S_3) was computationally much more expensive compared to CPC2 (S_1) and CPAT (S_2) and also had a lower correlation with LncFinder (S_4), the throughput difference of $1,202$ was only about 2.4% of the total lncRNAs in the test dataset, which is relatively small. Moreover, this throughput difference was drastically reduced as the available computational resources increased. For example, when the computational budget was set at 70% of the original cost, the throughput difference between the two pipelines was only 50 (see Appendix E for detailed simulation results).

In practice, real-world HTVS pipelines may involve various types of screening stages using multi-fidelity surrogate models. The computational complexity and the fidelity of such surrogate models may differ significantly and the structure of the pipeline may vastly vary depending on the domain experts designing the pipeline. Considering these factors, an important advantage of our proposed HTVS pipeline optimization framework is its capability to consistently attain efficient

Configuration	Potential candidates	Total cost (ms)	Effective cost	Computational savings	Accuracy	Sensitivity	Specificity	F1
$[S_4]$	$50,266$	$261,374,090$	$5,200$	0%	0.8440	0.9264	0.7936	0.8186
$[S_1, S_4]$	$48,875$	$161,357,081$	$3,301$	36.52%	0.8429	0.9075	0.8034	0.8144
$[S_2, S_4]$	$47,950$	$134,366,143$	$2,802$	46.12%	0.8624	0.9215	0.8262	0.8357
$[S_3, S_4]$	$47,083$	$176,963,736$	$3,758$	27.73%	0.8450	0.8876	0.8188	0.8131
$[S_1, S_2, S_4]$	$48,210$	$134,748,992$	$2,795$	46.25%	0.8600	0.9216	0.8222	0.8333
$[S_1, S_3, S_4]$	$49,100$	$168,490,516$	$3,432$	34.00%	0.8442	0.9120	0.8026	0.8164
$[S_2, S_1, S_4]$	$48,214$	$134,812,024$	$2,796$	46.23%	0.8600	0.9216	0.8222	0.8334
$[S_2, S_3, S_4]$	$48,295$	$141,710,246$	$2,934$	43.58%	0.8602	0.9230	0.8218	0.8338
$[S_3, S_1, S_4]$	$49,119$	$171,803,403$	$3,498$	32.73%	0.8444	0.9124	0.8026	0.8166
$[S_3, S_2, S_4]$	$48,326$	$146,100,080$	$3,023$	41.86%	0.8600	0.9231	0.8214	0.8336
$[S_1, S_2, S_3, S_4]$	$48,402$	$140,954,256$	$2,912$	44.00%	0.8591	0.9228	0.8200	0.8326
$[S_1, S_3, S_2, S_4]$	$48,332$	$141,229,518$	$2,922$	43.81%	0.8587	0.9215	0.8203	0.8321
$[S_2, S_1, S_3, S_4]$	$48,409$	$141,022,859$	$2,913$	43.98%	0.8591	0.9229	0.8200	0.8326
$[S_2, S_3, S_1, S_4]$	$48,414$	$141,225,328$	$2,917$	43.90%	0.8591	0.9230	0.8200	0.8327
$[S_3, S_1, S_2, S_4]$	$48,424$	$145,321,388$	$3,001$	42.29%	0.8589	0.9228	0.8197	0.8324
$[S_3, S_2, S_1, S_4]$	$48,429$	$145,388,626$	$3,002$	42.27%	0.8589	0.9229	0.8197	0.8325

Table 3.4: Performance evaluation of the lncRNA HTVS pipeline jointly optimized for throughput and efficiency. Results are shown for various pipeline configurations, where the optimized screening policy was used (with $\alpha = 0.5$).

and accurate screening performance that may weather the effect of potentially suboptimal design choices in constructing real-world HTVS pipelines.

Next, we evaluated the performance of the lncRNA HTVS pipeline, jointly optimized for both throughput and efficiency based on the proposed framework (with $\alpha = 0.5$). The results for various pipeline configurations are shown in Table 3.4. On average, the optimized HTVS pipeline detected $48,372$ lncRNAs out of $50,266$ total lncRNAs in the test dataset. The average effective cost was $3,067$. Pipeline configurations that include CPAT (S_2) achieved relatively higher computational savings (ranging from 41.86% to 46.25%) compared to those without S_2 (ranging from 27.73% to 36.52%). As we have previously observed, our proposed optimization framework was effective in maintaining its screening efficiency and accuracy even when the pipeline included a stage (*e.g.*, PLEK) that is less correlated with the last and the highest-fidelity stage (*i.e.*, LncFinder). In fact, the inclusion of a suboptimal stage in the HTVS pipeline does not significantly degrade the average screening performance. This is because the proposed optimization framework enables one to select the optimal threshold values that can sensibly combine the benefits of the most efficient stages (such as CPC2 and CPAT in this case) as well as the most accurate stage (LncFinder), thereby maximizing the expected ROCI. Similar observation can be made regarding the ordering of the

Approach	Potential candidates	Total cost (ms)	Effective cost	Computational savings	Accuracy	Sensitivity	Specificity	F1
Proposed ($\alpha = 0.75$)	$48,965$	$148,155,016$	$3,026$	41.81%	0.8553	0.9249	0.8126	0.8292
Proposed ($\alpha = 0.5$)	$48,402$	$140,954,256$	$2,912$	44.00%	0.8591	0.9228	0.8200	0.8326
Proposed ($\alpha = 0.25$)	$47,106$	$131,830,857$	$2,799$	46.17%	0.8650	0.9143	0.8348	0.8373
Baseline ($R_s = 0.75$)	$39,079$	$115,643,459$	$2,959$	43.10%	0.8366	0.7761	0.8737	0.7801
Baseline ($R_s = 0.5$)	$12,653$	$35,255,772$	$2,786$	46.42%	0.7170	0.2866	0.9807	0.4348
Baseline ($R_s = 0.25$)	$1,402$	$4,963,415$	$3,540$	31.92%	0.6318	0.0330	0.9986	0.0638

Table 3.5: Performance of the four-stage lncRNA HTVS pipeline $[S_1, S_2, S_3, S_4]$. The overall performance of the HTVS pipeline jointly optimized for throughput and efficiency is compared to that of the baseline screening approach.

multiple screening stages, as Table 3.4 shows that the average performance does not significantly depend on the actual ordering of the stages when the screening threshold values are optimized via our proposed framework. For example, when using all four stages in the HTVS pipeline ($N = 4$), the optimized pipeline detected $48,402$ lncRNAs on average and with consistent computational savings ranging between 42.27% and 44.00%. We also evaluated the accuracy of the potential candidates screened by the optimized HTVS pipeline based on four performance metrics: accuracy, sensitivity, specificity, and F1 score. Interestingly, all configurations except for $[S_1, S_4]$ outperformed LncFinder in terms of accuracy. In terms of sensitivity, the optimized pipeline achieved an average sensitivity of 0.9177. All pipeline configurations resulted in higher specificity compared to LncFinder. Besides, pipeline configurations that include S_2 consistently outperformed LncFinder in terms of the F1 score.

Finally, we compared the performance of the optimized pipeline to that of the baseline approach that selects the top $R_s\%$ of the incoming candidates for the next stage, where $R_s\%$ is a parameter to be determined by a domain expert. For this comparison, we considered the four-stage pipeline $[S_1, S_2, S_3, S_4]$. The optimal screening policy was found based on our proposed framework using three different values of $\alpha \in \{0.25, 0.50, 0.75\}$. The baseline screening approach was evaluated based on four different levels of $R_s \in \{25\%, 50\%, 75\%\}$. The performance assessment results are summarized in Table 3.5. As shown in Table 3.5, the baseline approach detected fewer lncRNAs for all values of R_s compared to the optimized pipeline. Specifically, the jointly optimized pipeline detected $48,965$; $48,402$; and $47,106$ lncRNAs at a cost of $148,155,016$ ($\alpha = 0.75$); $140,954,256$

($\alpha = 0.50$); and $131,830,857$ ($\alpha = 0.25$), respectively. On the other hand, the baseline approach with $R_s = 75\%$ detected only $39,079$ lncRNAs at a total cost of $115,643,459$. For $R_s = 50\%$ and $R_s = 25\%$, the baseline scheme detected only $12,653$ and $1,402$ lncRNAs, respectively. In terms of the four quality metrics (accuracy, sensitivity, specificity, and F1), the optimized pipeline outperformed the baseline scheme in terms of accuracy, sensitivity, and F1. The optimized pipeline resulted in lower specificity compared to the baseline. However, it should be noted that the potential candidates detected by the optimized HTVS pipeline are remarkably higher compared to the baseline approach. This is clearly reflected in the much lower sensitivity of the baseline approach, as shown in Table 3.5. As a result, the baseline approach tended to achieve significantly lower accuracy and F1 compared to the optimal screening scheme.

3.4 Concluding remarks

In this chapter, we proposed a general mathematical framework for identifying the optimal screening policy that can maximize the ROCI of an HTVS pipeline. The need for screening a large set of molecules to detect potential candidates that possess the desired properties frequently arise in various science and engineering domains, although the design and operation of such screening pipelines strongly depend on expert intuitions and *ad hoc* approaches. We aimed to rectify this problem by taking a principled approach to high-throughput virtual screening (HTVS), thereby maximizing the screening performance of a given HTVS pipeline, reducing the performance dependence on the pipeline configuration, and enabling quantitative comparison between different HTVS pipelines based on their optimal achievable performance.

We considered two scenarios for HTVS performance optimization in this study: first, maximizing the detection of potential candidate molecules that possess the desired property under a constrained computational budget; second, jointly optimizing the throughput and the computational efficiency of the HTVS pipeline when there is no fixed computational budget for the screening operation. For both scenarios, we have thoroughly tested the performance of our proposed HTVS optimization framework. Comprehensive performance assessment based on synthetic HTVS pipelines as well as real lncRNA screening pipelines both showed clear advantages of the

proposed framework. Not only does the HTVS optimization framework remove the guesswork in the operation of HTVS pipelines to maximize the throughput, enhance the screening accuracy, and minimize the computational cost, it leads to reliable and consistent screening performance across a wide variety of HTVS pipeline structures. This is a significant benefit of the proposed framework that is of practical importance since it makes the overall screening performance robust to variations and potentially suboptimal design choices in constructing real-world HTVS pipelines. As there can be infinite different ways of building an HTVS pipeline in real scientific and engineering applications, it is important to note that our proposed optimization framework can guarantee near-optimal screening performance for any reasonable design choice regarding the HTVS pipeline configuration.

4. CONSTRUCTION AND OPTIMIZATION OF GENERALIZED HIGH-THROUGHPUT VIRTUAL SCREENING (HTVS) PIPELINE

With the increasing interest in renewable energy sources, there has been an explosive need to develop novel energy storage devices that can overcome the problems that conventional Li-ion batteries have suffered [117, 118, 119, 120]. Remarkably, organic electrode material-based energy storage devices have attracted explosive attention as they possess favorable characteristics. First, the organic material can be synthesized from earth-abundant precursors such as C, H, O, or N. Besides, they do not utilize toxic heavy metals that cause serious environmental issues. More importantly, organic redox-active material-based batteries are highly potent for a significant increase in capabilities as opposed to the traditional inorganic material-based batteries [117].

One fundamental challenge in developing novel energy storage devices based on organic electrode materials is to construct a subset of the promising organic electrode material candidates that possess target redox potential (RP) computed at the desired fidelity. Since there are infinitely many organic materials to be considered and desired fidelity requires an excessive amount of computational resources per molecule, the exhaustive computational screening campaign is practically infeasible. Recently, several machine learning studies have been dedicated to predicting the structure-electrochemical property relationships efficiently [121, 122, 123, 124]. For example, in [121], a fully-connected neural network (fcNN) with two hidden layers accurately approximated the RP of molecules based on ten features–the number of B/C/Li/O/H, the number of aromatic rings, highest occupied molecular orbital (HOMO), lowest unoccupied molecular orbital (LUMO), HOMO-LUMO gap, and electron affinity (EA). However, despite the predictive efficiency, such machine-learning approaches have not been systematically exploited in the context of an objective-driven computational screening campaign, resulting in only utilizing them for prioritizing the materials for further evaluation based on the desired fidelity model.

One practical goal-driven approach for effective selection of promising candidates is to build a high-throughput virtual screening (HTVS) pipeline consisting of various mathematical or surro-

gate models with different fidelity. In the early stage, HTVS pipelines use computationally efficient models to discard samples that are unlikely to possess the desired property and pass the remaining samples to the next stage for further screening based on the higher fidelity models. The surviving molecules up to the final stage are evaluated at the desired (highest in general) fidelity which is very accurate but computationally complex. Thanks to the capability of reducing search space efficiently, HTVS pipeline-based approach has been widely used in various studies including biology [66, 67, 68, 69, 70, 71], chemistry [72, 73, 74, 75, 76, 77, 63], and materials science [79, 80].

However, operational strategies for such HTVS pipelines have relied on expert intuition, often resulting in reasonable but sub-optimal performance of the HTVS pipelines. Recently, a mathematical optimization framework for optimizing the throughput and computational efficiency of the HTVS pipelines has been proposed [125] (Chapter. 3). The central idea of the proposed approach is to estimate the joint distribution of the scores that are either predicted via approximated models or computed through mathematical models with different fidelity. Based on the joint score distribution reflecting how screening stages are interrelated, the proposed approach defines the objective function and identifies the optimal screening policy. The proposed approach improved the computational efficiency of the HTVS pipelines by a significant margin while achieving a given operational objective–maximizing the number of promising molecules whose property meets a given condition at the desired fidelity–when the scores of a molecule of the screening stages are highly correlated.

Another practical issue in the HTVS pipeline-based screening campaign is to construct an HTVS pipeline effectively. Although the previous study provided insight on how one can improve the structure of the existing HTVS pipeline to further enhance the performance of the HTVS pipeline, the effective HTVS pipeline construction strategy has been a still open problem, especially when one is given a high-fidelity computational model and a pre-specified target screening threshold.

In this chapter, we design an optimal computational campaign for computationally efficient detection of organic electrode materials whose RP computed at desired fidelity model is within a

target range. To accomplish this, we propose an effective strategy for the construction of an HTVS pipeline based on a given high-fidelity density functional theory (DFT) computational model. To be specific, we decompose the high fidelity model into four sequential sub-models, each of which computes intermediate properties, such as HOMO, LUMO, HOMO-LUMO gap, and EA, that are needed to compute RP at the high fidelity. The sub-models form a skeleton structure of the HTVS pipeline. Then, we learn five surrogate models that serve as screening stages by predicting the RP using the intermediate properties that are available based on the location of the surrogate models within the HTVS pipeline. Besides, we introduce a concept of sub-surrogate models that predict the next available intermediate properties based on available features. The predicted properties are used as virtual features for the surrogate models to improve the predictive accuracy. Finally, we generalize the HTVS pipeline optimization framework proposed in the previous study [125] such that the framework is capable of optimizing the HTVS pipeline that screens the materials according to a target range, not a target threshold. We rigorously evaluate the performance of the optimized HTVS pipelines in various scenarios.

4.1 Overview of generalized HTVS pipeline

Figure 4.1 illustrates an overview of the proposed computational screening campaign design for the efficient detection of promising organic electrode materials. Formally, the operational objective of the campaign is to find subset $\mathbb{Y} = \{x \mid \lambda_L \leq f\,(x \in \mathbb{X}) \leq \lambda_U\}$ that consists of promising redox-active materials whose RP $f\,(x)$ computed via the given high fidelity DFT model f is within target screening range $\boldsymbol{\lambda} = [\lambda_L, \lambda_U]$ from huge initial material set $\mathbb{X}$. We assume that the target screening range $\boldsymbol{\lambda}$ is pre-specified by domain experts. As pointed out in [121], due to the excessive computational complexity of the high-fidelity model f, it is practically impossible to screen all the materials based solely on the high-fidelity model f. In order to overcome the fundamental issue, we propose a two-step optimal computational campaign design: First, we construct an HTVS pipeline structure by decomposing the high-fidelity model f into four sub-models $f_1, f_2, \ldots, f_4$ and learning five machine learning-based surrogate models $g_1, g_2, \ldots, g_5$ that serve as screening stages. Then, we identify the optimal screening policy $\boldsymbol{\psi}^* = \left[\lambda^*_{1,L}, \lambda^*_{1,U}, \ldots, \lambda^*_{N-1,U}\right]$ for the constructed

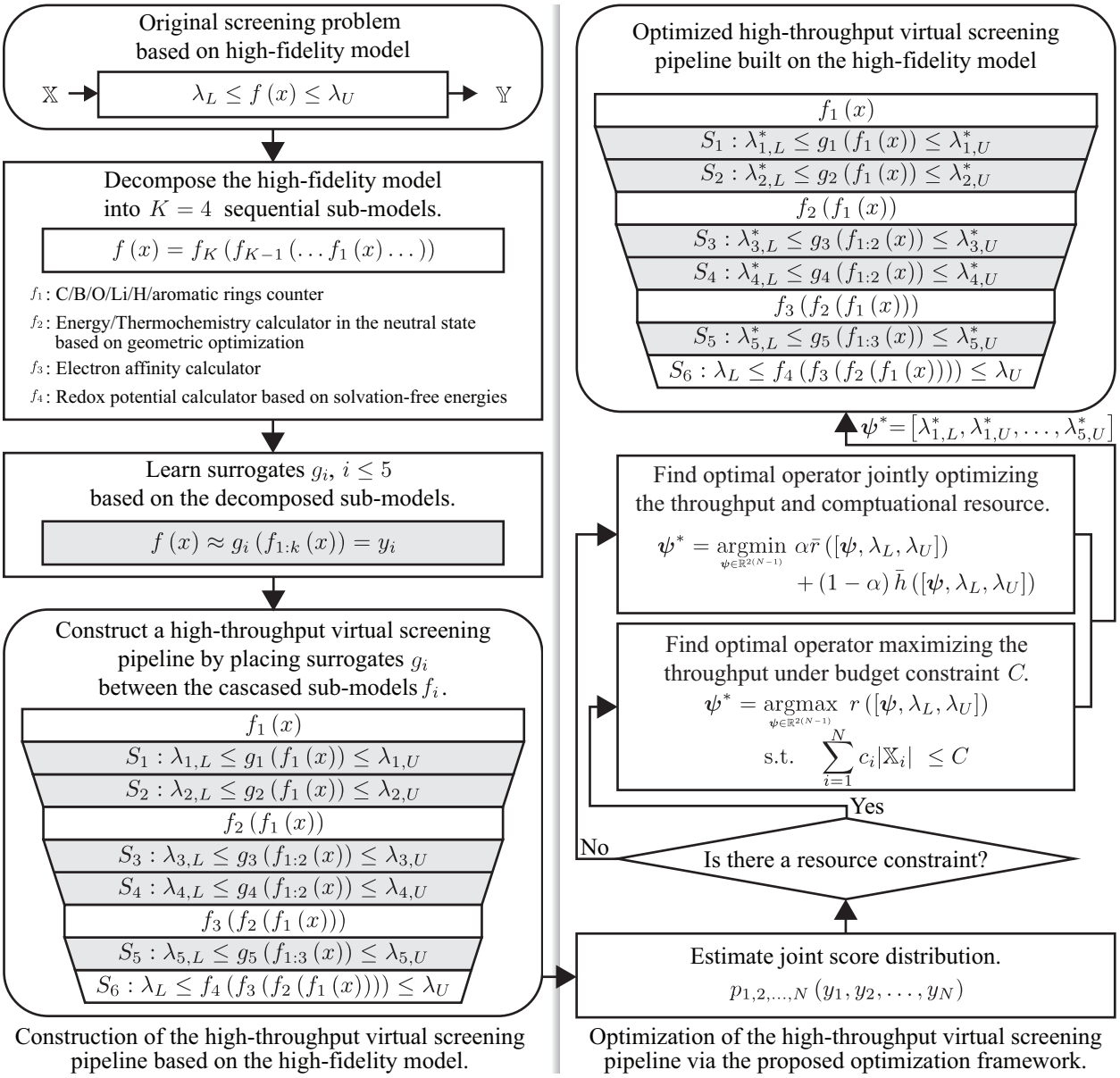


Figure 4.1: An overview of the proposed strategy based on a high-throughput virtual screening (HTVS) pipeline structure, where the primary operational objective is to efficiently detect promising organic electrode materials whose redox potential (RP) computed via the high fidelity density functional theory (DFT)-based model f is within pre-specified target range $[\lambda_L, \lambda_U]$. In the first phase (left panel), we decompose the high-fidelity model f into four sequential sub-models $f_1, f_2, \dots, f_4$, computing intermediate properties that are needed to compute RP at the high fidelity, to form a skeleton structure of the HTVS pipeline. Then, we learn the surrogate models g_i, $i = 1, 2, \dots, 5$ based on a different set of intermediate properties to build screening stages with different fidelity (left panel). In the second phase (right panel), we find the screening policy $\boldsymbol{\psi}^* = \left[\lambda^*_{1,L}, \lambda^*_{1,U}, \dots, \lambda^*_{N-1,U}\right]$ of the HTVS pipeline via the generalized optimization framework.

HTVS pipeline via the proposed optimization framework. We generalize the original optimization framework proposed in the previous study [125] such that the proposed framework can identify the optimal screening policy based on the target screening range, not a screening threshold.

The left panel of Figure 4.1 depicts the proposed HTVS pipeline construction strategy when one is given a high-fidelity computational model f with target screening range $\boldsymbol{\lambda} = [\lambda_L, \lambda_U]$. First, we decompose the high-fidelity DFT model f into four sequential sub-models f_i, $i = 1, 2, \ldots, 4$, each of which computes the intermediate properties of a material such as HOMO, LUMO, HOMO-LUMO gap, and EA. Then, we cascade the sub-models to construct the skeleton structure of the HTVS pipeline. Between the forms f_i and f_{i+1}, we learn up to two surrogate models g_j that predict the RP based on available intermediate properties as features. For the second surrogate model between the sub-models f_i and f_{i+1}, we learn sub-surrogate models $g_{j,l}$ that predict intermediate properties which will be computed via the following sub-model and use the predicted intermediate properties as features to improve the predictive accuracy of the surrogate model g_j. As shown in Figure 4.1 (left bottom), the resulting HTVS pipeline consists of five surrogate models, where surrogate model g_i is associated with screening stage S_i with screening policy $\boldsymbol{\lambda}_i = [\lambda_{i,L}, \lambda_{i,U}]$. Each stage S_i associated with surrogate model g_i or sub-module f_j predicts the RP of all the samples delivered from the previous stage S_{i-1}. Then, S_i discards the materials whose predicted potential is out of the corresponding range $\boldsymbol{\lambda}_i = [\lambda_{i,L}, \lambda_{i,U}]$ and passes the remaining samples for further DFT computation based on the remaining DFT computational steps marked in white. In this manner, we can gradually narrow down the search space while continuing to compute the intermediate features that are essential to computing RP at high fidelity for the surviving redox-active materials.

In the second phase, we find optimal screening policy $\boldsymbol{\psi}^* = \left[\lambda^*_{1,L}, \lambda^*_{1,U}, \ldots, \lambda^*_{N-1,U}\right]$ which is used for decision-making (whether pass the sample to the next stage or discard it) in screening stages associated with machine learning surrogates. To accomplish this, we generalize the original optimization framework proposed in [125] such that the generalized framework is capable of identifying the optimal screening policy for HTVS pipelines built for computational

screening campaigns with a target screening range. The proposed optimization framework is a two-step approach as shown in the right panel of Figure 4.1. First, we estimate joint distribution $p_{1,2,\ldots,N}(y_1, y_2, \ldots, y_N)$ of the RP values predicted via machine learning surrogate models or computed through the high-fidelity model. The joint score distribution provides information on how the screening stages are interrelated. Then, based on the joint score distribution $p_{1,2,\ldots,N}(y_1, y_2, \ldots, y_N)$, we formulate the objective function and find the optimal screening policy $\boldsymbol{\psi}^* = \left[\lambda_{1,L}^*, \lambda_{1,U}^*, \ldots, \lambda_{N-1,U}^*\right]$. In that regard, we considered two practical scenarios. We consider the case where we want to maximize the throughput of the HTVS pipeline with fixed computational budget constraint C. In the second case, the objective is to jointly optimize the throughput of the HTVS pipeline and computational efficiency. For the second scenario, we introduce weight $\alpha \in [0, 1]$ that determines the relative importance between the relative reward $\bar{r}(\boldsymbol{\psi}, \lambda_L, \lambda_U)$ and normalized cost function $\bar{h}(\boldsymbol{\psi}, \lambda_L, \lambda_U)$.

4.2 Methods

As shown in Figure 4.1, the proposed approach addresses two fundamental issues in computational screening campaigns based on an HTVS pipeline structure, designing an HTVS pipeline based on a high-fidelity model and finding the optimal screening policy.

As shown in Figure 4.2 (right panel), the high-fidelity DFT computational model computes several features of a molecule in neutral and anionic states to compute the RP. In order to construct the skeleton structure of the HTVS pipeline, we first decompose the high-fidelity model f into four computational modules $f_1, f_2, \ldots, f_4$ and cascade them sequentially, as shown in the right panel in Figure 4.2. For a given redox-active material, we first compute the primitive features such as the number of C, B, O, Li, H, and aromatic rings via f_1. Then, we compute the HOMO, LUMO, and HOMO-LUMO gap by combining a geometric optimizer and single-point energy/thermochemistry calculator for the material in a neutral state (f_2). Then, we compute the EA of the material based on the available intermediate features and geometrically optimized material in the neutral and anionic states via f_3. Finally, we calculate the solvation-free energies of the materials in both states to obtain the RP through f_4.

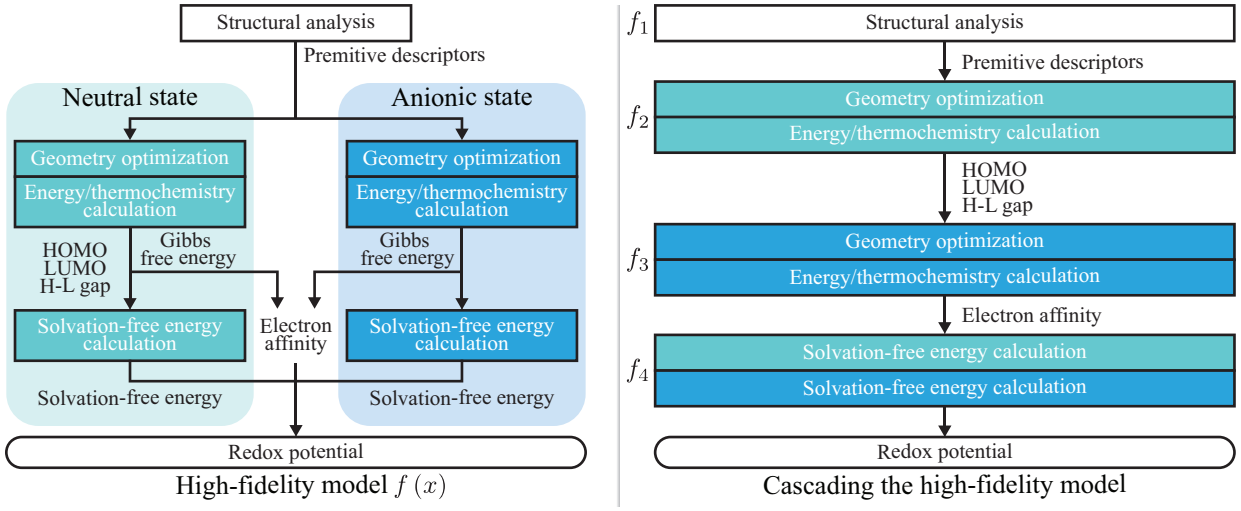


Figure 4.2: Illustration of constructing the skeleton structure of the HTVS pipeline based on the high-fidelity density functional theory (DFT) model.

Surrogate symbol	Model	Descriptors	Predicting property
g_1	Kernel ridge regressor (RBF)	Primitive features (PFs): #C, #B, #O, #Li, #H, # of aromatic rings	RP
g_2		PFs, HOMO (pred.), LUMO (pred.), HOMO-LUMO gap (pred.)	
g_3		PFs, HOMO, LUMO, HOMO-LUMO gap	
g_4		PFs, HOMO, LUMO, HOMO-LUMO gap, EA (pred.)	
g_5		PFs, HOMO, LUMO, HOMO-LUMO gap, EA	
$g_{2,1}$	Kernel ridge regressor (RBF)	PFs	HOMO
$g_{2,2}$			LUMO
$g_{2,3}$			HOMO-LUMO gap
$g_{4,1}$		PFs, HOMO, LUMO, HOMO-LUMO gap	EA

Table 4.1: Specifications of the surrogate models (1 to 5) and sub-surrogate models (2.1 to 2.3 and 4.1). Sub-surrogates predict intermediate properties used as virtual descriptors for the surrogate models to improve predictive capacity.

Based on the skeleton structure of the HTVS pipeline as shown in the right panel of Figure 4.2, we learn five surrogate models to build screening stages which will be placed between the sequential computational modules $f_1, f_2, \ldots, f_4$. The five surrogate models predict the RP using a different set of descriptors available based on the location of the surrogate model. For example, surrogate model g_1 located right after f_1 predicts the RP based on only the primitive descriptors. We introduce a concept of the sub-surrogate model that predicts the next available descriptors and use them as virtual features to improve the predictive accuracy of the surrogate models. For example, the second surrogate model g_2 located between g_1 and f_2 uses additional predicted features such as HOMO, LUMO, and HOMO-LUMO gap predicted via sub-surrogate models $g_{2,1}$, $g_{2,2}$, and $g_{2,3}$ in order to improve the prediction accuracy. Table 4.1 shows the specification of all the surrogate models trained in this study. We use a kernel ridge regression (KRR) model that effectively regresses the response in general (see Appendix G).

Similar to the original optimization framework proposed in [125], the first step of the generalized optimization framework for optimizing the performance of HTVS pipelines is to estimate joint score distribution p of the screening stages associated with machine learning-based surrogates g_i and the high-fidelity model f. In this study, we use parametric spectral estimation based on a multivariate Gaussian mixture model. Specifically, we estimate the parameters of the bi-modal multivariate Gaussian distribution via the expectation-maximization (EM) algorithm [100].

In the first computational campaign scenario, we assume that the operational objective is to maximize the number of organic electrode materials whose RP computed via a given high fidelity model f is within pre-specified target range $[\lambda_L, \lambda_U]$ under computational budget constraint C. To this aim, we identify the optimal screening policy $\boldsymbol{\psi}^* = \left[\lambda^*_{1,L}, \lambda^*_{1,U}, \ldots, \lambda^*_{N-1,U}\right]$ of the screening stages S_i, $i = 1, 2, \ldots, N-1$ associated with a machine learning-based surrogate f_i, such that the cardinality of the output set $\mathbb{Y}$ is maximized when target range $[\lambda_L, \lambda_U]$ of the last stage S_N and available computational budget C are given.

Let $p\left(y_1, y_2, \ldots, y_N\right)$ be a joint distribution of the RP values either computed via high-fidelity DFT model f or predicted through machine learning surrogate models g_i, $i = 1, 2, \ldots, 5$. Let us

denote the reward function $r(\boldsymbol{\lambda})$ according to screening ranges $\boldsymbol{\lambda}_{1:N} = [\lambda_{1,L}, \lambda_{1,U}, \lambda_{2,L}, \ldots, \lambda_U]$ of the screening stages S_i, $i = 1, 2, \ldots, N$, as follows:

$$r(\boldsymbol{\lambda}_{1:N}) = \int_{[\lambda_L, \lambda_{N-1,L}, \ldots, \lambda_{1,L}]}^{[\lambda_N, \lambda_{N-1,U}, \ldots, \lambda_{1,U}]} \cdots \int p(y_1, y_2, \ldots, y_N)\, dy_1 dy_2 \cdots dy_N. \tag{4.1}$$

Note that $r(\boldsymbol{\lambda}_{1:N})$ is proportional to the number of the potential samples that went through the screening pipeline.

We can find the optimal screening policy $\boldsymbol{\psi}^* = \left[\lambda^*_{1,L}, \lambda^*_{1,U}, \ldots, \lambda^*_{N-1,U}\right]$ of the surrogate-based stages S_i, $i = 1, 2, \ldots, N-1$, maximizing $|\mathbb{Y}|$, by solving the constrained optimization problem as follows:

$$\boldsymbol{\psi}^* = \underset{\boldsymbol{\psi} \in \mathbb{R}^{2(N-1)}}{\arg\max} \quad r([\boldsymbol{\psi}, \boldsymbol{\lambda}]) \tag{4.2}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} c_i |\mathbb{X}_i| \leq C, \tag{4.3}$$

where $|\mathbb{X}_i|$ is the number of samples that passed the previous stages from S_1 to S_{i-1}, defined as follows:

$$|\mathbb{X}_i| = |\mathbb{X}| \int_{[\lambda_{i-1,L}, \lambda_{i-2,L}, \ldots, \lambda_{1,L}]}^{[\lambda_{i-1,U}, \lambda_{i-2,U}, \ldots, \lambda_{1,U}]} \cdots \int p_{1:i-1}(y_1, y_2, \ldots, y_{i-1})\, dy_1 dy_2 \cdots dy_{i-1}, \tag{4.4}$$

where $p_{1:i-1}$ is a marginal score distribution by marginalizing over p of y_i to y_N.

In this scenario, we jointly optimize the throughput and computational resource consumption of the HTVS pipeline by solving the optimization problem as follows:

$$\boldsymbol{\psi}^* = \underset{\boldsymbol{\psi} \in \mathbb{R}^{2(N-1)}}{\arg\min}\ \alpha \bar{r}([\boldsymbol{\psi}, \boldsymbol{\lambda}]) + (1 - \alpha)\, \bar{h}([\boldsymbol{\psi}, \boldsymbol{\lambda}]), \tag{4.5}$$

where $\alpha \in [0, 1]$ is a weight parameter that determines the relative importance between the relative reward function $\bar{g}([\boldsymbol{\psi}, \boldsymbol{\lambda}])$ and the normalized total cost function $\bar{h}([\boldsymbol{\psi}, \boldsymbol{\lambda}])$ defined, respectively,

as follows:

$$\bar{r}\left([\boldsymbol{\psi}, \boldsymbol{\lambda}]\right) = \frac{r\left([-\infty, \infty, \ldots, \infty, \boldsymbol{\lambda}]\right) - r\left([\boldsymbol{\psi}, \boldsymbol{\lambda}]\right)}{r\left([-\infty, \infty, \ldots, \infty, \boldsymbol{\lambda}]\right)} \tag{4.6}$$

$$= \frac{\int_{\lambda_L}^{\lambda_U} p_N\left(y_N\right)\, dy_N - r\left([\boldsymbol{\psi}, \boldsymbol{\lambda}]\right)}{\int_{\lambda_{N,L}}^{\lambda_{N,U}} p_N\left(y_N\right)\, dy_N}, \tag{4.7}$$

$$\bar{h}\left([\boldsymbol{\psi}, \boldsymbol{\lambda}]\right) = \frac{1}{N|\mathbb{X}| \max_i c_i} \sum_{i=1}^{N} c_i |\mathbb{X}_i|, \tag{4.8}$$

where s_N is a marginal score distribution by marginalizing over p of y_1 to y_{N-1}.

4.3 Results and discussion

In order to validate the proposed computational screening design approach for detecting promising redox-active materials, we first collected 109 organic electrode materials designed in previous studies. [121, 123, 95, 92, 91, 93, 96, 99, 94]. Then, we computed electronic features of the materials, such as HOMO, LUMO, HOMO-LUMO gap, and RP (see Appendix F for further details).

Since our DFT dataset has been developed over multiple studies and under several different computational machines, we needed a method to fairly estimate the computational complexity (to calculate RP, as well as the input DFT features) for all the molecules in our dataset. Therefore, for consistency, we performed the necessary calculations on a single representative case, anthraquinone, and recorded the computational time. Using the computational time for this case and the well-known scaling factor for standard DFT computational complexity $O(N^3)$ [126, 127], we estimated the computational complexity for the remaining cases accordingly.

We performed simulations including learning surrogates and optimization on a system equipped with *Intel i7-8809G* and 32 GB memory. We utilized the differential evolution algorithm to optimize the HTVS pipelines. We evaluated the time complexity of a representative molecule on *Intel Xeon E5-2650 v3* and 64 GB memory.

To evaluate the efficacy of the proposed HTVS construction strategy, we computed Pearson's correlation of the RP values computed via the high-fidelity model f and predicted through sur-

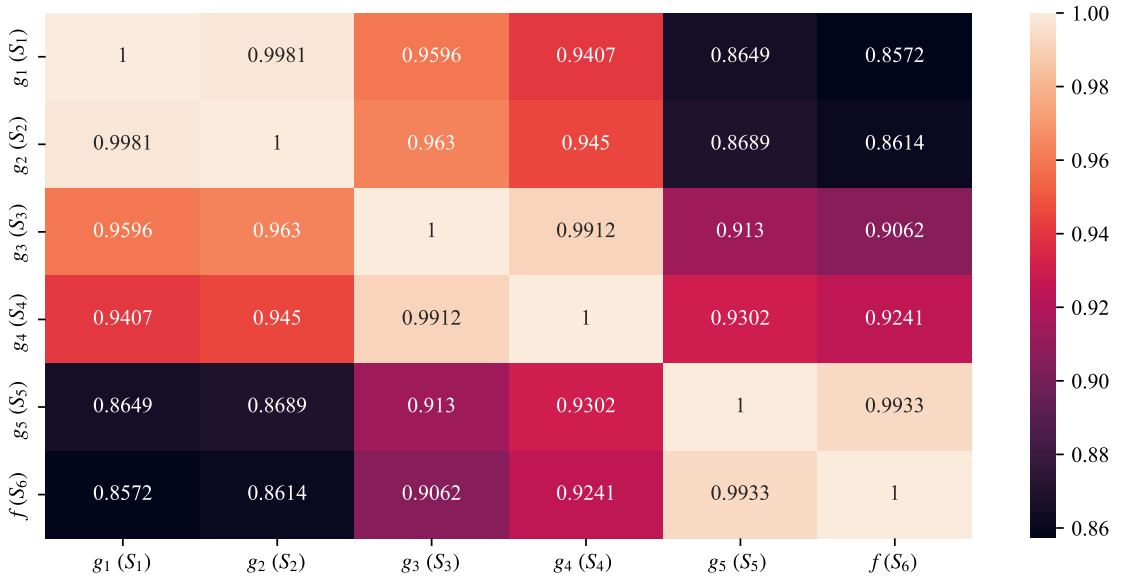


Figure 4.3: Pearson's correlation of the RP values either computed via the high-fidelity DFT model f or predicted surrogate models g_i, $i = 1, 2, \ldots, 5$. As we used more descriptors, the correlation of the RP predicted via the surrogate models in comparison to the high-fidelity DFT model increased as we expected. Note that the predicted descriptors via the sub-surrogate model helped improve the regression performance.

rogate models g_i, $i = 1, 2, \ldots, 5$. We used a KRR model that effectively regresses the response in general (see Appendix G). We optimized hyperparameters of each surrogate via a grid search based on 5-fold cross-validation (see Appendix H). Note that we used all the materials to learn the surrogate models as our major concern was not to design the best surrogate models. However, for completeness, we provide the performance evaluation results of all the computational screening scenario considered in this study based on a strict 5-fold cross-validation (see Appendix I). As shown in Figure 4.3, the correlation of the RP values between the surrogate model g_i and the high-fidelity model f gradually increased as the number of used properties increased. Specifically, the correlation between the RP values predicted via the first surrogate model g_1 that uses only the primitive features to the RP values computed via the high-fidelity model f is 0.8572. Interestingly, the predicted HOMO, LUMO, and HOMO-LUMO gap via the sub-surrogate models $g_{2,1}$, $g_{2,2}$, and $g_{2,3}$ help increase the correlation of the second surrogate model g_2, showing a correlation of

0.8614. Similarly, the predicted EA via sub-surrogate model $g_{4,1}$ helped improve the performance of the surrogate model by 0.0179. Lastly, the last surrogate model that utilizes all the chemical descriptors showed the highest correlation with respect to the high-fidelity model f. These simulation results clearly show that the proposed HTVS construction strategy is effective to configure the surrogate models that serve as screening stages with the increasing order of regression accuracy and computational complexity.

To evaluate the performance of the optimized HTVS pipeline, we first considered a realistic computational screening scenario where the operational objective is to effectively select the organic redox-active materials whose RP computed at high fidelity is above target threshold 2.5 V vs. Li/Li$^+$ (*i.e.*, $\lambda = [2.5 \text{ V}, \infty \text{ V}]$) which is exhibited by many organic cathode materials under a typical voltage window of $1 \sim 4$ V vs. Li/Li$^+$ [128].

Figure 4.4 shows the performance evaluation results of the optimized HTVS pipeline under a computational resource constraint in seconds (x-axis) in terms of sensitivity, specificity, F1 score, and accuracy based on a 5-fold cross-validation. Sensitivity is a ratio of the detected potential candidates whose RP exceeds or is equal to the minimum target threshold of 2.5 V to all the promising materials in the test dataset. Specificity is defined as a ratio of the discarded negative molecules to negative samples. F1 score is a harmonic mean between the positive predictive rate and specificity. Lastly, accuracy is a correctly selected promising material ratio. The shaded area along each performance curve represents the standard deviation of the performance on the five cross-validation datasets. The optimized HTVS pipeline effectively distributed a given computational budget over the stages and maximized throughput (*i.e.*, the number of promising redox-active materials meeting the given condition). On average, the optimized HTVS pipeline selected all potential materials with only 84.11% of the original computational resource budget ($6,286,056$, blue vertical line) that requires for screening all the organic materials via the high fidelity model f. Besides, 80% of potential materials were detected with 58.62% of the original budget. Note that specificity was always 1 throughout the simulation as the redox-active materials were screened based on the high fidelity model in the last stage S_6. In other words, all the remaining negative samples arriving at

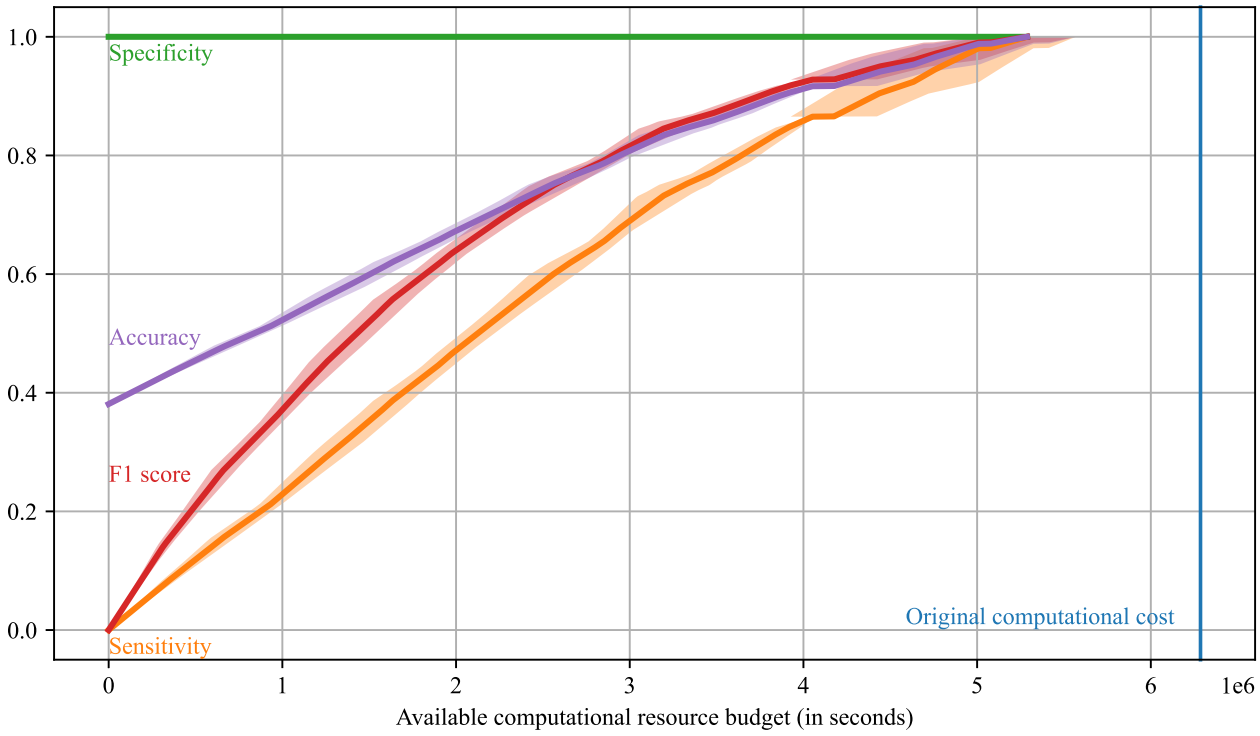


Figure 4.4: Performance evaluation of the optimized HTVS pipeline based on a 5-fold cross-validation. The shaded area along each curve represents the standard deviation of the performance on the five cross-validation datasets. The optimal screening policy maximized the throughput (*i.e.*, the number of potential candidates whose RP exceeds or is equal to the target threshold of 2.5 V) under computational budget constraints (x-axis). The optimized HTVS pipeline effectively allocated the computational resource over the multiple screening stages, thereby detecting all the potential candidates at only 84.11% of the original computational cost of $6,286,056$ (blue vertical line) which would be required if solely the high fidelity model f were used for screening.

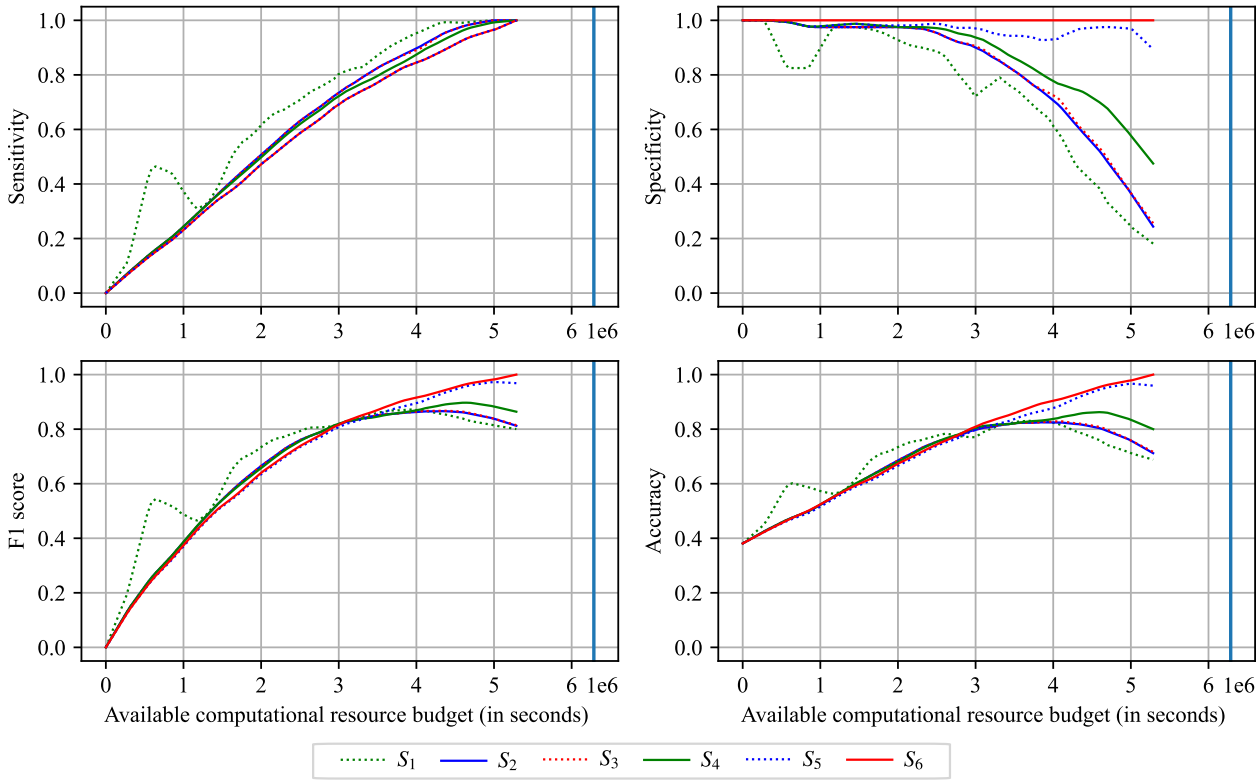


Figure 4.5: Performance evaluation of the individual stages constituting the HTVS pipeline based on a 5-fold cross-validation. In general, sensitivity tended to increase as the allocated computational budget increased. On the other hand, the specificity of the stages (except for the last stage) tended to decrease as the allocated resource increased. This was because the earlier stages were designed to pass a larger number of candidates to later stages as the available budget grew, in order to evaluate and screen the materials with higher accuracy. For the same reason, the F1 score and the accuracy generally increased as the computational budget grew, but they eventually decreased due to the increasing false-positive rates as a result of passing too many candidates to subsequent stages.

the final stage were discarded. For this reason, the F1 score reached 1 on average where sensitivity became 1. We could observe a similar trend in accuracy. Specifically, the accuracy reached 80.17% when only 46.73% of computational resources were given. The pipeline achieved perfect accuracy at the cost of $5,287,453$ seconds (84.11% of the original cost) on average.

Figure 4.5 illustrates the performance evaluation results at each stage in the optimized HTVS pipeline in terms of sensitivity, specificity, F1 score, and accuracy based on a 5-fold cross-validation. The sensitivity of the screening stages tended to increase as the computational resource budget rose.

For a given computational budget, the sensitivity of S_i was always greater than or equal to that of later stages $S_{j>i}$. This was due to the structure of the HTVS pipeline that the later stages processed only the materials delivered from the previous screening stages. On the other hand, the specificity except for the final stage tended to decrease as the available budget increased. In other words, the earlier screening stages allowed the later stages with higher accuracy to involve the screening campaign more as the computational resource grew. As a result, the F1 score tended to increase sharply at the beginning but slowed down the tendency to rise later. The trend of the accuracy was similar to that of the F1 score due to the same reason. The accuracy of the earlier stages eventually fell since they passed too many materials to later stages, resulting in higher false-positive rates. Note that the performance fluctuation of the preceding stages was relatively severe, while the performance of the HTVS pipeline (*i.e.*, S_6) showed a very gentle change, implying that the optimal screening policy was not unique from a numerical point of view.

Figure 4.6 shows the number of discarded materials at each stage in the optimized HTVS pipeline with respect to an available computational budget (x-axis) based on 5-fold cross-validation. On average, the first stage S_1 (left top, green dotted line) predicting the RP based on only primitive features, such as the numbers of various atoms and aromatic rings, contributed to significantly screening materials when the available computational budget was limited. As the computational budget increased, the number of molecules discarded in the first stage gradually decreased, allowing subsequent screening stages to involve in screening with higher accuracy. For example, the surrogate models discarded 75.09, 4.62, 0, 0, 0.21, and 0.01 materials, respectively, when there were only $320,452$ (5.1% of the original computational cost) computational resources. With a computational resource of $5,287,453$, the screening stages eliminated 5.8, 5.8, 0.4, 7.0, 13.4, and 3.4 materials, respectively. During the simulation, each stage rejected 37.72, 6.42, 0.13, 2.23, 4.13, and 0.95 materials on average, respectively.

Table 4.2 shows the performance evaluation results of the jointly optimized HTVS pipeline with various α in terms of detected materials, total cost (in seconds), effective cost, sensitivity, specificity, F1 score, and accuracy based on a 5-fold cross-validation. $\alpha \in [0, 1]$ is a parameter

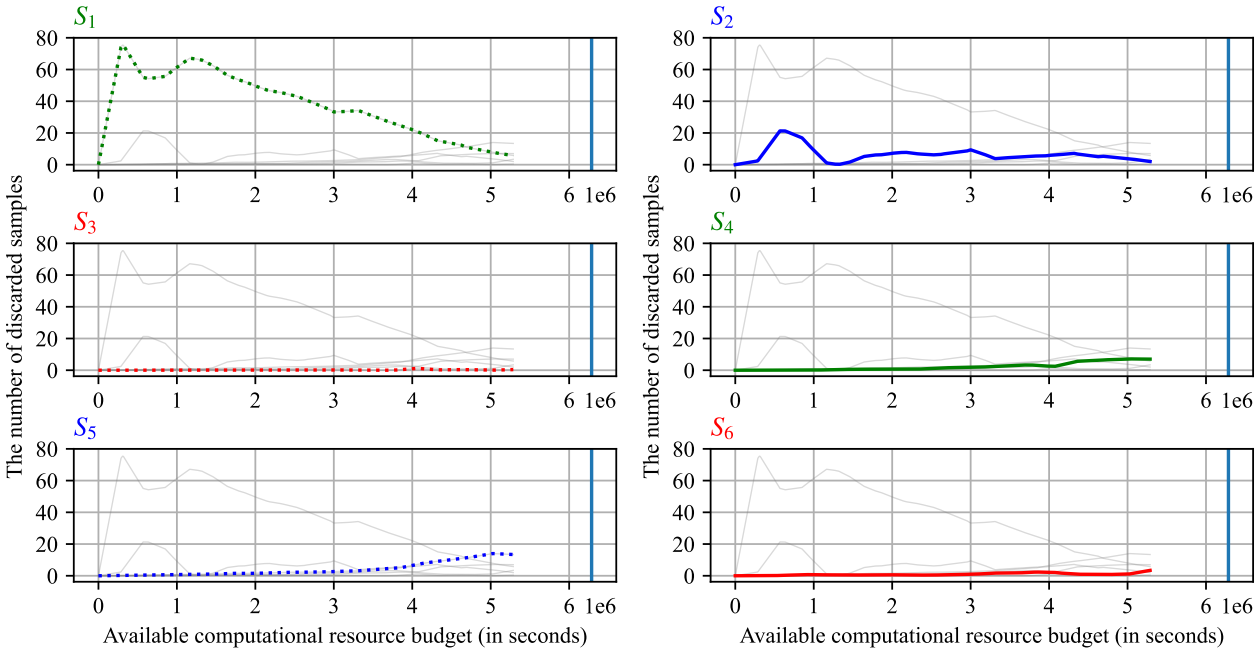


Figure 4.6: The number of discarded molecules at each screening stage with respect to available computational resource budget (x-axis) based on a 5-fold cross-validation. The first stage S_1 (left top, green dotted line) that predicts the RP based only on primitive features filtered out a significant proportion of candidates when the computational budget was tightly constrained. As the computational budget increased, the number of molecules discarded at the first stage gradually decreased, allowing subsequent higher-accuracy stages to get more actively involved in screening.

α	Detected materials	Total cost (seconds)	Effective cost (seconds)	Sensitivity	Specificity	F1 score	Accuracy
0.25	40.4	$3,450,440$	$85,407$	0.7769	1	0.8714	0.8619
0.5	47.8	$4,365,990$	$91,339$	0.9192	1	0.9574	0.95
0.75	49.8	$4,645,890$	$93,291$	0.9577	1	0.9782	0.9738

Table 4.2: Performance evaluation of the jointly optimized HTVS pipeline based on a 5-fold cross-validation (target RP threshold at the last stage set to 2.5 V). As α weighting between the throughput and computational efficiency increased from 0.25 to 0.75, all the throughput-related performance metrics tended to improve at the cost of higher computational requirements (*i.e.*, increased total cost and effective cost). Overall, the optimized HTVS pipeline struck a good balance between throughput and computational efficiency.

that weights between the throughput and computational efficiency of the pipeline. The detected materials is the cardinality of the resulting subset set $\mathbb{Y}$. Total cost is the amount of time to screen input set $\mathbb{X}$ and the effective cost is defined as the ratio of the total cost to the detected materials. Sensitivity is a ratio of the number of selected materials to the number of total promising redox-active materials in the test dataset. Specificity is defined as a ratio of the discarded samples that do not meet the desired criterion to negative samples. F1 score is a harmonic mean between the positive predictive rate and specificity, and accuracy is a correctly classified material ratio. As α increased from 0.25 to 0.75, the number of selected materials whose RP value at high fidelity is greater than or equal to 2.5 V rose from 40.4 to 49.8 out of 52 promising organic electrode materials. To be specific, on average, the pipeline picked 49.8 out of 52 promising materials when α was 0.75 at the effective cost of $93,291$. When alpha was 0.25, the optimized HTVS pipeline detected 40.4 samples at the effective cost of $85,407$. In terms of saving computational resources, although the total computational cost and effective cost grew when α changed from 0.25 to 0.75, the overall computational complexity was significantly less than that of the original computational cost of $6,286,056$ and original effective cost of $120,886$, respectively. Besides, other evaluation metrics, including accuracy, sensitivity, and F1 score noticeably improved when α increased. Overall, the optimized HTVS pipeline with various α found an efficient and reasonable consensus between throughput and computational complexity.

Next, we considered a more practical computational screening campaign scenario where the operational objective of the campaign is to detect promising redox-active materials whose RP computed at the desired fidelity is within a target range. In fact, higher RP of organic cathode materials is desirable for increasing the output voltage of a Li-ion cell. However, the peak voltage could be constrained due to limiting factors such as the thermodynamic stability of organic electrolyte material. Based on our previous works, we selected 3.2 V vs. Li/Li^+ as the target upper bound for the computational screening campaign, resulting in targeting screening range $[2.5\text{ V}, 3.2\text{ V}]$.

Figure 4.7 demonstrates the performance evaluation results of the optimized HTVS pipeline under a computational resource budget constraint (x-axis) for detecting organic electrode materi-

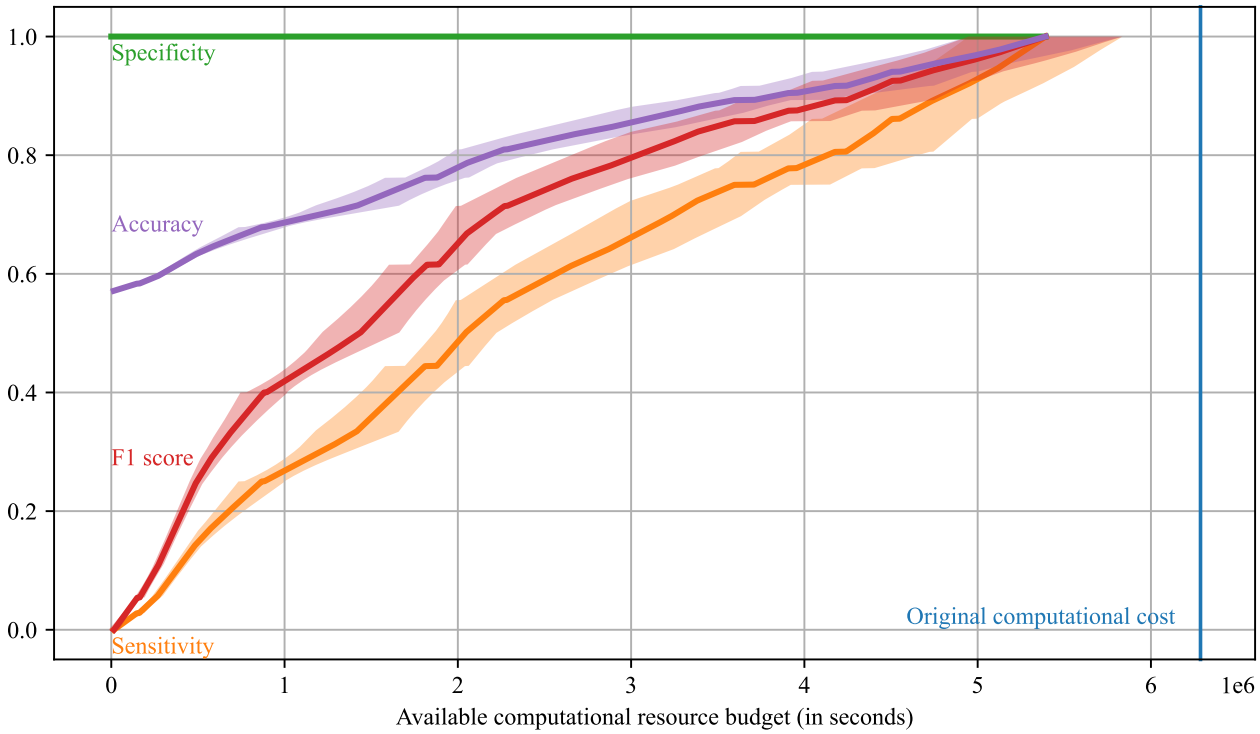


Figure 4.7: Performance evaluation of the optimized HTVS pipeline that aims to detect promising redox-active materials whose RP at the desired fidelity is within the target range $[2.5\ \text{V}, 3.2\ \text{V}]$ based on a 5-fold cross-validation. The average performance metrics are shown as a function of the total available computational budget (x-axis). The shaded area along each performance curve represents the standard deviation of the performance on the five cross-validation datasets. The optimized HTVS pipeline detected all promising materials that meet the target screening condition at only 85.78% of the original computational cost (blue vertical line) that would be required for screening all materials solely based on the high fidelity model f. The HTVS pipeline built on the high fidelity model achieved perfect specificity regardless of the available computational budget.

als in terms of sensitivity, specificity, F1 score, and accuracy based on a 5-fold cross-validation. Similarly, the shaded area of each performance curve stands for the standard deviation of the performance on the five cross-validation datasets. As shown in Figure 4.7, the HTVS pipeline optimized via the generalized optimization framework effectively allocated a computational budget to the stages, thereby saving a significant amount of computational resources without (or with minimal) throughput loss for the screening campaign according to the target RP range $[2.5\text{ V}, 3.2\text{ V}]$. On average, the optimized pipeline consisting of five surrogates saved 14.22% of the original computational resources $6,286,056$ (in seconds) for detecting chemical compounds via the high fidelity model f. Besides, the optimized pipeline selected 80% of the promising organic electrode materials with only 65.85% of the original cost. Note that the HTVS pipeline always guaranteed perfect sensitivity (*i.e.*, 1) as the HTVS pipeline was built based on the high fidelity model f. Therefore, the envelope of the F1 score with respect to the computational resource constraint (*i.e.*, x-axis) had a similar shape to that of sensitivity. In terms of accuracy, the HTVS pipeline trivially assured the accuracy of 0.5714 which is equal to the ratio of the negative samples. The accuracy of the optimized HTVS pipeline gradually increased as the available computational budget increased and achieved the perfect accuracy at the computational resource of $5,391,914$ (85.78% of the original cost).

Figure 4.8 demonstrates the performance evaluation results of the stages in the optimized HTVS pipeline detecting the organic electrode materials according to the target RP range of $[2.5\text{ V}, 3.2\text{ V}]$ based on a 5-fold cross-validation. We could observe a similar trend to the previous computational campaign scenario. The sensitivity of the screening stages tended to increase as the computational resource budget increased. Similarly, for a given computational budget, the sensitivity of S_i was always greater than or equal to those of later stages $S_{j>i}$. For example, the stages $S_1, S_2, \ldots, S_6$ achieved the sensitivity of 0.9874, 0.8716, 0.8698, 0.8130, 0.6882, and 0.6882, respectively, when the available computational complexity was $3,158,898$ (50.25% of the original computational cost). Again, we could observe that the specificity except for the final stage decreased as the available budget rose. As a result, both the F1 score and accuracy tended to increase

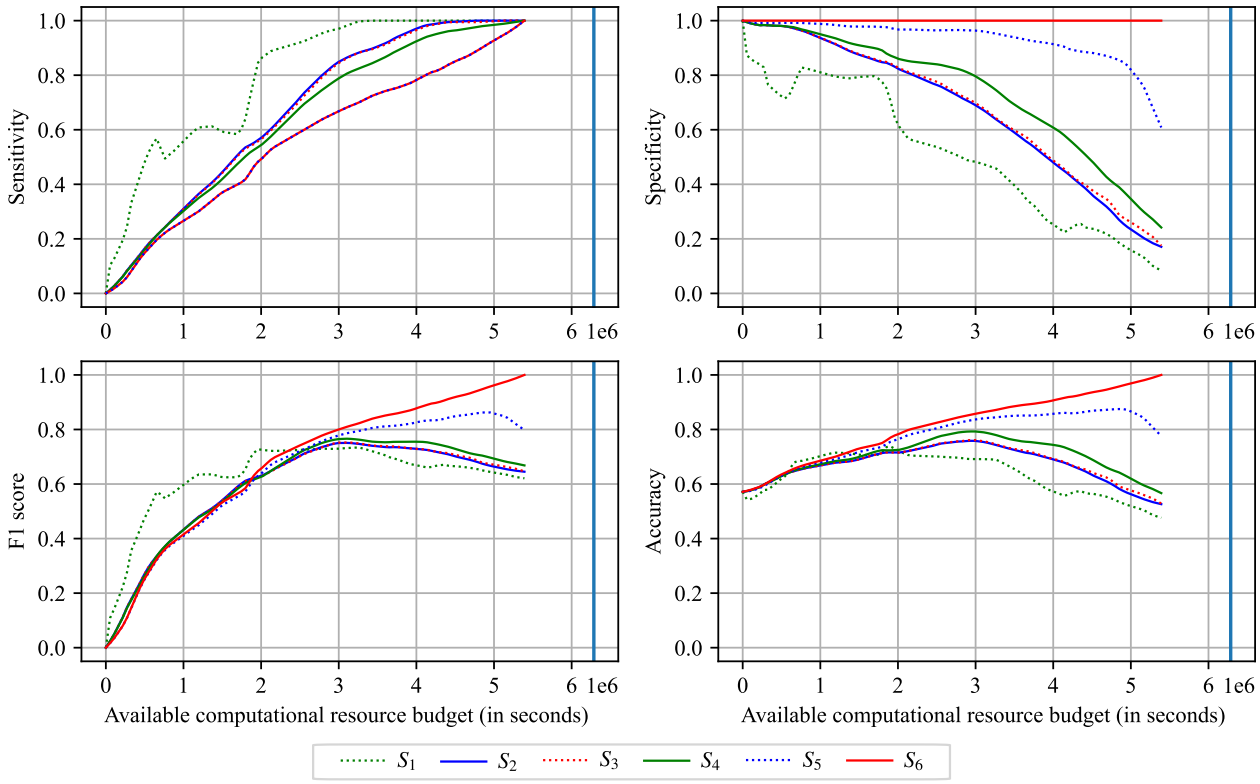


Figure 4.8: Performance evaluation of the screening stages in the optimized HTVS pipeline designed to detect the organic electrode materials according to target RP range $[2.5\ \text{V}, 3.2\ \text{V}]$ based on a 5-fold cross-validation. As before, the sensitivity of the screening stages tended to increase as the computational budget (x-axis) grew. In general, the specificity decreased as the available resource rose (except for the last stage). The F1 score and accuracy improved as the available budget got larger, but they eventually decreased due to the increasing false-positive rates due to passing too many materials to the later stages.

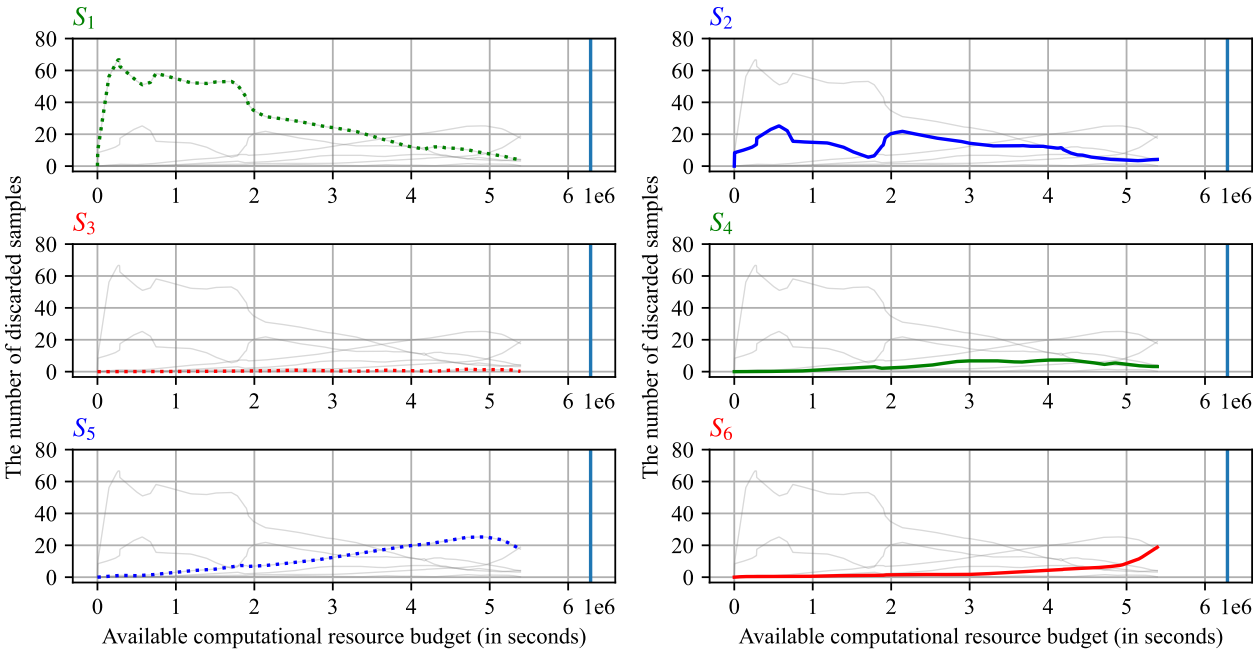


Figure 4.9: The number of molecules that were discarded at each stage for the case when the target RP range was set to $[2.5\text{ V}, 3.2\text{ V}]$. The results were obtained based on a 5-fold cross-validation for a given computational resource constraint (x-axis). As before, when the computational budget was tightly limited, the most efficient first stage S_1 (top left, green dotted curve) filtered out a significant number of redox-active materials and passed only candidate materials that are expected to satisfy the target screening condition at the high fidelity. In general, the number of molecules discarded in the first stage decreased gradually as the computational budget increased, allowing subsequent screening stages with higher accuracy to get more involved in screening.

sharply but eventually decreased later.

Figure 4.9 illustrates the number of discarded redox-active materials at each stage in the optimized HTVS pipeline based on a 5-fold cross-validation. Similarly, as the available computational budget increased, the number of materials discarded in the first stage S_1 gradually decreased. For example, the screening stages $S_1, S_2, \ldots, S_6$ discarded 66.72, 13.23, 0, 0.08, 1.17, and 0.4 materials, respectively, when there were only 27.2319 (4.33% of the original computational cost) computational resources. With computational resource of $5,391,914$ (85.78% of the original cost), the screening stages dropped 4.0, 4.2, 0.2, 3.2, 17.6, and 18.8, respectively. During the simulation, each stage rejected 30.06, 12.45, 0.5, 3.85, 11.8, and 2.2 materials on average, respectively.

Table 4.3 shows the performance evaluation results of the HTVS pipeline jointly optimized via

α	Detected materials	Total cost (seconds)	Effective cost (seconds)	Sensitivity	Specificity	F1	Accuracy
0.25	23.4	$2,923,471$	$124,935$	0.65	1	0.7689	0.85
0.5	29.2	$3,921,249$	$134,289$	0.8111	1	0.8892	0.9190
0.75	29.8	$4,257,906$	$142,883$	0.8278	1	0.9003	0.9262

Table 4.3: Performance evaluation of the jointly optimized HTVS pipeline based on a 5-fold cross-validation, where the target RP range was set to $[2.5\ \text{V}, 3.2\ \text{V}]$. As α increased, the overall throughput of the HTVS pipeline increased with the higher consumption of the computational resources (*i.e.*, increased total cost and effective cost). As before, the optimized HTVS pipeline struck a good balance between throughput and computational efficiency.

the proposed optimization framework with different values of α based on a 5-fold cross-validation. We could observe a similar trend compared to the computational campaign detecting the potential materials according to the target screening threshold (*i.e.*, $[2.5\ \text{V}, \infty\ \text{V}]$). As α increased, all the throughput quality metrics improved at the cost of higher computational complexity (*i.e.*, total cost and effective cost). When α was 0.25, the jointly optimized pipeline operated conservatively from the perspective of resource utilization. To be specific, the optimized pipeline with $\alpha = 0.25$ consumed $2,923,471$ seconds for detecting 23.4 promising compounds out of 36 promising organic electrode materials in the test datasets. On the other hand, the optimized HTVS pipeline with $\alpha = 0.75$ selected 29.8 promising candidates, on average, whose RP is between 2.5 V and 3.2 V at the cost of $4,257,906$. Overall, the jointly optimized HTVS pipeline found an efficient and reasonable consensus between throughput and computational complexity according to the value of α

4.4 Concluding remarks

In this chapter, we designed the optimal computational screening campaigns, where the operational objective is to construct a subset of promising organic electrode materials whose RP computed at high fidelity meets the desired condition from a huge initial material set. At the essence of the proposed design lie the HTVS construction strategy when a high fidelity model f is given and the generalized HTVS pipeline optimization framework. As shown in Figure 4.1, we first decomposed the high fidelity model into four sub-components f_i, $i = 1, 2, \ldots, 4$, that compute

intermediate properties of a material, such as HOMO, LUMO, HOMO-LUMO gap, or EA. Then, we cascaded them to construct a skeleton structure of the HTVS pipeline. Based on the structure, we learned five machine learning surrogate models that predict the RP with available intermediate descriptors according to the locations of the surrogate models. Surrogate model g_i was associated with screening stage S_i with screening policy $[\lambda_{i,L}, \lambda_{i,U}]$ in order to pass only materials that are likely to meet the desired condition at the high fidelity to the next stage S_{i+1} for further RP computation, thereby significantly saving computational resources. Besides, we introduced a concept of the sub-surrogate model that predicts the next available descriptors and used them as virtual features to improve the predictive accuracy of the surrogate models. In the second phase, we optimized the optimal screening policy of the stages that are associated with machine learning-based surrogate models through the optimization framework. Specifically, we found optimal screening ranges $\left[\lambda^*_{i,L}, \lambda^*_{i,U}\right]$ of stages S_i, $i = 1, 2, \ldots, 5$, which leads to the optimal performance of the HTVS pipeline. To this aim, we further generalized the original optimization framework for HTVS pipelines proposed in [125], which enables optimizing HTVS pipelines screening materials according to target range, in addition to target threshold.

We validated the proposed approach by optimizing the constructed HTVS pipeline for a screening campaign whose operational goal is to maximally detect promising redox-active materials according to the target RP threshold set to 2.5 V. As shown in Figure 4.4, the optimized pipeline consumed 84.11% of the original computational resources to detect all the promising redox-active materials at the desired fidelity. The HTVS pipeline consumed only 58.62% of the original computational cost to find 80% of the potential materials. Then, we found the optimal screening policy that jointly optimized the throughput and computational efficiency. According to the α, the pipeline found 77.69% to 95.77% of the promising redox-active materials with the accuracy of 86.19% to 97.38% at the effective computational cost of $85,407$ to $93,291$.

We also validated the proposed approach based on the computational screening campaign where the objective is to efficiently detect organic redox-active materials whose RP computed at the high fidelity is within the target range ($[2.5$ V$, 3.2$ V$]$). We utilized the same HTVS pipeline struc-

ture that was used for the first computational screening campaign. As shown in Figure 4.7, when a computational resource budget is given, the optimized HTVS pipeline selected all the promising organic electrode materials at 85.78% of the original cost. In case the available computational resource was not fixed, we jointly optimized the HTVS pipeline for optimizing the throughput and computational efficiency according to the value of α. Specifically, when alpha was set to 0.75, the optimized HTVS pipeline found 29.8 potential candidates (82.78% of the promising potential compounds in the test dataset) while consuming only $4,257,906$ seconds. The optimized pipeline with $\alpha = 0.25$ selected 65% of the potential candidates at the cost of $2,923,471$.

Besides, based on the simulation result shown in Figure 4.3, we repeated the simulations on the identical setup with updated the structure of the HTVS pipeline. Specifically, we discarded the first and third stages (*i.e.*, S_1 and S_3) from the original HTVS pipeline structure and optimized the pipeline $[S_2, S_4, S_5, S_6]$ accordingly. The comprehensive simulation results showed that discarding computationally very efficient and moderately correlated screening stage does not significantly impact the performance of the optimized HTVS pipeline (see Appendix J). However, it should be noted that reducing redundant stages reduces the dimensionality of the score joint distribution, potentially resulting in higher density estimation accuracy. Besides, one might save computational resources in screening operations and avoid the burden of training surrogate models.

It should be noted that the computational screening campaign considered in this study has a fundamental performance bound that we cannot overcome, as we were interested in selecting the promising materials whose property computed at the high fidelity model meets a pre-specified condition. For example, when a test dataset contains only promising materials, in order to achieve our fundamental objective which is to detect all the promising molecules in an initial search space, we eventually have to compute all the molecules in the high-fidelity model. In fact, the positive sample ratio of the first and second computational campaigns in this study were 0.381 and 0.4286, respectively, which affects the overall performance of the optimized pipeline. However, in real-world screening campaigns, the positive sample ratio is extremely small, and the proposed approach showed an impressive performance improvement as shown in the original study [125]. In fact,

we evaluated the performance of the optimized pipeline based on a strict 5-fold cross-validation when the target screening threshold is set to 4.3 resulting in only one promising redox-active material out of 21 samples in the test deadset. The optimized HTVS pipeline identified the promising redox-active material at the cost of 18.78% of the original computational cost (see Appendix K).

Another critical factor affecting the computational complexity of the optimized HTVS pipeline is the ratio of the positive samples satisfying a given screening condition due to the nature of the designed experiment built on the decomposed highest fidelity model. In other words, the chemical compounds arriving at the final stage are evaluated at the highest fidelity. In the worst case that there are only positive samples in the test dataset, the overall computational complexity of the optimized HTVS pipeline is slighter higher than that of the screening campaign based solely on the highest fidelity model due to the prediction cost of the surrogates. However, in real-world applications, the positive sample ratio is more than often extremely small, which makes the designed experiments more effective and practical.

It is worthwhile noting that the performance of the HTVS pipeline optimized via the proposed framework is also dependent on the predictive power of the surrogate models. In this study, we utilized the KRR model for predicting the RP. One possible way to further improve the performance of the HTVS pipeline is to employ deep-learning models having high predictive potential. In that regard, other highly structured descriptors or features, such as simplified molecular input line entry system (SMILES) [129] or self-referencing embedded strings (SELFIES) [130], can be considered.

5. CONCLUDING REMARKS

In this dissertation, we addressed optimal decision-making problems to accelerate scientific discoveries for scientific and engineering applications that often involve intense computations. In the first part of the dissertation, we proposed a machine learning (ML) approach that significantly accelerates the optimal decision-making in mean objective cost of uncertainty (MOCU)-based optimal experimental design (OED). One fundamental obstacle that has reduced the applicability of MOCU-based OED is excessive computational complexity in identifying the optimal experiment. Specifically, in order to prioritize the experiments to conduct, the expected remaining uncertainty affecting the operational performance (*i.e.*, MOCU) of each experiment needs to be quantified, which is computationally complex in practice. The proposed approach is to replace the part of complex computations in quantifying the remaining uncertainty with an ML surrogate model that regresses or classifies efficiently. We validated the proposed approach in the context of OED for robust control of uncertain Kuramoto models by replacing the computational costly differential equation (DE) solver with an ML model, which remarkably speeds up the process of predicting the optimal controller. The trained ML model classifies the asymptotic behavior of a given Kuramoto model, namely, whether all oscillators in the model will be eventually frequency synchronized or not.

It is worth noting that the proposed approach can be generalized and applied to other MOCU-based OED problems concerning scientific and engineering applications that do not possess closed-form (remaining expected) MOCU. In such cases, based on the applications and the data types, one may consider various ML/deep learning (DL) models including convolutional neural network (CNN) [131], recurrent neural network (RNN) [132], long short term memory (LSTM) [133], and graph convolutional network (GCN) [134]. In fact, as a follow-up study, we developed a deep surrogate model based on the neural message-passing model [135], that directly predicts the MOCU of an uncertain Kuramoto model [136].

A potential direction for future research is to utilize scientific ML models to learn scientific

knowledge from data. Scientific knowledge is a fundamental building block for defining experimental space, which affects the computational complexity and the efficacy of MOCU-based OED. In the context of robust synchronization of the uncertain Kuramoto model, we considered pairwise synchronization experiments as the necessary and sufficient condition for frequency synchronization of the Kuramoto models is limited to two-oscillator models. Discovering relational knowledge regarding the model parameters via ML can lead to the design of more effective experiments and a significant expansion of the potential experimental design space.

In the second part of the dissertation, we focused on an optimal decision-making problem in the context of operating high-throughput virtual screening (HTVS) pipelines to accelerate the optimal selection of potential molecular candidates whose property meets specific criteria at the desired fidelity. To this aim, we first generalized and formulated the structure and operations of HTVS pipelines. Based on this, we designed a mathematical optimization framework that identifies the optimal screening policy that leads to the optimal performance of HTVS pipelines. The optimal screening policy at each stage is used to decide whether the evaluation result is advantageous enough to pass it to the next stage without unnecessarily wasting computational resources and time.

We validated the proposed optimization framework for computational screening campaigns based on HTVS pipelines on both synthetic and real-world datasets. The comprehensive simulation results demonstrated that the proposed framework identified the optimal policy that maximizes the throughput or jointly maximizes the throughput and computational efficiency of the HTVS pipelines according to the computational screening scenarios. Besides, the proposed optimization framework was effective in alleviating the performance fluctuations depending on the structures of HTVS pipelines. In the third part of the dissertation, based on the optimization framework in the second chapter, we designed an optimal computational campaign (OCC) intending to efficiently select redox-active organic materials whose redox potential (RP) satisfies a specific condition at the desired fidelity. To this aim, we proposed an effective strategy for building an HTVS pipeline structure based on the high fidelity model. Specifically, we decomposed the fidelity model into

sequential forms that compute several intermediate chemical properties such as highest occupied molecular orbital (HOMO), lowest unoccupied molecular orbital (LUMO), HOMO-LUMO gap, and electron affinity (EA). Then, we learned a set of surrogate models, each of which serves as a screening stage or feature predictor. Besides, we further generalized the screening condition and the optimization framework accordingly such that the optimized HTVS pipeline selects the promising materials according to the target screening threshold, not a threshold. Comprehensive simulation with various scenarios on a real dataset showed that the proposed HTVS pipeline remarkably enhances the overall throughput for a given computational budget.

REFERENCES

[1] E. R. Dougherty, *Optimal signal processing under uncertainty*. SPIE Press, 2018.

[2] B.-J. Yoon, X. Qian, and E. R. Dougherty, "Quantifying the objective cost of uncertainty in complex dynamical systems," *IEEE Transactions on Signal Processing*, vol. 61, no. 9, pp. 2256–2266, 2013.

[3] S. S. Baboo and I. K. Shereef, "An efficient weather forecasting system using artificial neural network," *International journal of environmental science and development*, vol. 1, no. 4, p. 321, 2010.

[4] B.-H. Chen, S.-C. Huang, C.-Y. Li, and S.-Y. Kuo, "Haze removal using radial basis function networks for visibility restoration applications," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3828–3838, 2017.

[5] A. Shukla, R. Tiwari, P. Kaur, and R. Janghel, "Diagnosis of thyroid disorders using artificial neural networks," in *2009 IEEE International Advance Computing Conference*, pp. 1016–1020, IEEE, 2009.

[6] D. J. Hemanth, J. Anitha, L. H. Son, and M. Mittal, "Diabetic retinopathy diagnosis from retinal images using modified hopfield neural network," *Journal of medical systems*, vol. 42, no. 12, pp. 1–6, 2018.

[7] K. M. Fanning and K. O. Cogger, "A comparative analysis of artificial neural networks using financial distress prediction," *Intelligent Systems in Accounting, Finance and Management*, vol. 3, no. 4, pp. 241–252, 1994.

[8] M. Dixon, D. Klabjan, and J. H. Bang, "Classification-based financial markets prediction using deep neural networks," *Algorithmic Finance*, vol. 6, no. 3-4, pp. 67–77, 2017.

[9] J. Smits, W. Melssen, L. Buydens, and G. Kateman, "Using artificial neural networks for solving chemical problems: Part i. multi-layer feed-forward networks," *Chemometrics and*

Intelligent Laboratory Systems, vol. 22, no. 2, pp. 165–189, 1994.

[10] J. Meiler, "Proshift: protein chemical shift prediction using artificial neural networks," *Journal of biomolecular NMR*, vol. 26, no. 1, pp. 25–37, 2003.

[11] A. Basu and E. B. Bartlett, "Detecting faults in a nuclear power plant by using dynamic node architecture artificial neural networks," *Nuclear Science and Engineering*, vol. 116, no. 4, pp. 313–325, 1994.

[12] E. Zio, "A study of the bootstrap method for estimating the accuracy of artificial neural networks in predicting nuclear transient processes," *IEEE Transactions on Nuclear Science*, vol. 53, no. 3, pp. 1460–1478, 2006.

[13] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[15] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[16] A. L. Strehl, L. Li, E. Wiewiora, J. Langford, and M. L. Littman, "Pac model-free reinforcement learning," in *Proceedings of the 23rd international conference on Machine learning*, pp. 881–888, 2006.

[17] H. R. Maei, C. Szepesvári, S. Bhatnagar, and R. S. Sutton, "Toward off-policy learning control with function approximation," in *ICML*, 2010.

[18] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.

[19] C. C. Aggarwal *et al.*, "Neural networks and deep learning," *Springer*, vol. 10, pp. 978–3, 2018.

[20] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[21] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[22] P. Niu, M. J. Soto, B.-J. Yoon, E. R. Dougherty, F. J. Alexander, I. Blaby, and X. Qian, "TRIMER: Transcription regulation integrated with metabolic regulation," *iScience*, p. 103218, 2021.

[23] F. Shen, R. Sun, J. Yao, J. Li, Q. Liu, N. D. Price, C. Liu, and Z. Wang, "OptRAM: In-silico strain design via integrative regulatory-metabolic network modeling," *PLoS computational biology*, vol. 15, no. 3, p. e1006835, 2019.

[24] E. R. Dougherty, "Scientific epistemology in the context of uncertainty," in *Berechenbarkeit der Welt?*, pp. 129–154, Springer, 2017.

[25] E. R. Dougherty, L. A. Dalton, and R. Dehghannasiri, *Objective Uncertainty Quantification*, pp. 541–560. Springer International Publishing, 2019.

[26] B.-J. Yoon, X. Qian, and E. R. Dougherty, "Quantifying the multi-objective cost of uncertainty," *IEEE Access*, vol. 9, pp. 80351–80359, 2021.

[27] Y. Hong, B. Kwon, and B.-J. Yoon, "Optimal experimental design for uncertain systems based on coupled differential equations," *IEEE Access*, vol. 9, pp. 53804–53810, 2021.

[28] G. Zhao, X. Qian, B.-J. Yoon, F. J. Alexander, and E. R. Dougherty, "Model-based robust filtering and experimental design for stochastic differential equation systems," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3849–3859, 2020.

[29] R. Dehghannasiri, B.-J. Yoon, and E. R. Dougherty, "Efficient experimental design for uncertainty reduction in gene regulatory networks," *BMC Bioinformatics*, vol. 16, no. 13, p. S2, 2015.

[30] R. Dehghannasiri, B.-J. Yoon, and E. R. Dougherty, "Optimal experimental design for gene regulatory networks in the presence of uncertainty," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 12, no. 4, pp. 938–950, 2015.

[31] A. Broumand, M. S. Esfahani, B.-J. Yoon, and E. R. Dougherty, "Discrete optimal bayesian classification with error-conditioned sequential sampling," *Pattern Recognition*, vol. 48, no. 11, pp. 3766–3782, 2015.

[32] G. Zhao, E. Dougherty, B.-J. Yoon, F. Alexander, and X. Qian, "Bayesian active learning by soft mean objective cost of uncertainty," in *24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

[33] G. Zhao, E. Dougherty, B.-J. Yoon, F. Alexander, and X. Qian, "Uncertainty-aware active learning for optimal Bayesian classifier," in *9th International Conference on Learning Representations (ICLR)*, 2021.

[34] G. Zhao, E. Dougherty, B.-J. Yoon, F. J. Alexander, and X. Qian, "Efficient active learning for gaussian process classification by error reduction," in *Thirty-fifth Conference on Neural Information Processing Systems*, 2021.

[35] Y. Kuramoto, "Self-entrainment of a population of coupled non-linear oscillators," in *International symposium on mathematical problems in theoretical physics*, pp. 420–422, Springer, 1975.

[36] K. Wiesenfeld, P. Colet, and S. Strogatz, "Frequency locking in josephson arrays: Connection with the kuramoto model," *Physical Review E*, vol. 57, pp. 1563–1569, feb 1998.

[37] Z. Néda, E. Ravasz, T. Vicsek, Y. Brechet, and A. L. Barabási, "Physics of the rhythmic applause," *Physical Review E*, vol. 61, pp. 6987–6992, jun 2000.

[38] C. Hammond, H. Bergman, and P. Brown, "Pathological synchronization in parkinson's disease: networks, models and treatments," *Trends in neurosciences*, vol. 30, no. 7, pp. 357–364, 2007.

[39] M. G. Kitzbichler, M. L. Smith, S. R. Christensen, and E. Bullmore, "Broadband criticality of human brain network synchronization," *PLoS Comput Biol*, vol. 5, no. 3, p. e1000314, 2009.

[40] M. Breakspear, S. Heitmann, and A. Daffertshofer, "Generative models of cortical oscillations: Neurobiological implications of the kuramoto model," *Frontiers in Human Neuroscience*, vol. 4, 2010.

[41] D. Bhowmik and M. Shanahan, "How well do oscillator models capture the behaviour of biological neurons?," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, IEEE, jun 2012.

[42] J. W. Simpson-Porco, F. Dörfler, and F. Bullo, "Droop-controlled inverters are kuramoto oscillators," *IFAC Proceedings Volumes*, vol. 45, no. 26, pp. 264–269, 2012.

[43] B. Fernandez, D. Gérard-Varet, and G. Giacomin, "Landau damping in the kuramoto model," *Annales Henri Poincaré*, vol. 17, pp. 1793–1823, dec 2015.

[44] P. S. Skardal and A. Arenas, "Control of coupled oscillator networks with application to microgrid technologies," *Science advances*, vol. 1, no. 7, p. e1500339, 2015.

[45] A. Mohseni, S. Gharibzadeh, and F. Bakouie, "The role of driver nodes in managing epileptic seizures: Application of kuramoto model," *Journal of theoretical biology*, vol. 419, pp. 108–115, 2017.

[46] H. Choi and S. Mihalas, "Synchronization dependent on spatial structures of a mesoscopic whole-brain network," *PLoS computational biology*, vol. 15, no. 4, p. e1006978, 2019.

[47] Y. Guo, D. Zhang, Z. Li, Q. Wang, and D. Yu, "Overviews on the applications of the kuramoto model in modern power system analysis," *International Journal of Electrical Power & Energy Systems*, vol. 129, p. 106804, 2021.

[48] M. Rohden, A. Sorge, M. Timme, and D. Witthaut, "Self-organized synchronization in decentralized power grids," *Phys. Rev. Lett.*, vol. 109, p. 064101, Aug 2012.

[49] A. Motter, S. Myers, M. Anghel, and T. Nishikawa, "Spontaneous synchrony in power-grid networks," *Nature Physics*, vol. 9, pp. 191–197, Mar. 2013.

[50] J. W. Simpson-Porco, F. Dorfler, and F. Bullo, "Synchronization and power sharing for droop-controlled inverters in islanded microgrids," *Automatica*, vol. 49, no. 9, pp. 2603–2611, 2013.

[51] F. Dörfler, M. Chertkov, and F. Bullo, "Synchronization in complex oscillator networks and smart grids," *Proceedings of the National Academy of Sciences*, vol. 110, no. 6, pp. 2005–2010, 2013.

[52] K. Lehnertz, S. Bialonski, M.-T. Horstmann, D. Krug, A. Rothkegel, M. Staniek, and T. Wagner, "Synchronization phenomena in human epileptic brain networks," *Journal of neuroscience methods*, vol. 183, no. 1, pp. 42–48, 2009.

[53] J. C. Pang, L. L. Gollo, and J. A. Roberts, "Stochastic synchronization of dynamics on the human connectome," *NeuroImage*, vol. 229, p. 117738, 2021.

[54] A. Arenas, A. Díaz-Guilera, J. Kurths, Y. Moreno, and C. Zhouu, "Synchronization in complex networks," *Physics Reports*, vol. 469, no. 3, pp. 93 – 153, 2008.

[55] F. A. Rodrigues, T. K. D. Peron, P. Ji, and J. Kurths, "The kuramoto model in complex networks," *Physics Reports*, vol. 610, pp. 1–98, 2016. The Kuramoto model in complex networks.

[56] J. A. Acebrón, L. L. Bonilla, C. J. Pérez Vicente, F. Ritort, and R. Spigler, "The kuramoto model: A simple paradigm for synchronization phenomena," *Rev. Mod. Phys.*, vol. 77, pp. 137–185, Apr 2005.

[57] S. Boluki, M. S. Esfahani, X. Qian, and E. R. Dougherty, "Incorporating biological prior knowledge for bayesian learning via maximal knowledge-driven information priors," *BMC bioinformatics*, vol. 18, no. 14, pp. 61–80, 2017.

[58] S. Boluki, M. S. Esfahani, X. Qian, and E. R. Dougherty, "Constructing pathway-based priors within a gaussian mixture model for bayesian regression and classification," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 16, no. 2, pp. 524–537, 2017.

[59] I. Lagaris, A. Likas, and D. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987–1000, 1998.

[60] J. Han, A. Jentzen, and W. E, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 34, pp. 8505–8510, 2018.

[61] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[62] M. Allen, D. Poggiali, K. Whitaker, T. R. Marshall, and R. A. Kievit, "Raincloud plots: a multi-platform tool for robust data visualization," *Wellcome open research*, vol. 4, 2019.

[63] A. A. Saadi, D. Alfe, Y. Babuji, A. Bhati, B. Blaiszik, A. Brace, T. Brettin, K. Chard, R. Chard, A. Clyde, *et al.*, "Impeccable: integrated modeling pipeline for covid cure by assessing better leads," in *50th International Conference on Parallel Processing*, pp. 1–12, 2021.

[64] B. Roy, J. Dhillon, N. Habib, and B. Pugazhandhi, "Global variants of covid-19: Current understanding," *Journal of Biomedical Sciences*, vol. 8, no. 1, pp. 8–11, 2021.

[65] T. Sterling and J. J. Irwin, "Zinc 15–ligand discovery for everyone," *Journal of chemical information and modeling*, vol. 55, no. 11, pp. 2324–2337, 2015.

[66] N. Rieber, B. Knapp, R. Eils, and L. Kaderali, "Rnaither, an automated pipeline for the statistical analysis of high-throughput rnai screens," *Bioinformatics*, vol. 25, no. 5, pp. 678–679, 2009.

[67] M. H. Studer, J. D. DeMartini, S. Brethauer, H. L. McKenzie, and C. E. Wyman, "Engineering of a high-throughput screening system to identify cellulosic biomass, pretreatments, and enzyme formulations that enhance sugar release," *Biotechnology and Bioengineering*, vol. 105, no. 2, pp. 231–238, 2010.

[68] A. Hartmann, T. Czauderna, R. Hoffmann, N. Stein, and F. Schreiber, "Htpheno: an image analysis pipeline for high-throughput plant phenotyping," *BMC bioinformatics*, vol. 12, no. 1, pp. 1–9, 2011.

[69] K. Sikorski, A. Mehta, M. Inngjerdingen, F. Thakor, S. Kling, T. Kalina, T. A. Nyman, M. E. Stensland, W. Zhou, G. A. de Souza, *et al.*, "A high-throughput pipeline for validation of antibodies," *Nature methods*, vol. 15, no. 11, pp. 909–912, 2018.

[70] A. Clyde, S. Galanie, D. W. Kneller, H. Ma, Y. Babuji, B. Blaiszik, A. Brace, T. Brettin, K. Chard, R. Chard, *et al.*, "High throughput virtual screening and validation of a sars-cov-2 main protease non-covalent inhibitor," *bioRxiv*, 2021.

[71] A. Clyde, T. Brettin, A. Partin, H. Yoo, Y. Babuji, B. Blaiszik, A. Merzky, M. Turilli, S. Jha, A. Ramanathan, *et al.*, "Protein-ligand docking surrogate models: A sars-cov-2 benchmark for deep learning accelerated virtual screening," *arXiv preprint arXiv:2106.07036*, 2021.

[72] R. L. Martin, C. M. Simon, B. Smit, and M. Haranczyk, "In silico design of porous polymer networks: high-throughput screening for methane storage materials," *Journal of the American Chemical Society*, vol. 136, no. 13, pp. 5006–5022, 2014.

[73] L. Cheng, R. S. Assary, X. Qu, A. Jain, S. P. Ong, N. N. Rajput, K. Persson, and L. A. Curtiss, "Accelerating electrolyte discovery for energy storage with high-throughput screening," *The journal of physical chemistry letters*, vol. 6, no. 2, pp. 283–291, 2015.

[74] J. J. F. Chen and D. P. Visco Jr, "Developing an in silico pipeline for faster drug candidate discovery: Virtual high throughput screening with the signature molecular descriptor using support vector machine models," *Chemical Engineering Science*, vol. 159, pp. 31–42, 2017.

[75] D. L. Filer, P. Kothiya, R. W. Setzer, R. S. Judson, and M. T. Martin, "tcpl: the toxcast pipeline for high-throughput screening data," *Bioinformatics*, vol. 33, no. 4, pp. 618–620, 2017.

[76] Y. Li, J. Zhang, N. Wang, H. Li, Y. Shi, G. Guo, K. Liu, H. Zeng, and Q. Zou, "Therapeutic drugs targeting 2019-ncov main protease by high-throughput screening," *BioRxiv*, 2020.

[77] R. T. Rebbeck, D. P. Singh, K. A. Janicek, D. M. Bers, D. D. Thomas, B. S. Launikonis, and R. L. Cornea, "Ryr1-targeted drug discovery pipeline integrating fret-based high-throughput screening and human myofiber dynamic ca 2+ assays," *Scientific reports*, vol. 10, no. 1, pp. 1–13, 2020.

[78] A. Tran, T. Wildey, and S. McCann, "smf-bo-2cogp: A sequential multi-fidelity constrained bayesian optimization framework for design applications," *Journal of Computing and Information Science in Engineering*, vol. 20, no. 3, 2020.

[79] Q. Yan, J. Yu, S. K. Suram, L. Zhou, A. Shinde, P. F. Newhouse, W. Chen, G. Li, K. A. Persson, J. M. Gregoire, *et al.*, "Solar fuels photoanode materials discovery by integrating high-throughput theory and experiment," *Proceedings of the National Academy of Sciences*, vol. 114, no. 12, pp. 3040–3043, 2017.

[80] B. Zhang, X. Zhang, J. Yu, Y. Wang, K. Wu, and M.-H. Lee, "First-principles high-throughput screening pipeline for nonlinear optical materials: Application to borates," *Chemistry of Materials*, vol. 32, no. 15, pp. 6772–6779, 2020.

[81] C. Chen, Y. Wang, Y. Xia, B. Wu, D. Tang, K. Wu, Z. Wenrong, L. Yu, and L. Mei, "New development of nonlinear optical crystals for the ultraviolet region with molecular engineering approach," *Journal of applied physics*, vol. 77, no. 6, pp. 2268–2272, 1995.

[82] G. Shi, Y. Wang, F. Zhang, B. Zhang, Z. Yang, X. Hou, S. Pan, and K. R. Poeppelmeier, "Finding the next deep-ultraviolet nonlinear optical material: Nh4b4o6f," *Journal of the American Chemical Society*, vol. 139, no. 31, pp. 10645–10648, 2017.

[83] B. Zhang, G. Shi, Z. Yang, F. Zhang, and S. Pan, "Fluorooxoborates: beryllium-free deep-ultraviolet nonlinear optical materials without layered growth," *Angewandte Chemie International Edition*, vol. 56, no. 14, pp. 3916–3919, 2017.

[84] M. Luo, F. Liang, Y. Song, D. Zhao, F. Xu, N. Ye, and Z. Lin, "M2b10o14f6 (m= ca, sr): Two noncentrosymmetric alkaline earth fluorooxoborates as promising next-generation

deep-ultraviolet nonlinear optical materials," *Journal of the American Chemical Society*, vol. 140, no. 11, pp. 3884–3887, 2018.

[85] M. Mutailipu, M. Zhang, B. Zhang, L. Wang, Z. Yang, X. Zhou, and S. Pan, "Srb5o7f3 functionalized with [b5o9f3] 6- chromophores: Accelerating the rational design of deep-ultraviolet nonlinear optical materials," *Angewandte Chemie*, vol. 130, no. 21, pp. 6203–6207, 2018.

[86] Y. Wang, B. Zhang, Z. Yang, and S. Pan, "Cation-tuned synthesis of fluorooxoborates: Towards optimal deep-ultraviolet nonlinear optical materials," *Angewandte Chemie*, vol. 130, no. 8, pp. 2172–2176, 2018.

[87] Z. Zhang, Y. Wang, B. Zhang, Z. Yang, and S. Pan, "Cab5o7f3: A beryllium-free alkaline-earth fluorooxoborate exhibiting excellent nonlinear optical performances," *Inorganic chemistry*, vol. 57, no. 9, pp. 4820–4823, 2018.

[88] C. E. Wilmer, M. Leaf, C. Y. Lee, O. K. Farha, B. G. Hauser, J. T. Hupp, and R. Q. Snurr, "Large-scale screening of hypothetical metal–organic frameworks," *Nature chemistry*, vol. 4, no. 2, pp. 83–89, 2012.

[89] S. Gupta, D. Parihar, M. Shah, S. Yadav, H. Managori, S. Bhowmick, P. C. Patil, S. A. Alissa, S. M. Wabaidur, and M. A. Islam, "Computational screening of promising beta-secretase 1 inhibitors through multi-step molecular docking and molecular dynamics simulations-pharmacoinformatics approach," *Journal of Molecular Structure*, vol. 1205, p. 127660, 2020.

[90] D. Y. Kim, M. Ha, and K. S. Kim, "A universal screening strategy for the accelerated design of superior oxygen evolution/reduction electrocatalysts," *Journal of Materials Chemistry A*, vol. 9, no. 6, pp. 3511–3519, 2021.

[91] T. Liu, K. C. Kim, R. Kavian, S. S. Jang, and S. W. Lee, "High-density lithium-ion energy storage utilizing the surface redox reactions in folded graphene films," *Chemistry of Materials*, vol. 27, no. 9, pp. 3291–3298, 2015.

[92] K. C. Kim, T. Liu, S. W. Lee, and S. S. Jang, "First-principles density functional theory modeling of li binding: thermodynamics and redox properties of quinone derivatives for lithium-ion batteries," *Journal of the American Chemical Society*, vol. 138, no. 7, pp. 2374–2382, 2016.

[93] S. Kim, K. C. Kim, S. W. Lee, and S. S. Jang, "Thermodynamic and redox properties of graphene oxides for lithium-ion battery applications: a first principles density functional theory modeling approach," *Physical Chemistry Chemical Physics*, vol. 18, no. 30, pp. 20600–20606, 2016.

[94] T. Liu, K. C. Kim, B. Lee, Z. Chen, S. Noda, S. S. Jang, and S. W. Lee, "Self-polymerized dopamine as an organic cathode for li-and na-ion batteries," *Energy & Environmental Science*, vol. 10, no. 1, pp. 205–215, 2017.

[95] J. H. Park, T. Liu, K. C. Kim, S. W. Lee, and S. S. Jang, "Systematic molecular design of ketone derivatives of aromatic molecules for lithium-ion batteries: First-principles dft modeling," *ChemSusChem*, vol. 10, no. 7, pp. 1584–1591, 2017.

[96] J. Kang, K. C. Kim, and S. S. Jang, "Density functional theory modeling-assisted investigation of thermodynamics and redox properties of boron-doped corannulenes for cathodes in lithium-ion batteries," *The Journal of Physical Chemistry C*, vol. 122, no. 20, pp. 10675–10681, 2018.

[97] P. Sood, K. C. Kim, and S. S. Jang, "Electrochemical and electronic properties of nitrogen doped fullerene and its derivatives for lithium-ion battery applications," *Journal of energy chemistry*, vol. 27, no. 2, pp. 528–534, 2018.

[98] P. Sood, K. C. Kim, and S. S. Jang, "Electrochemical properties of boron-doped fullerene derivatives for lithium-ion battery applications," *ChemPhysChem*, vol. 19, no. 6, pp. 753–758, 2018.

[99] Y. Zhu, K. C. Kim, and S. S. Jang, "Boron-doped coronenes with high redox potential for organic positive electrodes in lithium-ion batteries: a first-principles density functional

theory modeling study," *Journal of Materials Chemistry A*, vol. 6, no. 21, pp. 10111–10120, 2018.

[100] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[101] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[102] S.-Y. Ng, L. Lin, B. S. Soh, and L. W. Stanton, "Long noncoding rnas in development and disease of the central nervous system," *Trends in Genetics*, vol. 29, no. 8, pp. 461–468, 2013.

[103] L. Tan, J.-T. Yu, N. Hu, and L. Tan, "Non-coding rnas in alzheimer's disease," *Molecular neurobiology*, vol. 47, no. 1, pp. 382–393, 2013.

[104] Q. Luo and Y. Chen, "Long noncoding rnas and alzheimer's disease," *Clinical interventions in aging*, vol. 11, p. 867, 2016.

[105] A. Congrains, K. Kamide, R. Oguro, O. Yasuda, K. Miyata, E. Yamamoto, T. Kawai, H. Kusunoki, H. Yamamoto, Y. Takeya, *et al.*, "Genetic variants at the 9p21 locus contribute to atherosclerosis through modulation of anril and cdkn2a/b," *Atherosclerosis*, vol. 220, no. 2, pp. 449–455, 2012.

[106] Z. Xue, S. Hennelly, B. Doyle, A. A. Gulati, I. V. Novikova, K. Y. Sanbonmatsu, and L. A. Boyer, "A g-rich motif in the lncrna braveheart interacts with a zinc-finger transcription factor to specify the cardiovascular lineage," *Molecular cell*, vol. 64, no. 1, pp. 37–50, 2016.

[107] G. Yang, X. Lu, and L. Yuan, "Lncrna: a link between rna and cancer," *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, vol. 1839, no. 11, pp. 1097–1109, 2014.

[108] X. Shi, M. Sun, H. Liu, Y. Yao, R. Kong, F. Chen, and Y. Song, "A critical role for the long non-coding rna gas5 in proliferation and apoptosis in non-small-cell lung cancer," *Molecular carcinogenesis*, vol. 54, no. S1, pp. E1–E12, 2015.

[109] W.-X. Peng, P. Koirala, and Y.-Y. Mo, "Lncrna-mediated regulation of cell signaling in cancer," *Oncogene*, vol. 36, no. 41, pp. 5661–5667, 2017.

[110] J. Carlevaro-Fita, A. Lanzós, L. Feuerbach, C. Hong, D. Mas-Ponte, J. S. Pedersen, and R. Johnson, "Cancer lncrna census reveals evidence for deep functional conservation of long noncoding rnas in tumorigenesis," *Communications biology*, vol. 3, no. 1, pp. 1–16, 2020.

[111] L. Wang, H. J. Park, S. Dasari, S. Wang, J.-P. Kocher, and W. Li, "Cpat: Coding-potential assessment tool using an alignment-free logistic regression model," *Nucleic acids research*, vol. 41, no. 6, pp. e74–e74, 2013.

[112] A. Li, J. Zhang, and Z. Zhou, "Plek: a tool for predicting long non-coding rnas and messenger rnas based on an improved k-mer scheme," *BMC bioinformatics*, vol. 15, no. 1, pp. 1–10, 2014.

[113] Y.-J. Kang, D.-C. Yang, L. Kong, M. Hou, Y.-Q. Meng, L. Wei, and G. Gao, "Cpc2: a fast and accurate coding potential calculator based on sequence intrinsic features," *Nucleic acids research*, vol. 45, no. W1, pp. W12–W16, 2017.

[114] S. Han, Y. Liang, Q. Ma, Y. Xu, Y. Zhang, W. Du, C. Wang, and Y. Li, "Lncfinder: an integrated platform for long non-coding rna identification utilizing sequence intrinsic composition, structural information and physicochemical property," *Briefings in bioinformatics*, vol. 20, no. 6, pp. 2009–2027, 2019.

[115] A. Frankish, M. Diekhans, I. Jungreis, J. Lagarde, J. E. Loveland, J. M. Mudge, C. Sisu, J. C. Wright, J. Armstrong, I. Barnes, *et al.*, "Gencode 2021," *Nucleic acids research*, vol. 49, no. D1, pp. D916–D923, 2021.

[116] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.

[117] Y. Liang, Z. Tao, and J. Chen, "Organic electrode materials for rechargeable lithium batteries," *Advanced Energy Materials*, vol. 2, no. 7, pp. 742–769, 2012.

[118] Z. Song and H. Zhou, "Towards sustainable and versatile energy storage devices: an overview of organic electrode materials," *Energy & Environmental Science*, vol. 6, no. 8, pp. 2280–2301, 2013.

[119] M. E. Bhosale, S. Chae, J. M. Kim, and J.-Y. Choi, "Organic small molecules and polymers as an electrode material for rechargeable lithium ion batteries," *Journal of Materials Chemistry A*, vol. 6, no. 41, pp. 19885–19911, 2018.

[120] C. N. Gannett, L. Melecio-Zambrano, M. J. Theibault, B. M. Peterson, B. P. Fors, and H. D. Abruña, "Organic electrode materials for fast-rate, high-power battery applications," *Materials Reports: Energy*, vol. 1, no. 1, p. 100008, 2021.

[121] O. Allam, B. W. Cho, K. C. Kim, and S. S. Jang, "Application of dft-based machine learning for developing molecular electrode materials in li-ion batteries," *RSC advances*, vol. 8, no. 69, pp. 39414–39420, 2018.

[122] Y. Okamoto and Y. Kubo, "Ab initio calculations of the redox potentials of additives for lithium-ion batteries and their prediction through machine learning," *ACS omega*, vol. 3, no. 7, pp. 7868–7874, 2018.

[123] O. Allam, R. Kuramshin, Z. Stoichev, B. Cho, S. Lee, and S. Jang, "Molecular structure–redox potential relationship for organic electrode materials: density functional theory–machine learning approach," *Materials Today Energy*, vol. 17, p. 100482, 2020.

[124] H. Guo, Q. Wang, A. Stuke, A. Urban, and N. Artrith, "Accelerated atomistic modeling of solid-state battery materials with machine learning," *Frontiers in Energy Research*, vol. 9, p. 265, 2021.

[125] H.-M. Woo, X. Qian, L. Tan, S. Jha, F. J. Alexander, E. R. Dougherty, and B.-J. Yoon, "Optimal decision making in high-throughput virtual screening pipelines," *arXiv preprint arXiv:2109.11683*, 2021.

[126] P. Suryanarayana, "On nearsightedness in metallic systems for o (n) density functional theory calculations: A case study on aluminum," *Chemical Physics Letters*, vol. 679, pp. 146–151, 2017.

[127] J. Leszczynski, *Handbook of computational chemistry*. Springer Science & Business Media, 2012.

[128] H. Lyu, X.-G. Sun, and S. Dai, "Organic cathode materials for lithium-ion batteries: Past, present, and future," *Advanced Energy and Sustainability Research*, vol. 2, no. 1, p. 2000044, 2021.

[129] D. Weininger, "Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules," *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[130] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik, "Selfies: a robust representation of semantically constrained graphs with an example application in chemistry," *arXiv preprint arXiv:1905.13741*, 2019.

[131] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*, pp. 267–285, Springer, 1982.

[132] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[133] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[134] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[135] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*, pp. 1263–1272, PMLR, 2017.

[136] Q. Chen, H.-M. Woo, X. Chen, and B.-J. Yoon, "Neural message passing for objective-based uncertainty quantification and optimal experimental design," *arXiv preprint arXiv:2203.07120*, 2022.

[137] A. D. Bochevarov, E. Harder, T. F. Hughes, J. R. Greenwood, D. A. Braden, D. M. Philipp, D. Rinaldo, M. D. Halls, J. Zhang, and R. A. Friesner, "Jaguar: A high-performance quantum chemistry software program with strengths in life and materials sciences," *International Journal of Quantum Chemistry*, vol. 113, no. 18, pp. 2110–2142, 2013.

[138] J. Paier, R. Hirschl, M. Marsman, and G. Kresse, "The perdew–burke–ernzerhof exchange-correlation functional applied to the g2-1 test set using a plane-wave basis set," *The Journal of chemical physics*, vol. 122, no. 23, p. 234102, 2005.

[139] R. Dithcfield, W. Hehre, and J. Pople, "Self-consistent molecular-orbital methods. 9. extended gaussian-type basis for molecular-orbital studies of organic molecules," *Journal of Chemical Physics*, vol. 54, no. 2, pp. 724–728, 1971.

[140] P. Winget, C. J. Cramer, and D. G. Truhlar, "Computation of equilibrium oxidation and reduction potentials for reversible and dissociative electron-transfer reactions in solution," *Theoretical Chemistry Accounts*, vol. 112, no. 4, pp. 217–227, 2004.

[141] P. Winget, E. J. Weber, C. J. Cramer, and D. G. Truhlar, "Computational electrochemistry: aqueous one-electron oxidation potentials for substituted anilines," *Physical Chemistry Chemical Physics*, vol. 2, no. 6, pp. 1231–1239, 2000.

[142] S. P. Ong, V. L. Chevrier, G. Hautier, A. Jain, C. Moore, S. Kim, X. Ma, and G. Ceder, "Voltage, stability and diffusion barrier differences between sodium-ion and lithium-ion intercalation materials," *Energy & Environmental Science*, vol. 4, no. 9, pp. 3680–3688, 2011.

APPENDIX A

PROOF OF THEOREM 1

Proof. Without loss of generality, we assume that $\omega_1 \geq \omega_2$ and $0 \leq \Theta(0) < 2\pi$, where $\Theta(t) = \theta_1(t) - \theta_2(t)$. Suppose that $\frac{|\omega_1 - \omega_2|}{2} \leq a$. Then,

$$\Theta'(t) = \omega_1 - \omega_2 - 2a \sin(\Theta(t)) \tag{A.1}$$

$$\triangleq F(\Theta(t)). \tag{A.2}$$

As $0 \leq \omega_1 - \omega_2 \leq 2a$, there always exists a value $\Theta^* \in \left[0, \frac{\pi}{2}\right]$ satisfying $\sin(\Theta^*) = \frac{\omega_1 - \omega_2}{2a}$. Note that Θ^* is a unique stable critical point as $F'(\Theta^*) = -2a\cos(\Theta^*) < 0$. Thus, $\Theta(t) \to \Theta^*$ as $t \to \infty$, resulting in $\Theta'(t) \to 0$, unless $\Theta(0) = \pi - \Theta^*$ which is an unstable critical point. If $\frac{|\omega_1 - \omega_2|}{2} > a$,

$$\Theta'(t) \geq |\omega_1 - \omega_2| - 2a\,|\sin(\theta_1(t) - \theta_2(t))| \tag{A.3}$$

$$\geq |\omega_1 - \omega_2| - 2a \tag{A.4}$$

$$\geq 0. \tag{A.5}$$

□

APPENDIX B

ASYMPTOTIC CLASSIFICATION ACCURACY

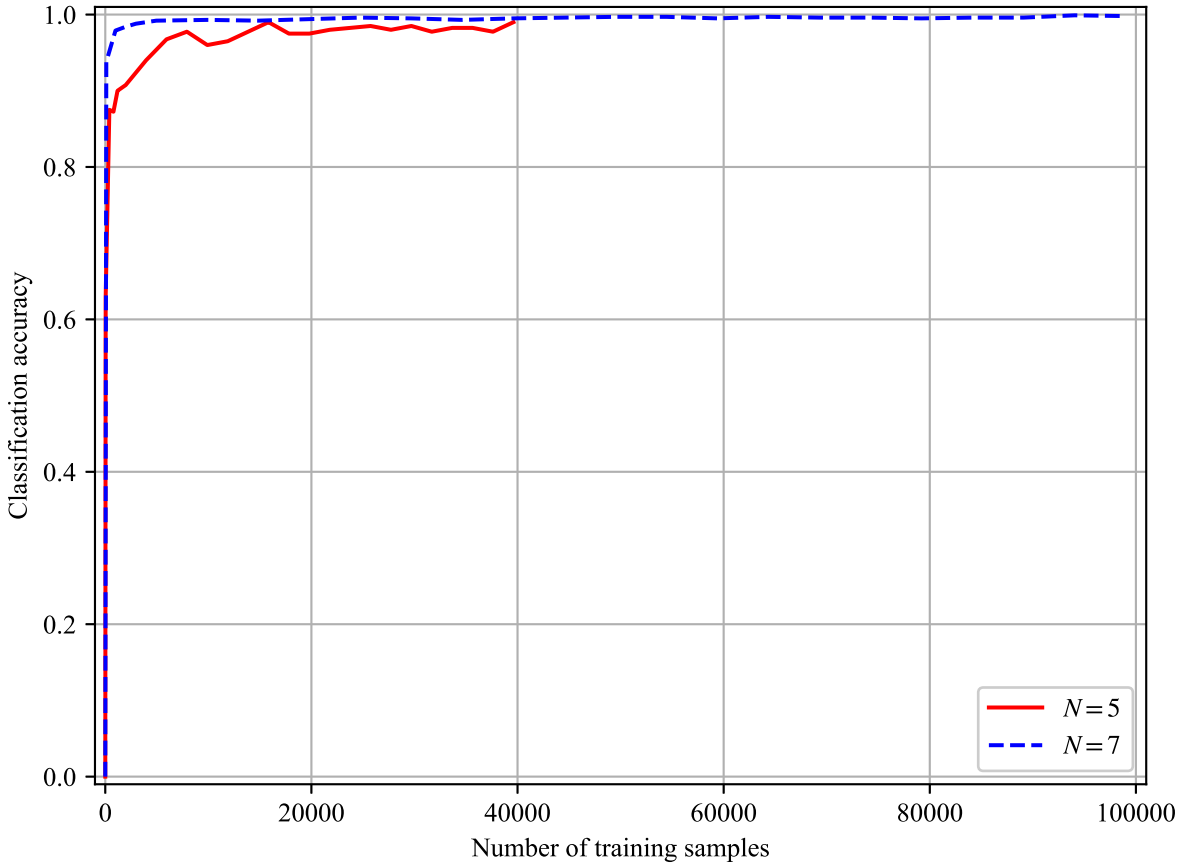


Figure B.1: Asymptotic classification accuracy of the neural network model trained with different amounts of training data. The accuracy sharply increased as the number of training samples increased. Note that for $N = 5$, we generated the training samples while randomly changing the natural frequencies of the oscillators, resulting in small fluctuations in the classification accuracy.

APPENDIX C

EXPERIMENTAL DESIGN PERFORMANCE

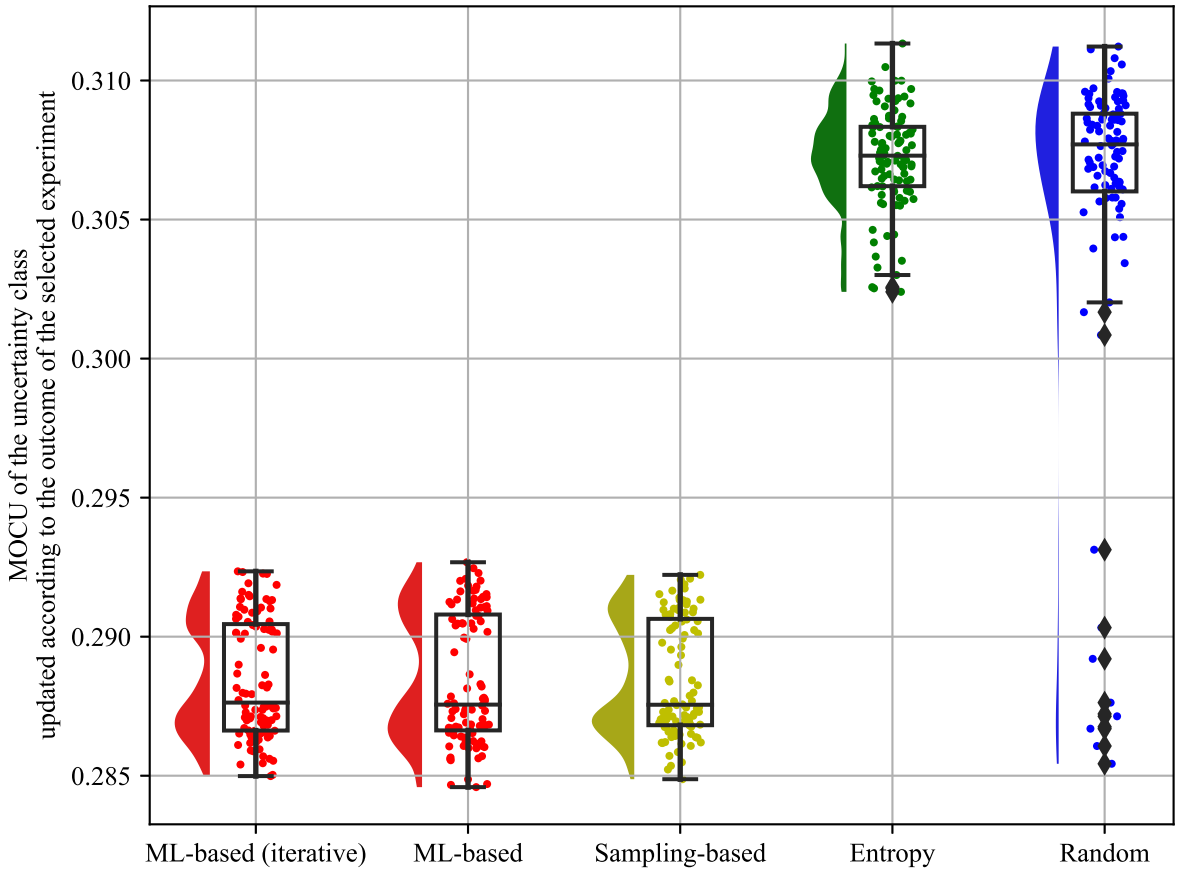


Figure C.1: Remaining uncertainty, measured by mean objective cost of uncertainty (MOCU), after performing the first experiment selected by each algorithm for the uncertain Kuramoto model with five oscillators. The results are shown based on one hundred reevaluations with different true models. As can be seen, the three MOCU-based experimental design approaches yielded the best overall performance in terms of reducing model uncertainty.

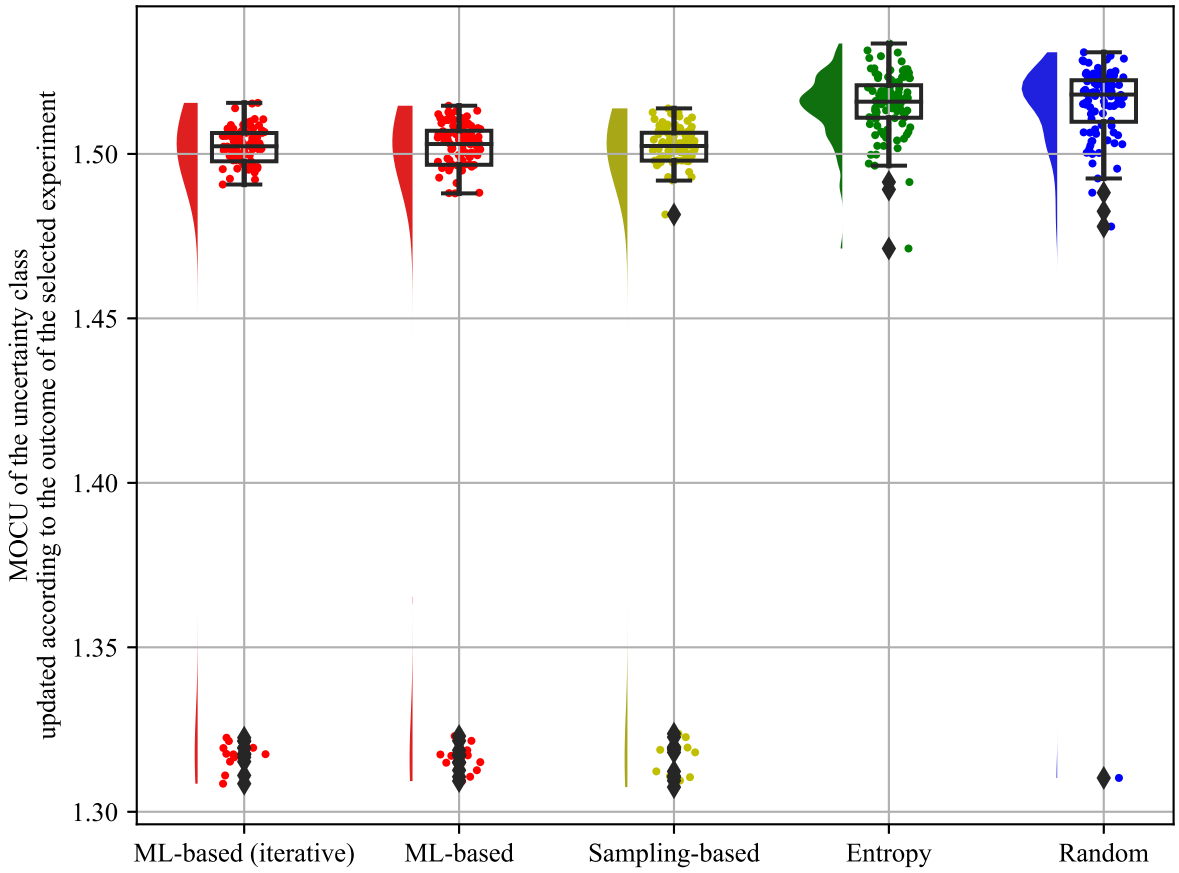


Figure C.2: Remaining uncertainty, measured by MOCU, after performing the first experiment selected by each algorithm for the uncertain Kuramoto model with seven oscillators. The results are shown based on one hundred reevaluations with different true models. This example showed that the uncertainty reduction performance of the first experiment depends on the underlying true (unknown) model, which we do not have any control over. This was not surprising, since the MOCU-based optimal experimental design (OED) aimed to predict the best experiment based on its expected performance for *all* possible models in the uncertainty class. But the efficacy of the selected experiment naturally varied across different true models. Nevertheless, the results in this figure showed that the proposed machine learning-based (ML-based) scheme faithfully replicates the sampling-based approach, which was the primary goal of this study.

APPENDIX D

ANALYTIC PERFORMANCE OF OPTIMIZED HTVS PIPELINES

$$p\left(y_1, y_2, y_3, y_4\right) \sim \mathcal{N}\left(\boldsymbol{\mu}=\mathbf{0}, \boldsymbol{\Sigma}=\begin{bmatrix} 1 & 0.2 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 & 0.2 \\ 0.2 & 0.2 & 1 & 0.2 \\ 0.2 & 0.2 & 0.2 & 1 \end{bmatrix}\right)$$

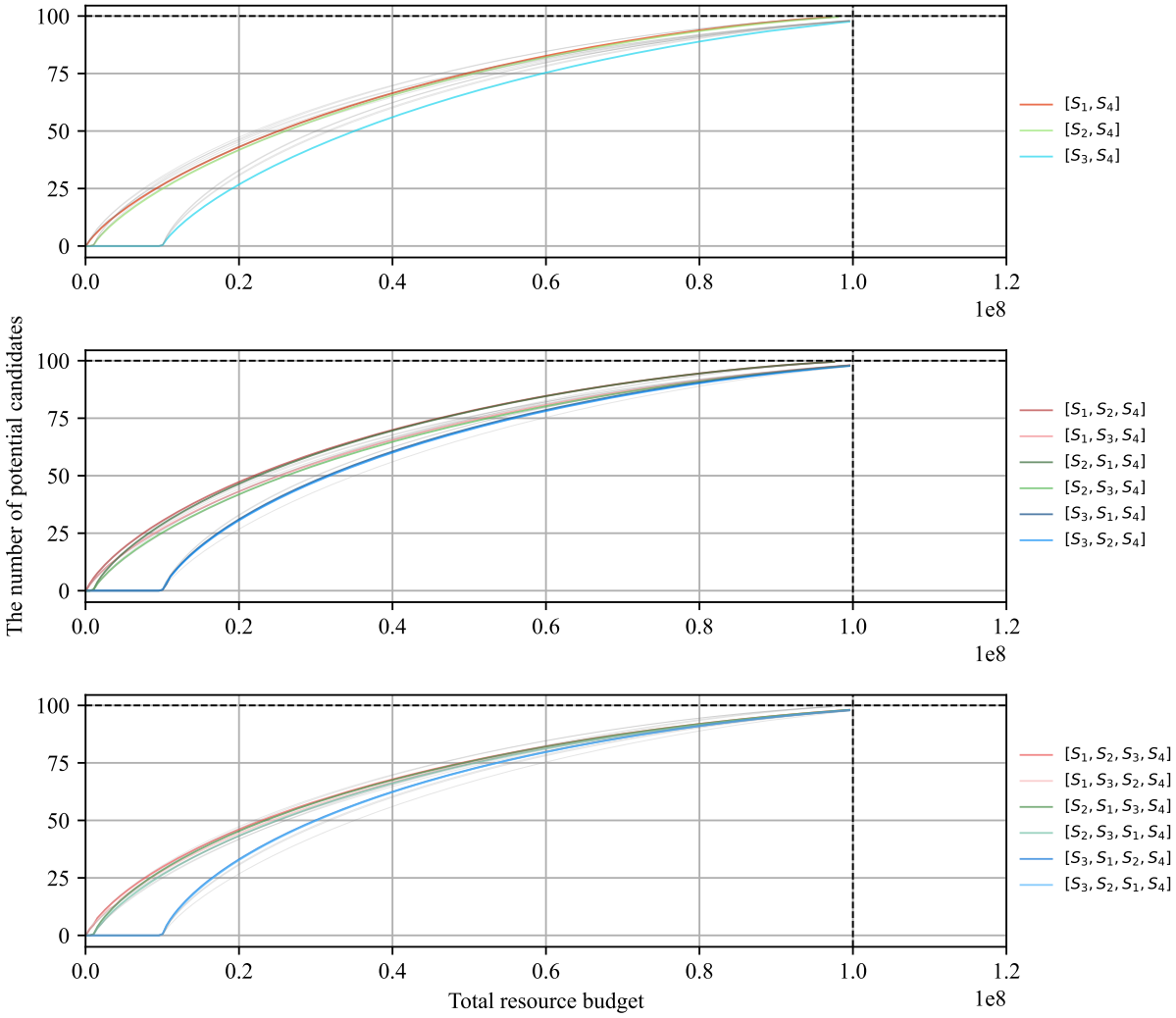


Figure D.1: Performance comparison of the optimized HTVS pipelines in terms of discovery capability in scenario 1.

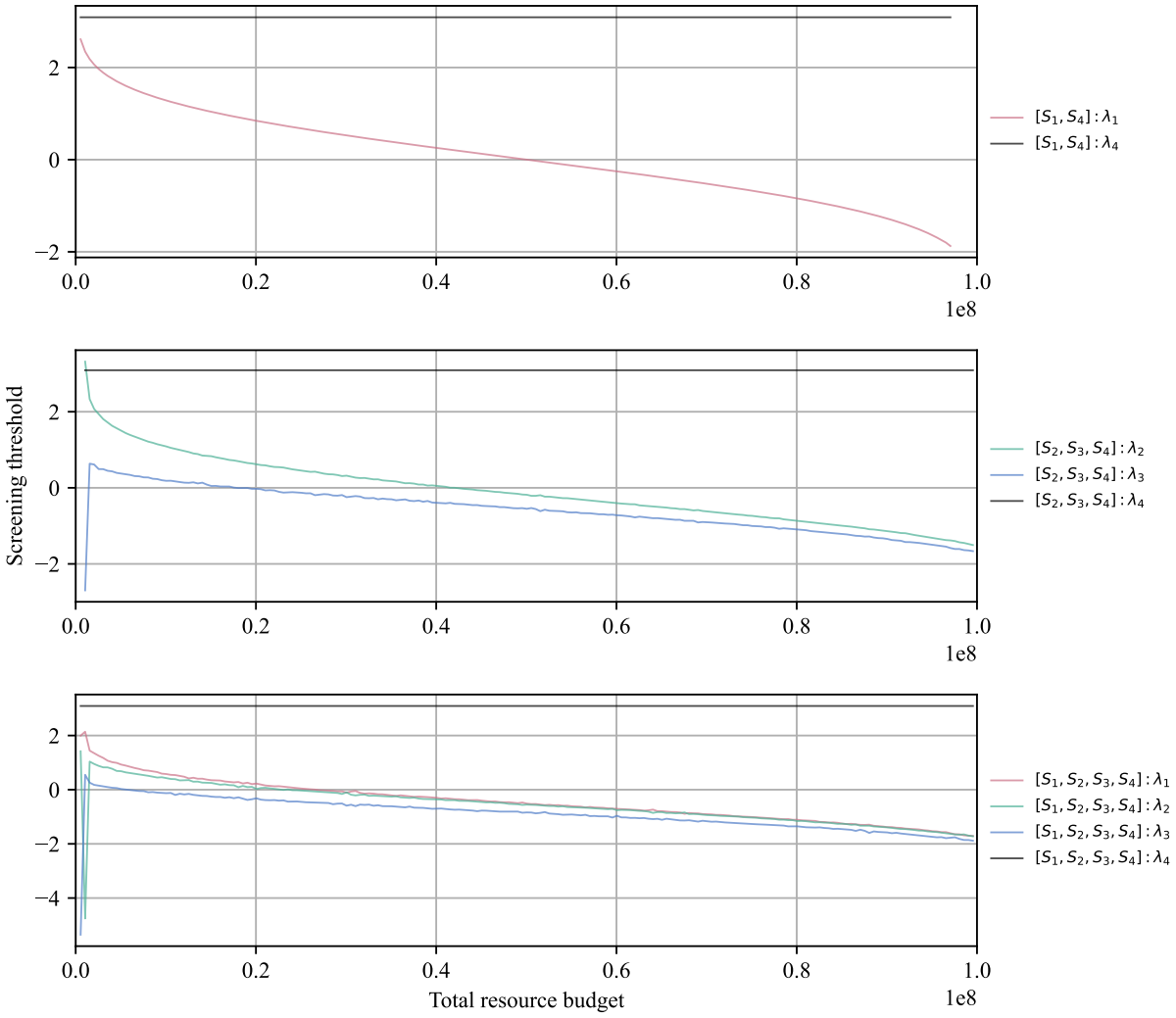


Figure D.2: Screening thresholds of the optimized pipelines in scenario 1.

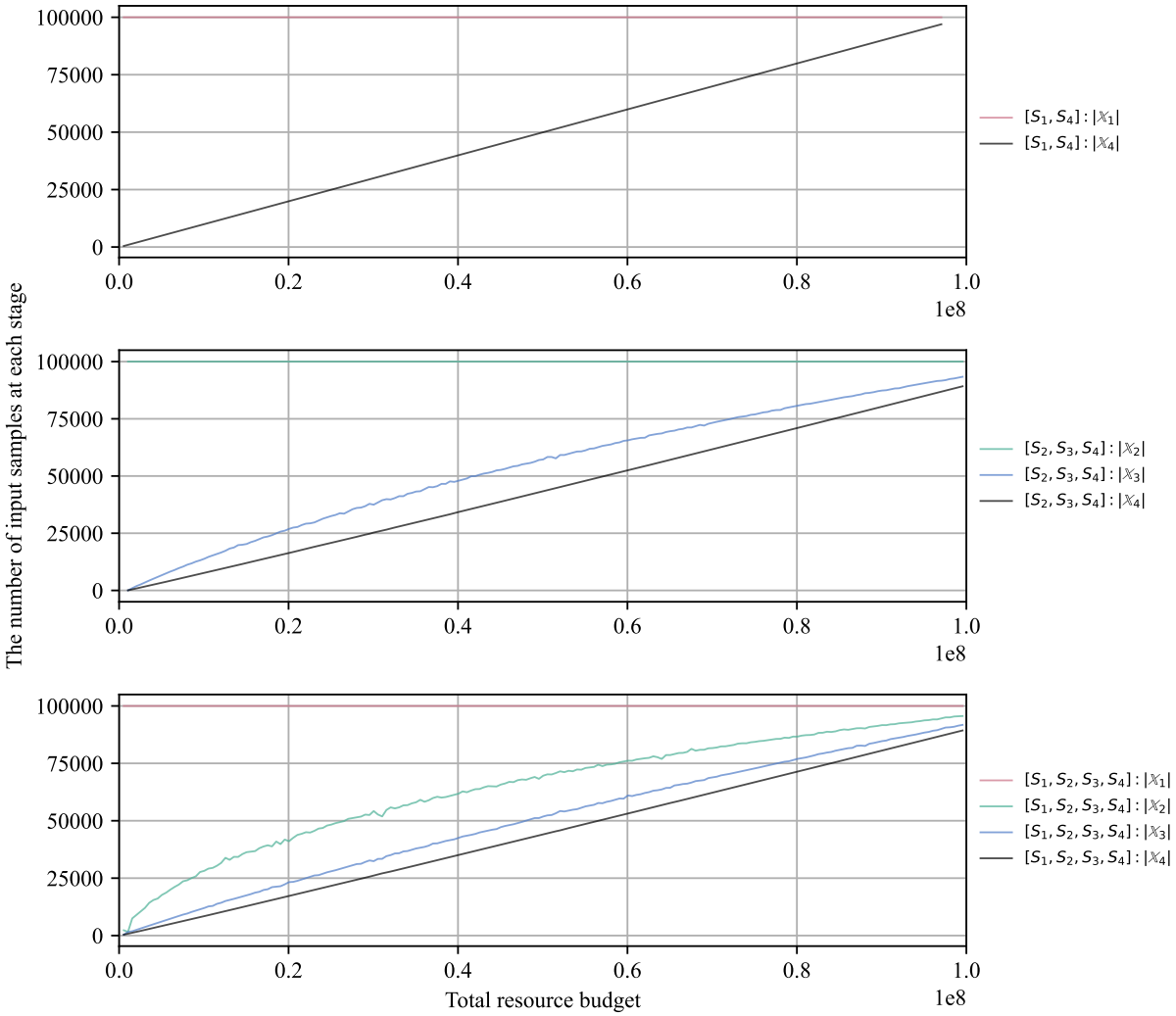


Figure D.3: The number of input samples at each stage in scenario 1.

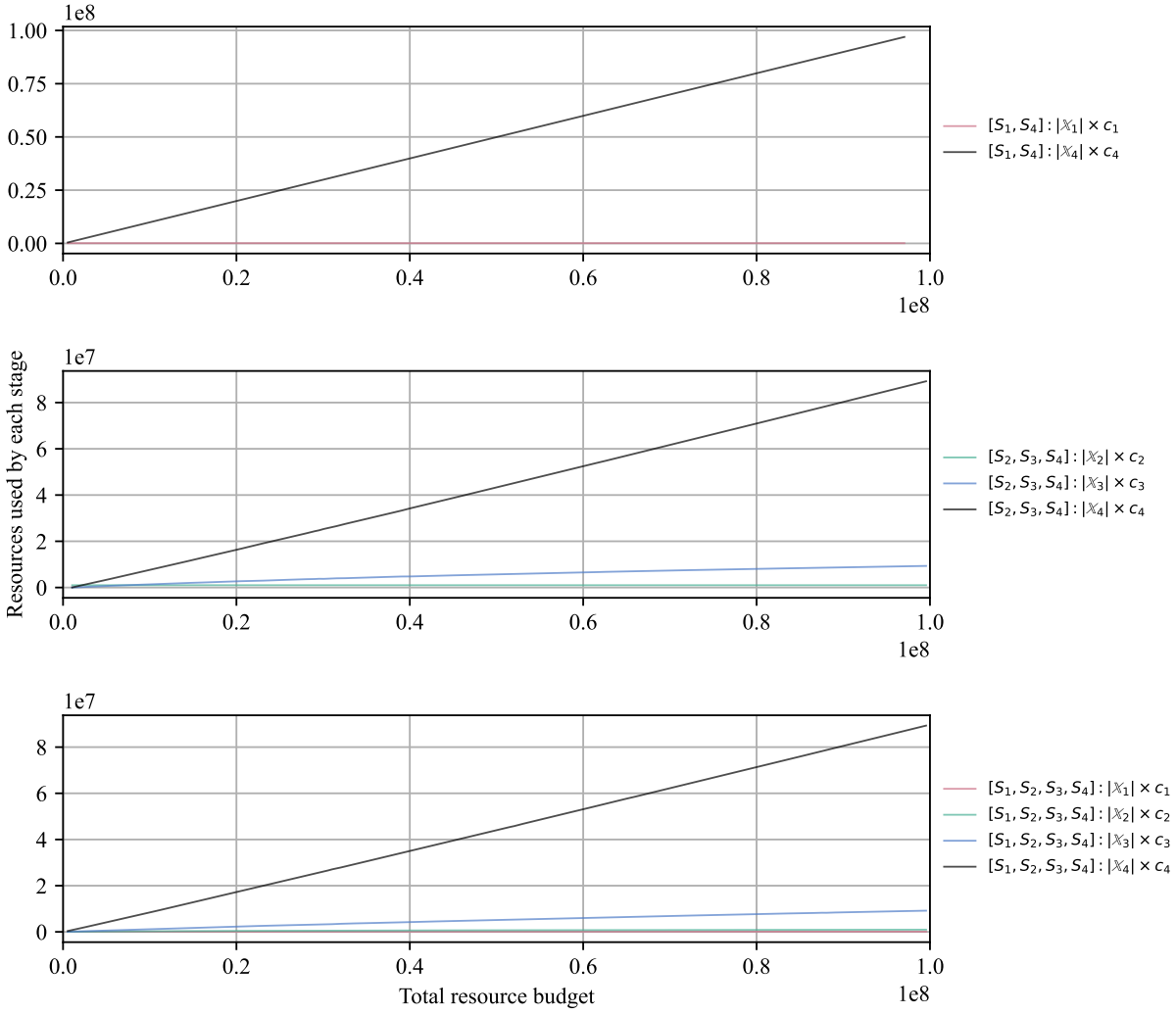


Figure D.4: Resources used by each stage in scenario 1.

$$p\left(y_1, y_2, y_3, y_4\right) \sim \mathcal{N}\left(\boldsymbol{\mu}=\mathbf{0}, \boldsymbol{\Sigma}=\begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1 \end{bmatrix}\right)$$

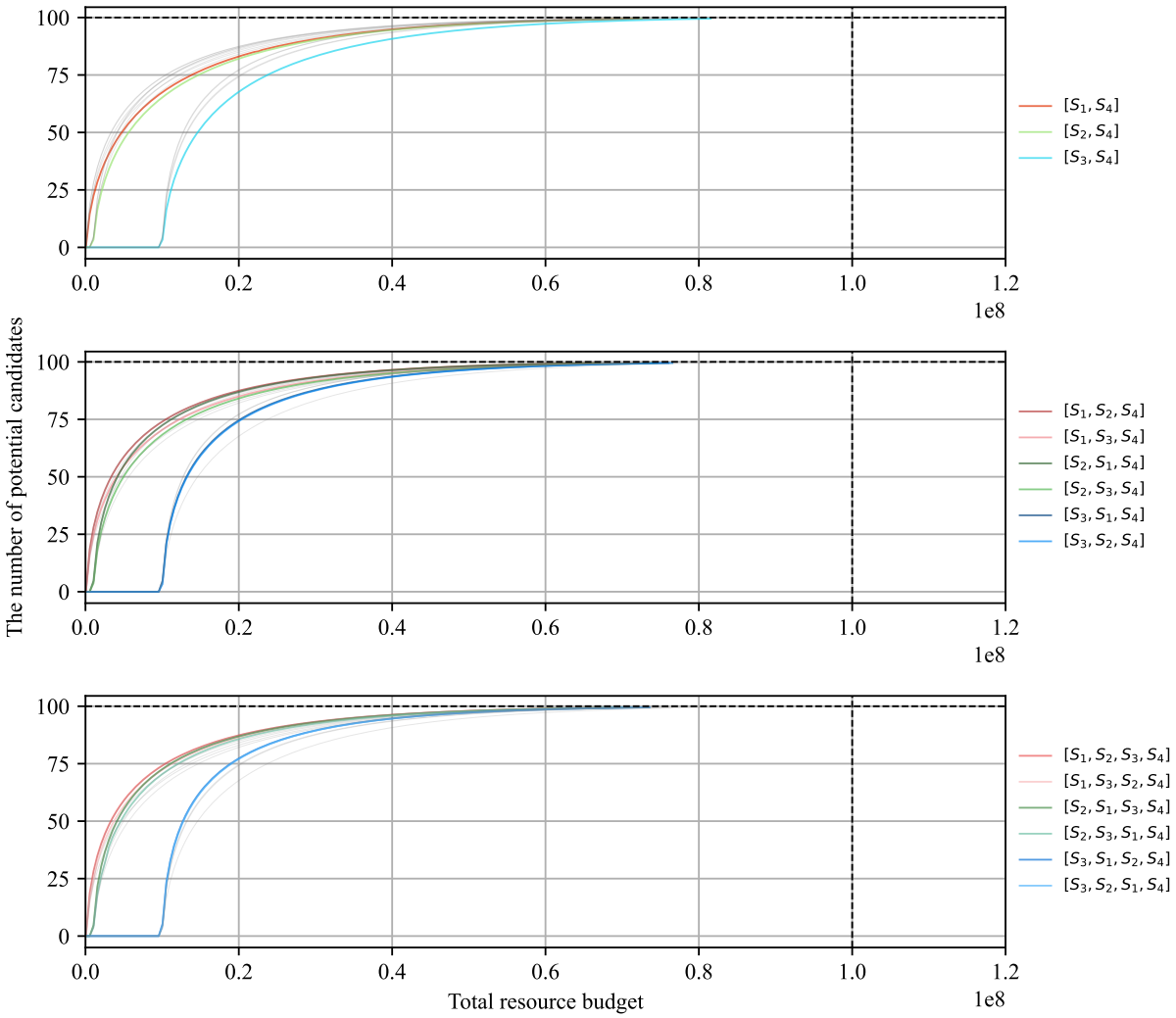


Figure D.5: Performance comparison of the optimized HTVS pipelines in terms of discovery capability in scenario 2.

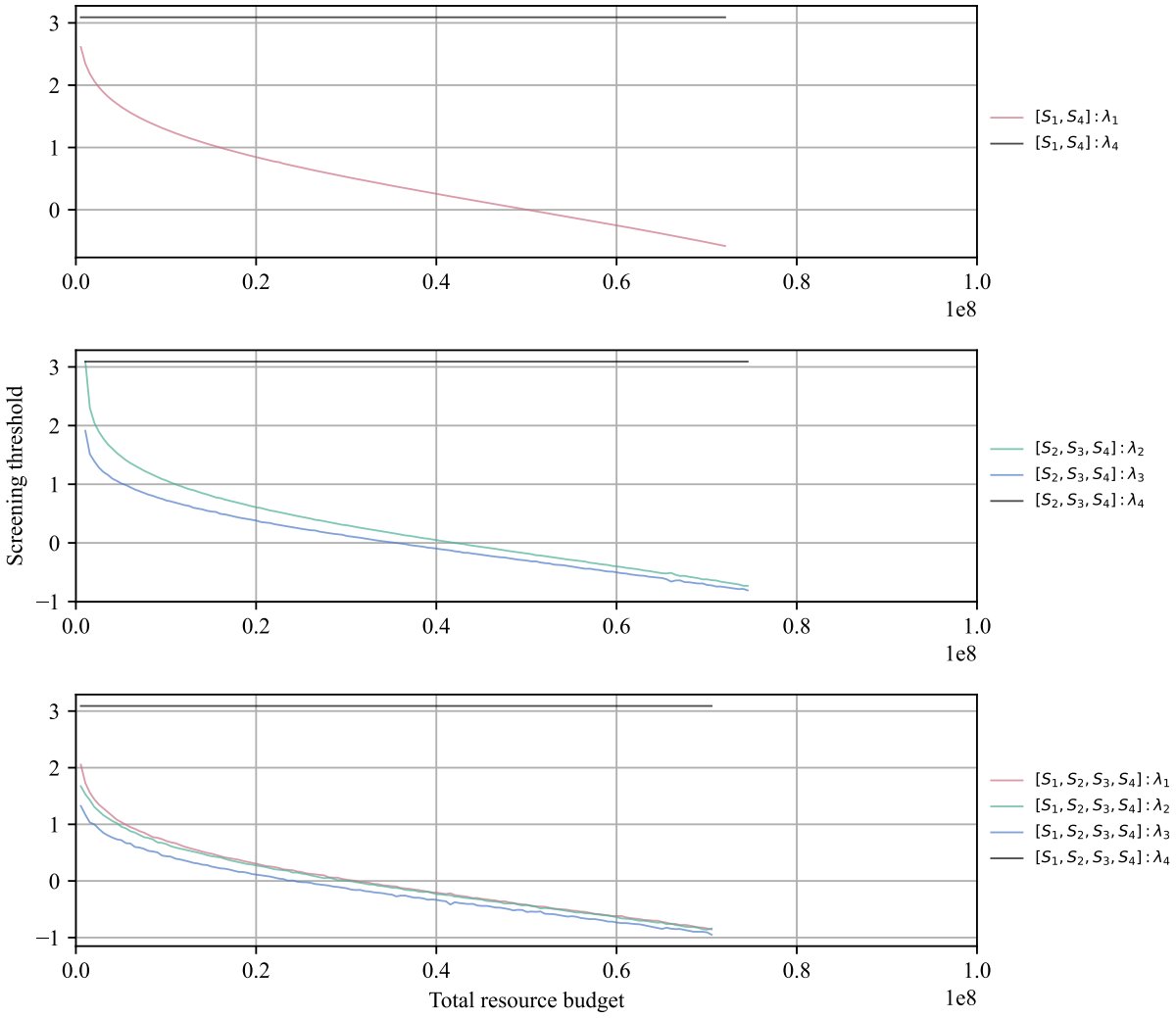


Figure D.6: Screening thresholds of the optimized pipelines in scenario 2.

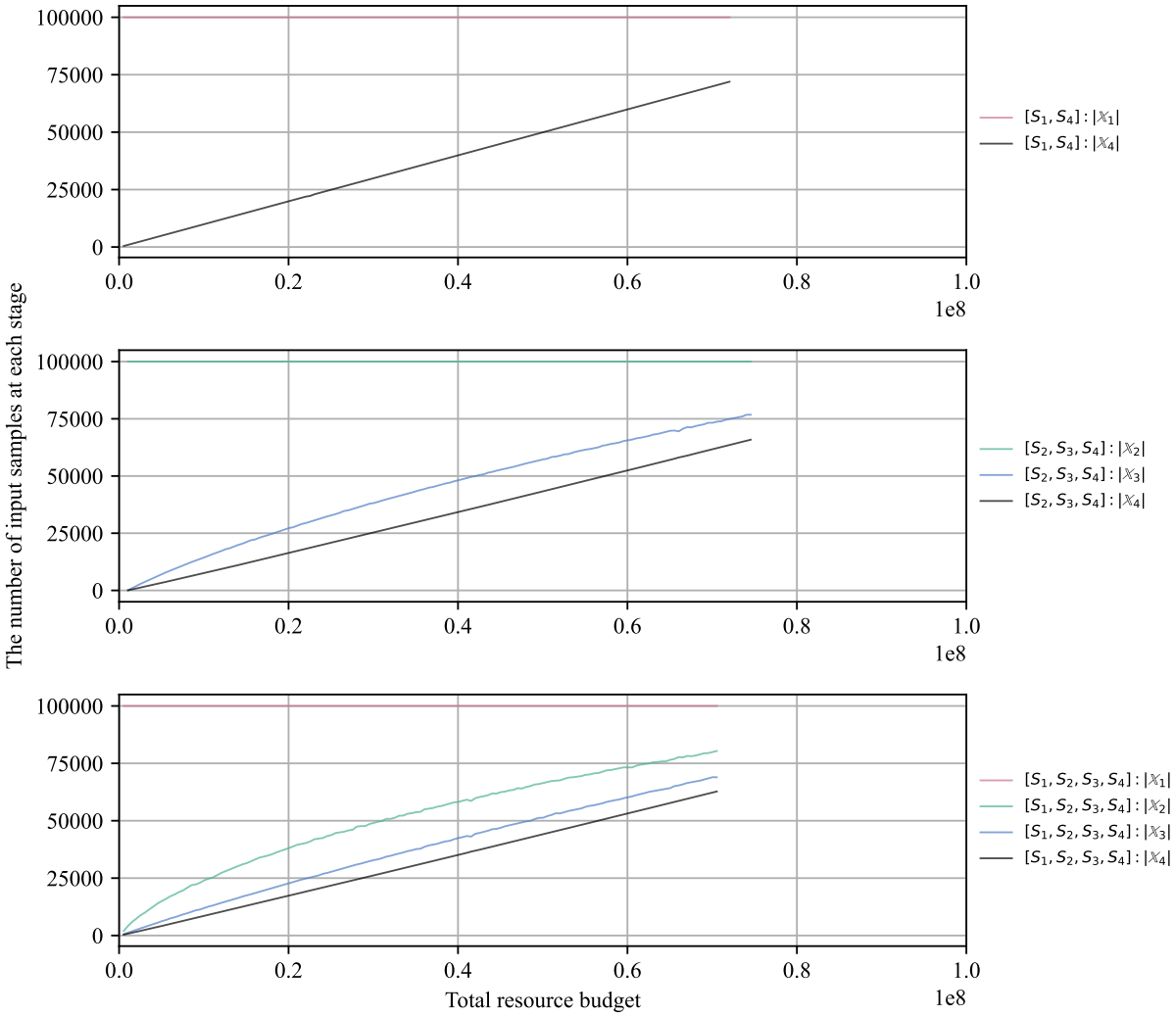


Figure D.7: The number of input samples at each stage in scenario 2.

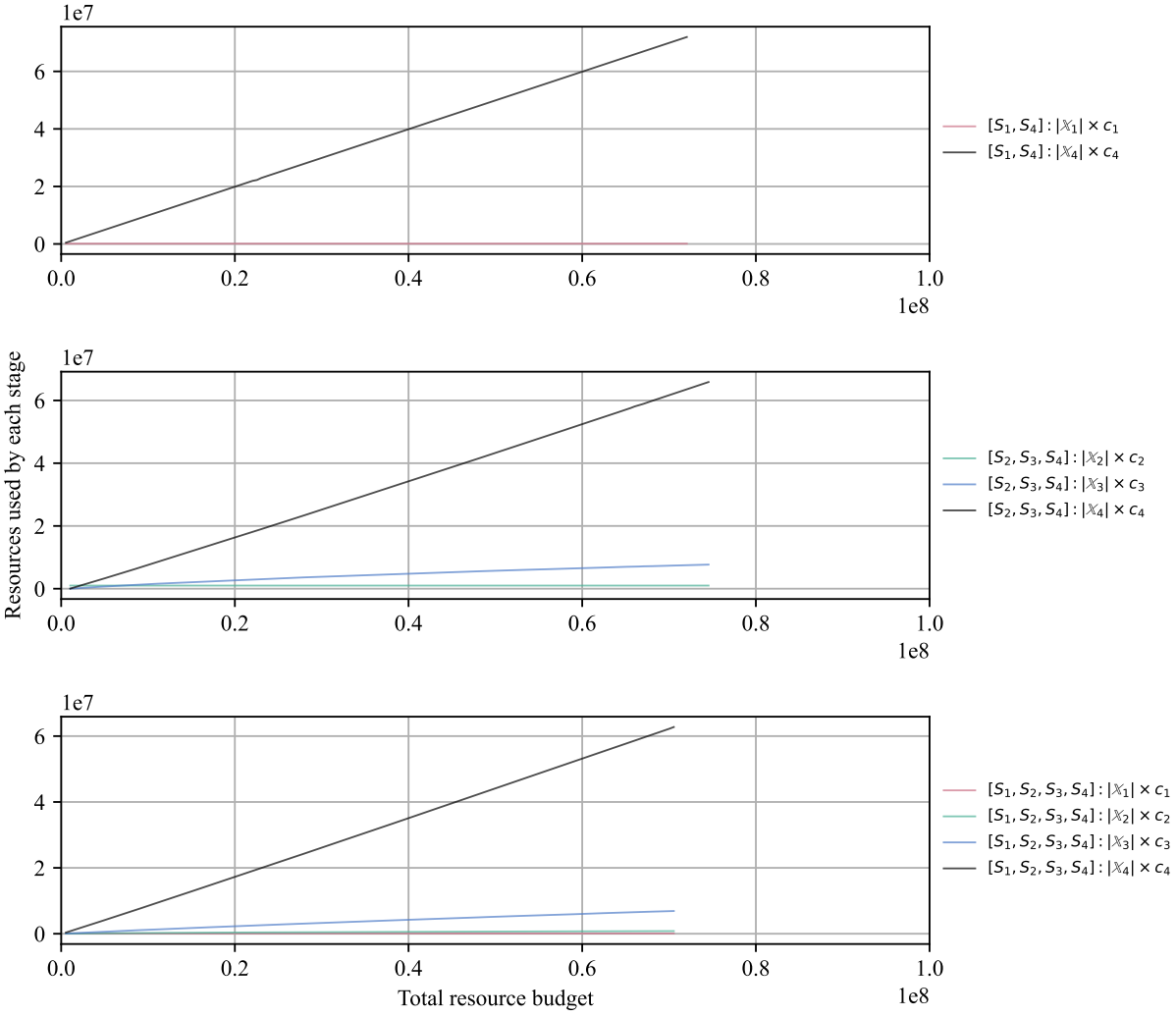


Figure D.8: Resources used by each stage in scenario 2.

$$p\left(y_1, y_2, y_3, y_4\right) \sim \mathcal{N}\left(\boldsymbol{\mu}=\mathbf{0}, \boldsymbol{\Sigma}=\begin{bmatrix} 1 & 0.8 & 0.8 & 0.8 \\ 0.8 & 1 & 0.8 & 0.8 \\ 0.8 & 0.8 & 1 & 0.8 \\ 0.8 & 0.8 & 0.8 & 1 \end{bmatrix}\right)$$

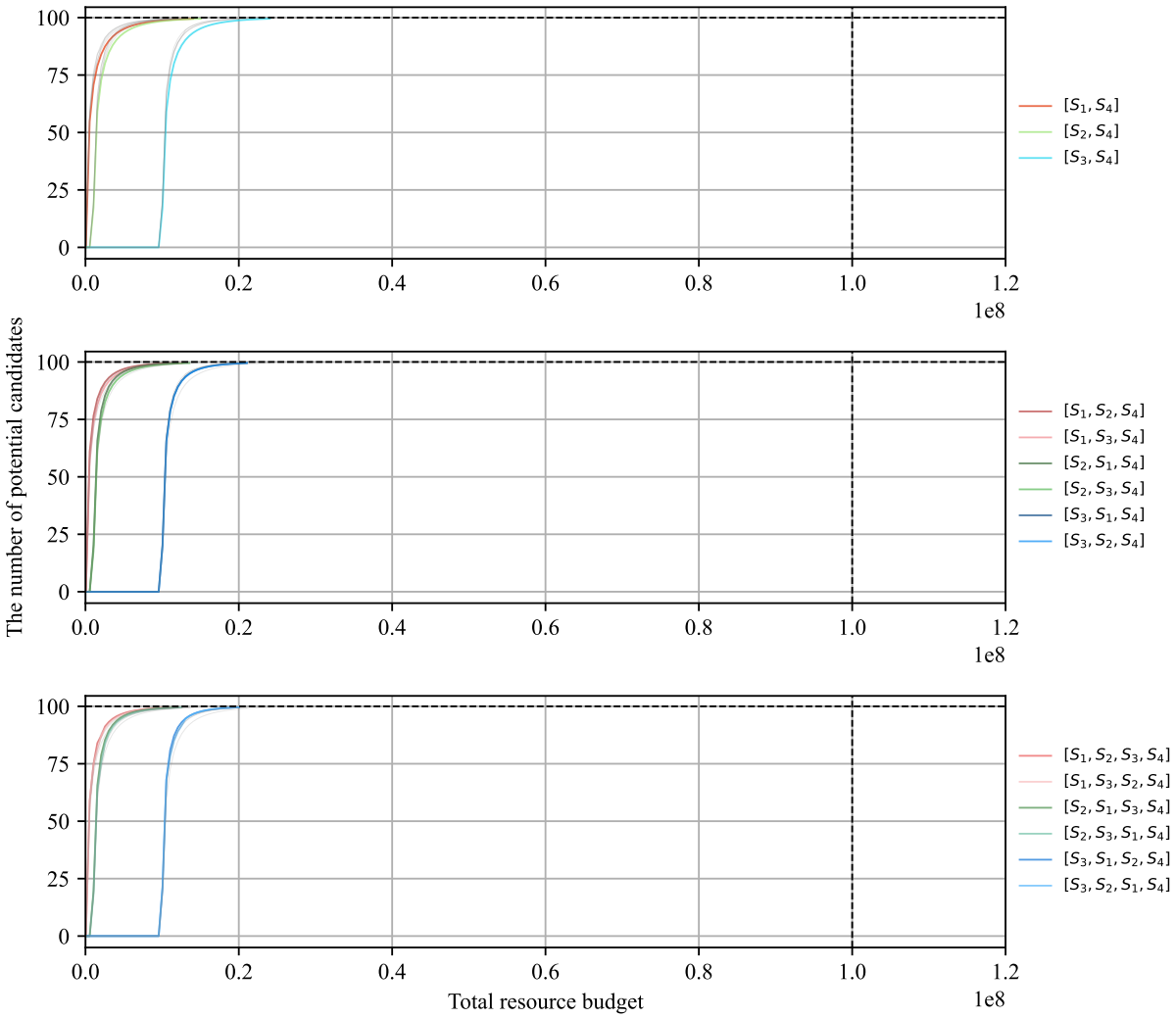


Figure D.9: Performance comparison of the optimized HTVS pipelines in terms of discovery capability in scenario 3.

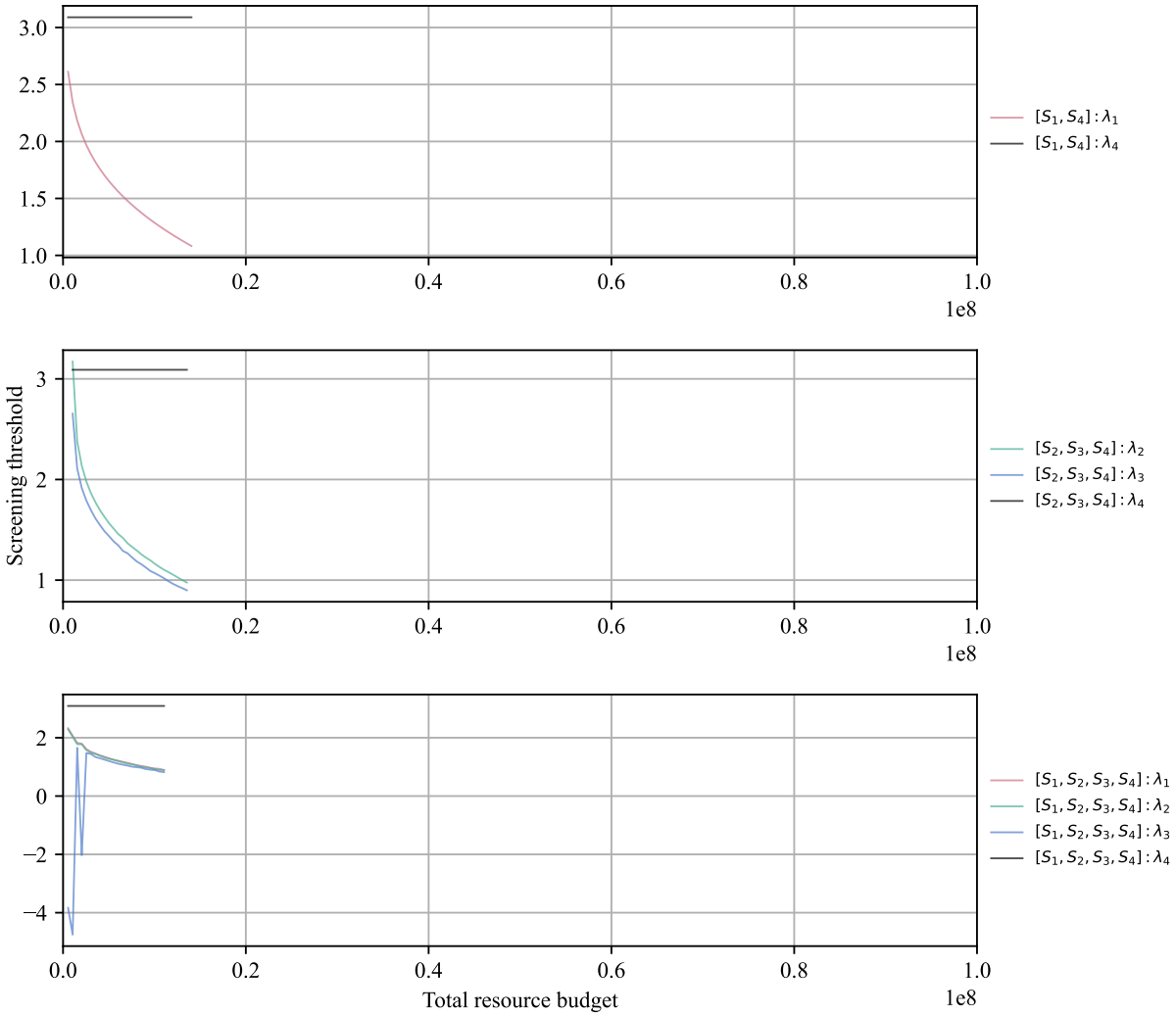


Figure D.10: Screening thresholds of the optimized pipelines in scenario 3.

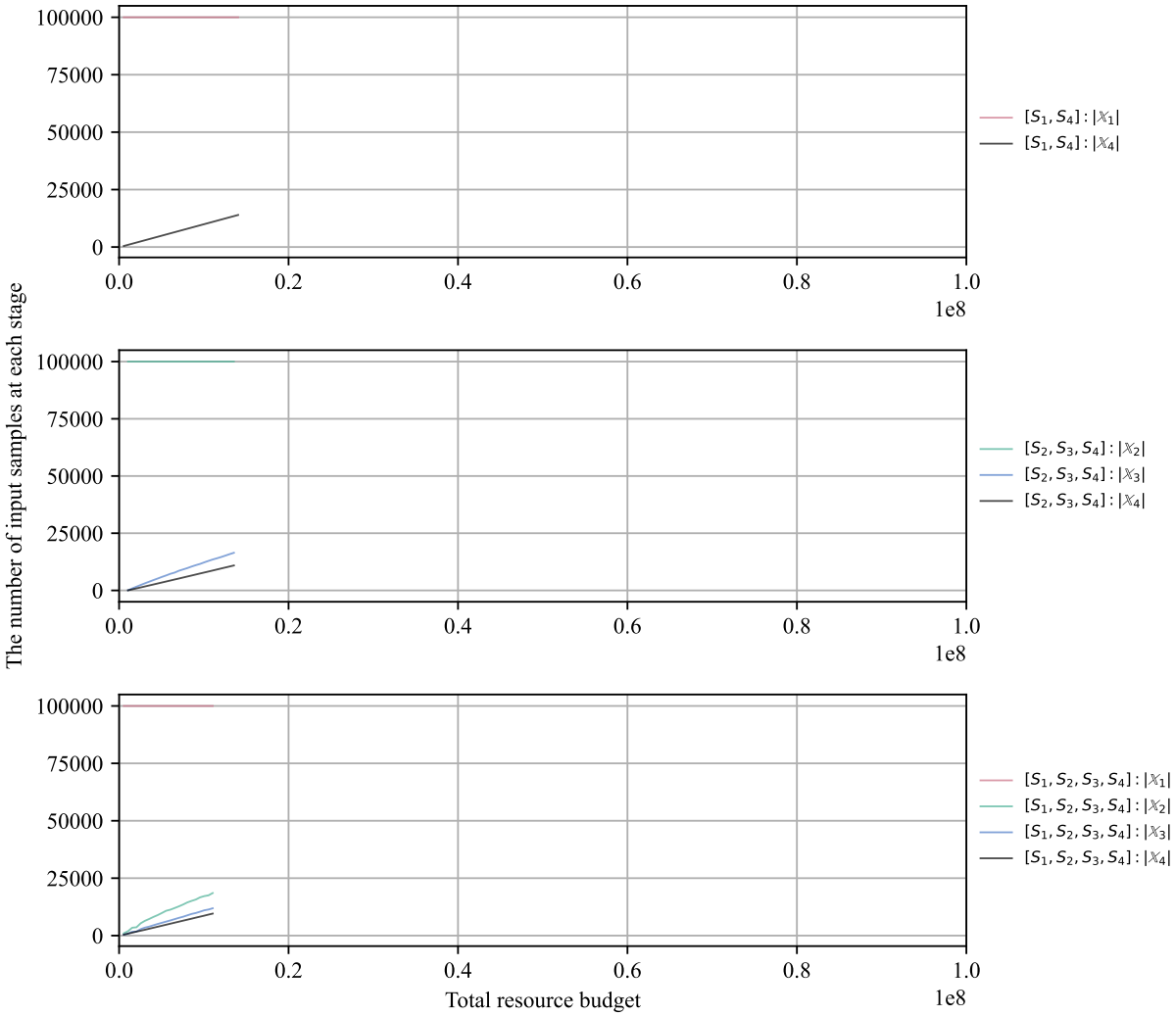


Figure D.11: The number of input samples at each stage in scenario 3.

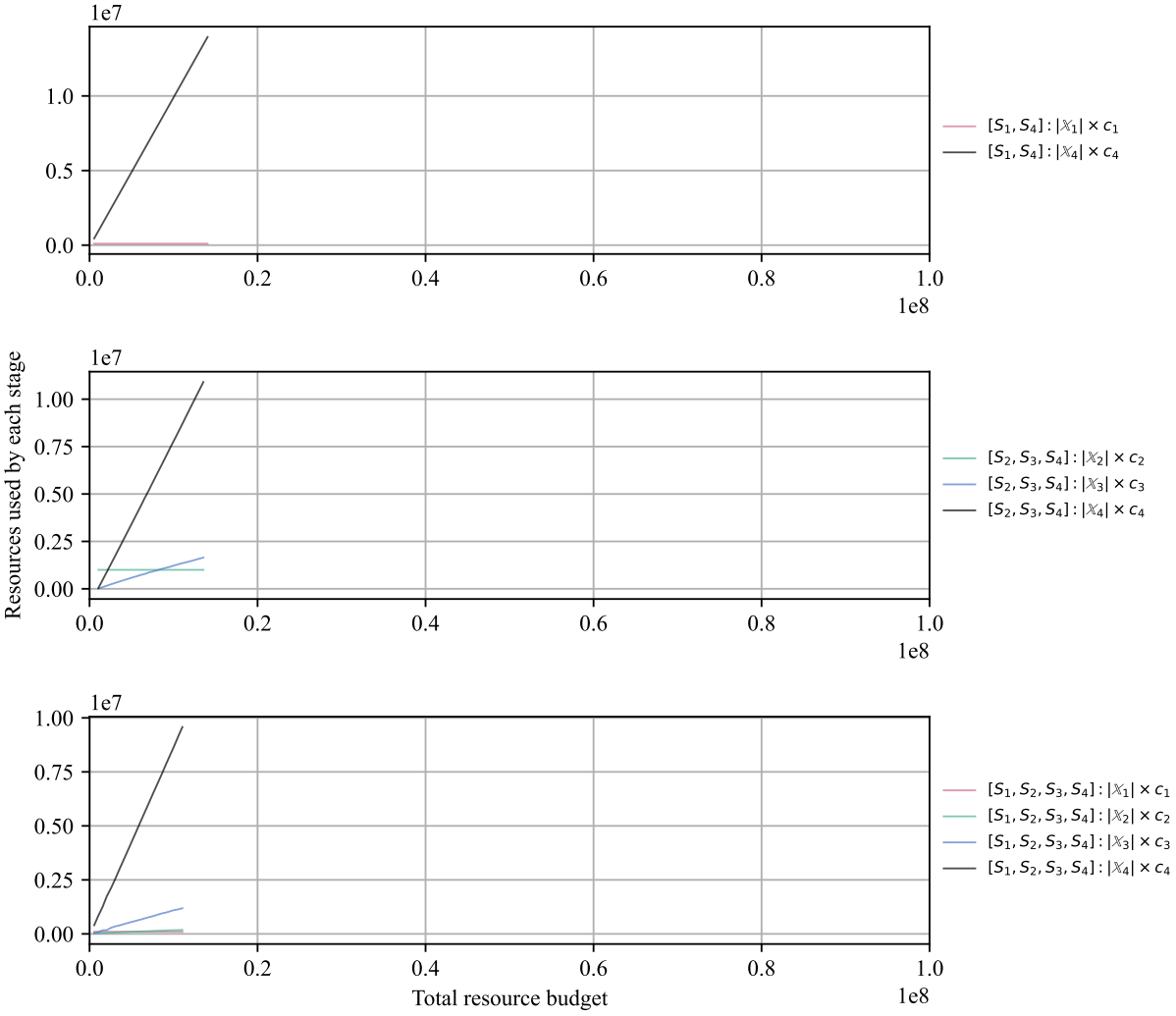


Figure D.12: Resources used by each stage in scenario 3.

$$p\left(y_1, y_2, y_3, y_4\right) \sim \mathcal{N}\left(\boldsymbol{\mu}=\mathbf{0}, \boldsymbol{\Sigma}=\begin{bmatrix} 1 & 0.2 & 0.2 & 0.8 \\ 0.2 & 1 & 0.2 & 0.2 \\ 0.2 & 0.2 & 1 & 0.2 \\ 0.8 & 0.2 & 0.2 & 1 \end{bmatrix}\right)$$

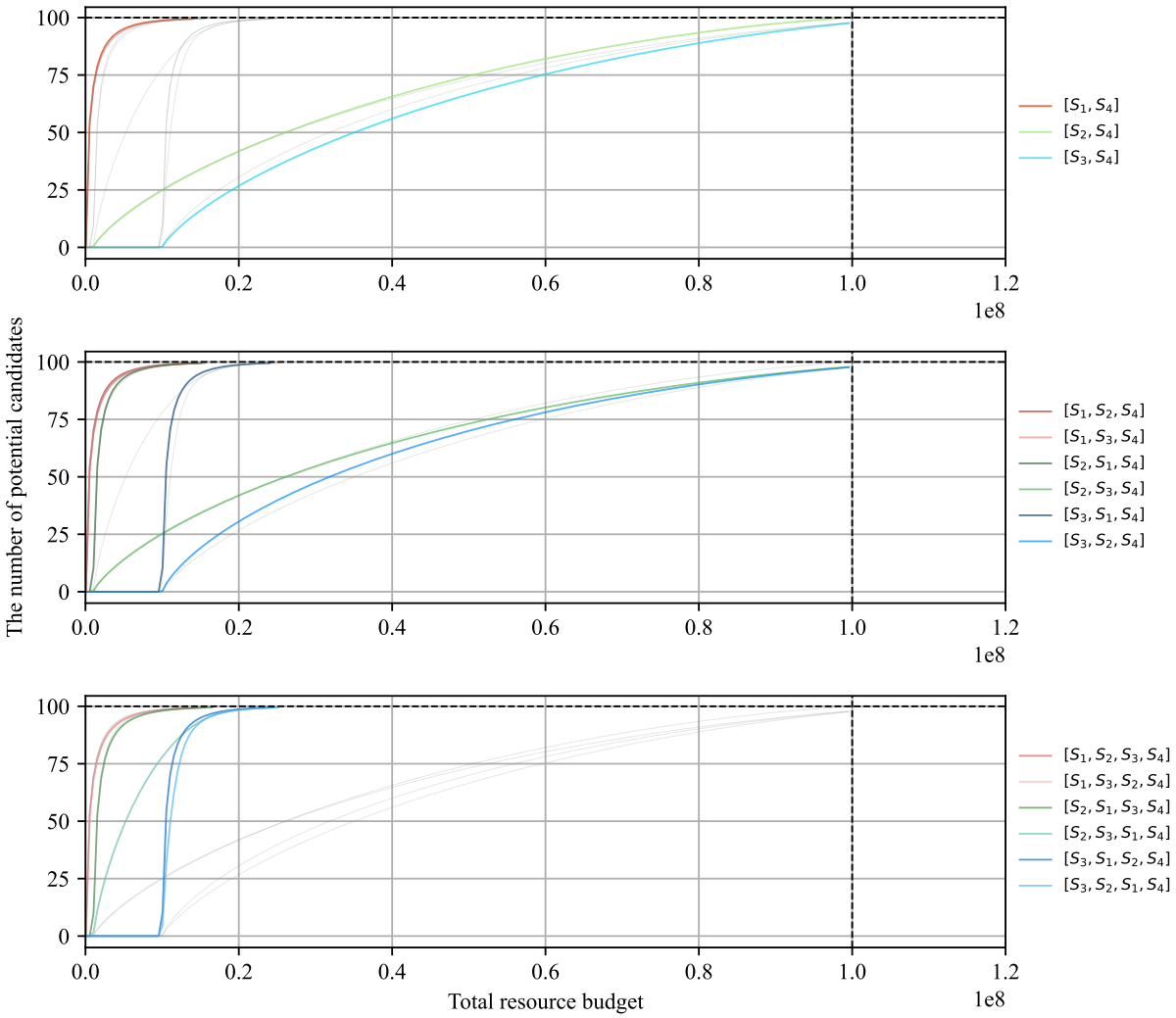


Figure D.13: Performance comparison of the optimized HTVS pipelines in terms of discovery capability in scenario 4.

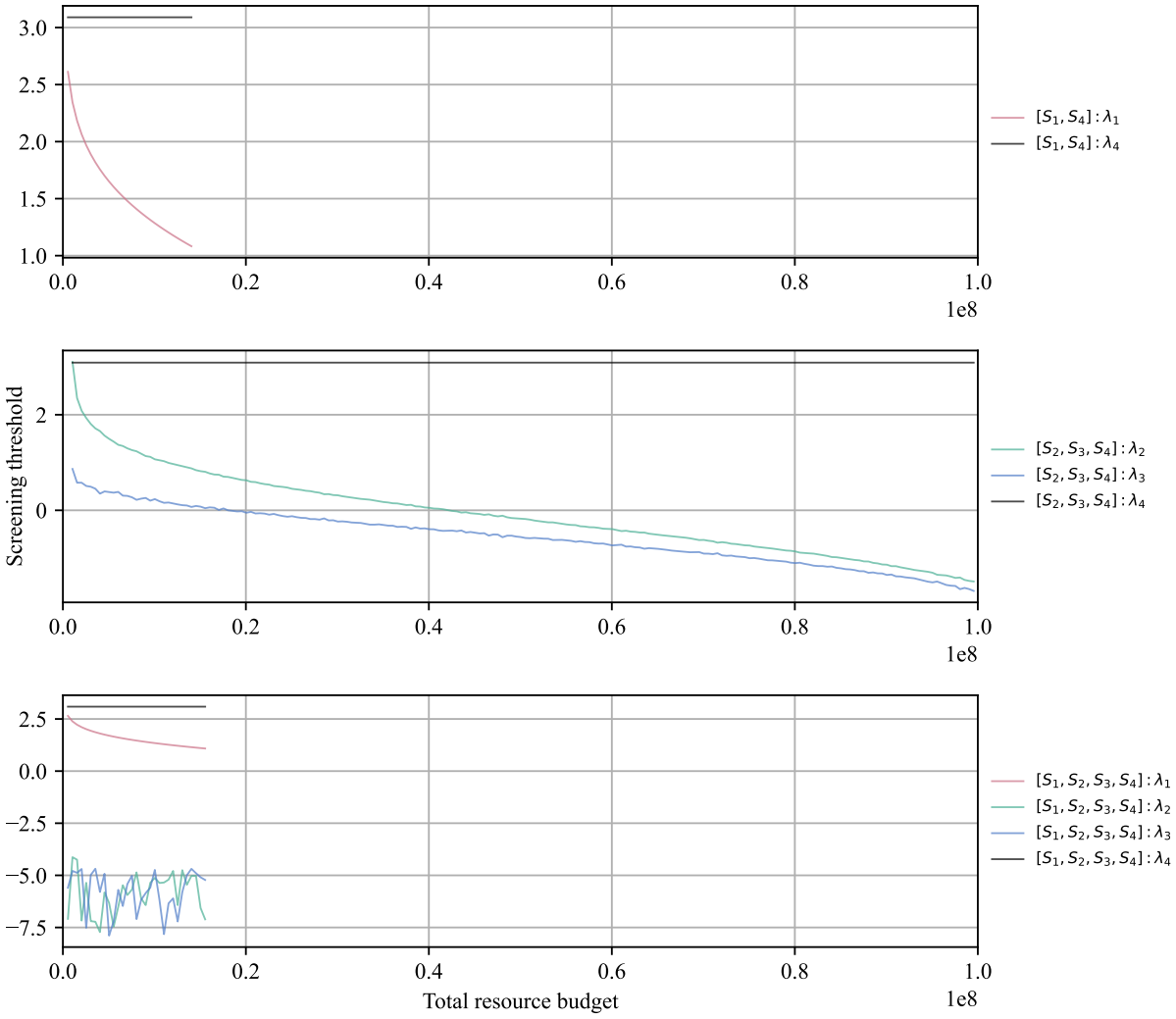


Figure D.14: Screening thresholds of the optimized pipelines in scenario 4.

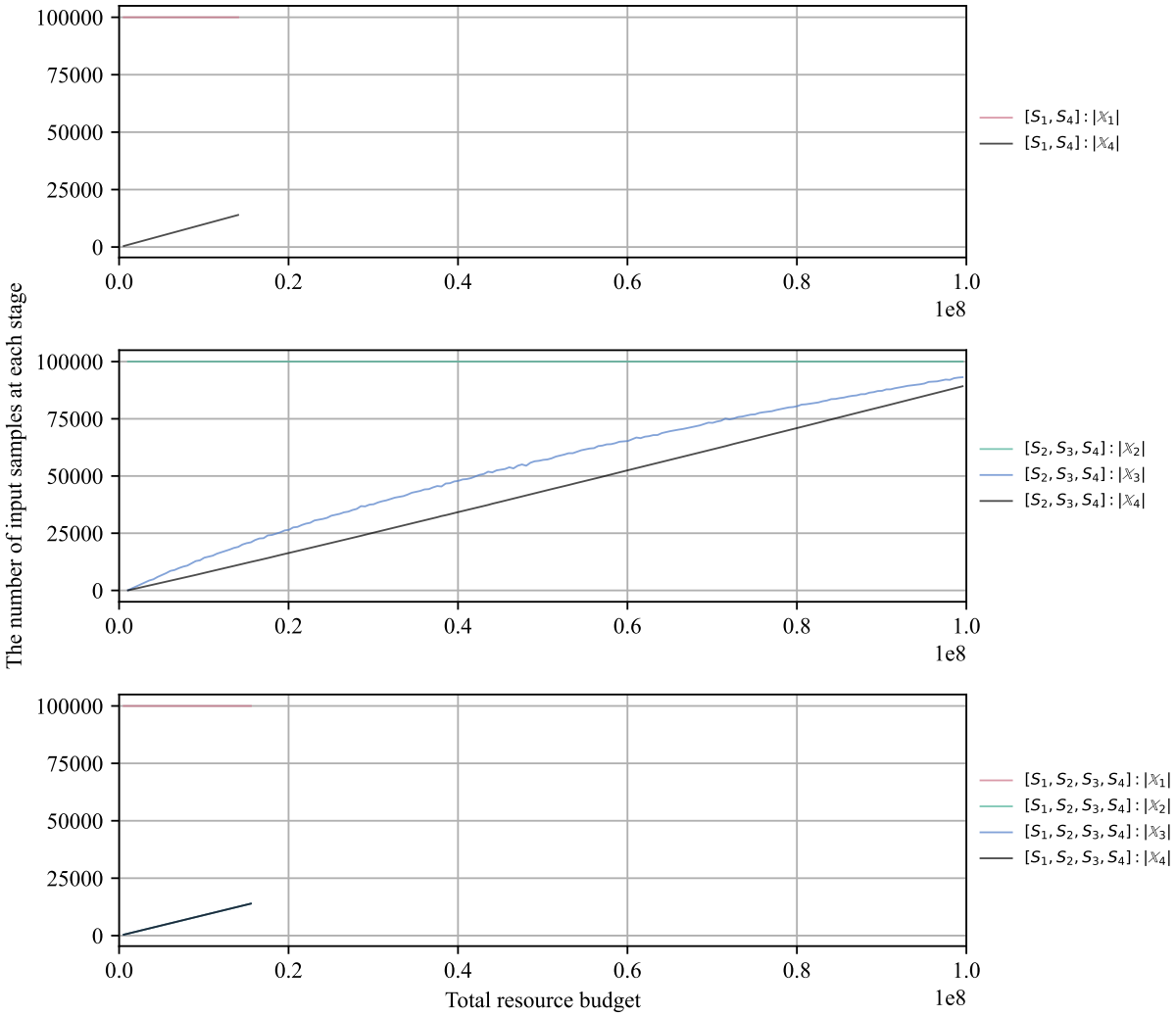


Figure D.15: The number of input samples at each stage in scenario 4.

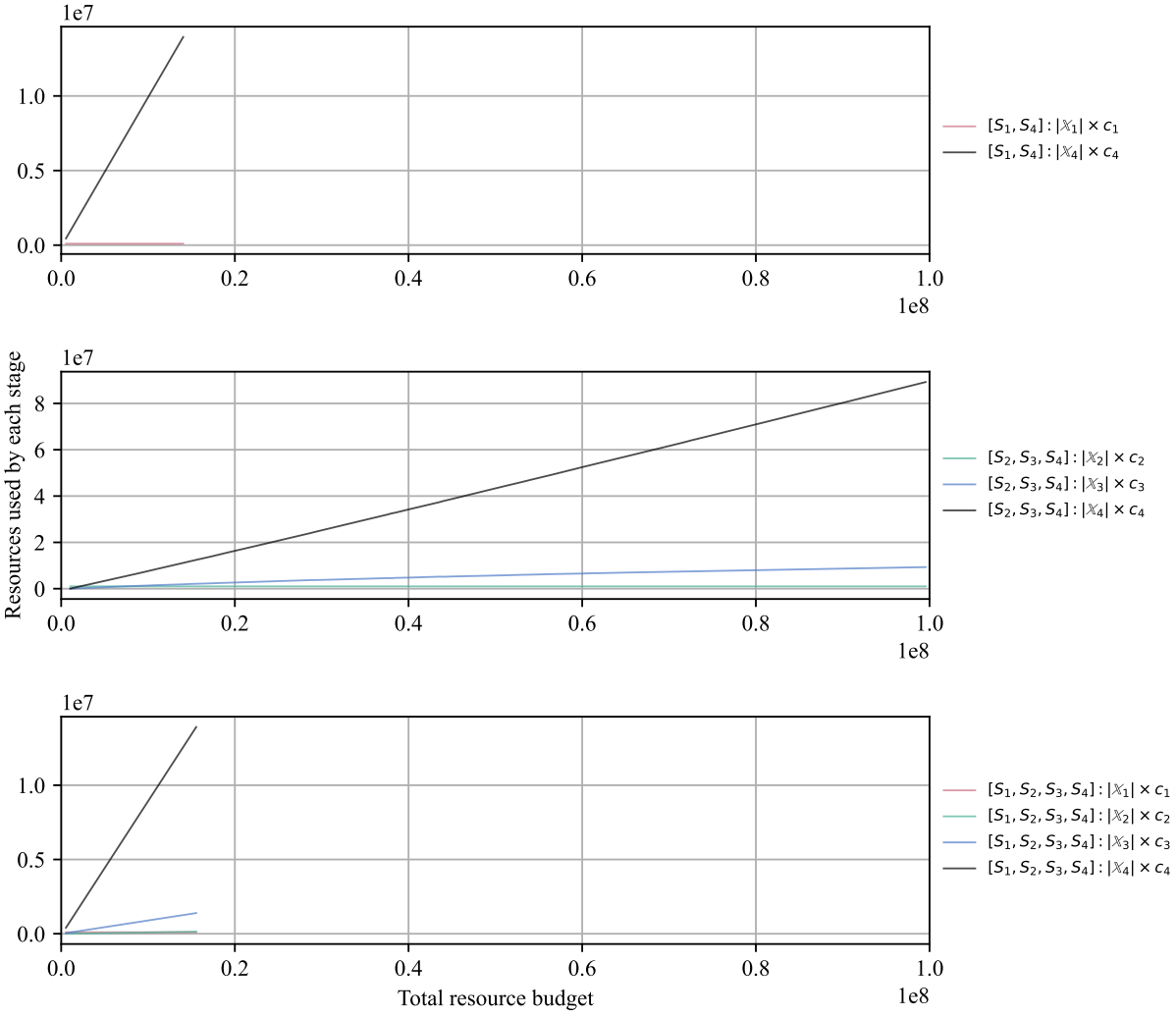


Figure D.16: Resources used by each stage in scenario 4.

$$p\left(y_1, y_2, y_3, y_4\right) \sim \mathcal{N}\left(\boldsymbol{\mu}=\mathbf{0}, \boldsymbol{\Sigma}=\begin{bmatrix} 1 & 0.2 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 & 0.8 \\ 0.2 & 0.2 & 1 & 0.2 \\ 0.2 & 0.8 & 0.2 & 1 \end{bmatrix}\right)$$

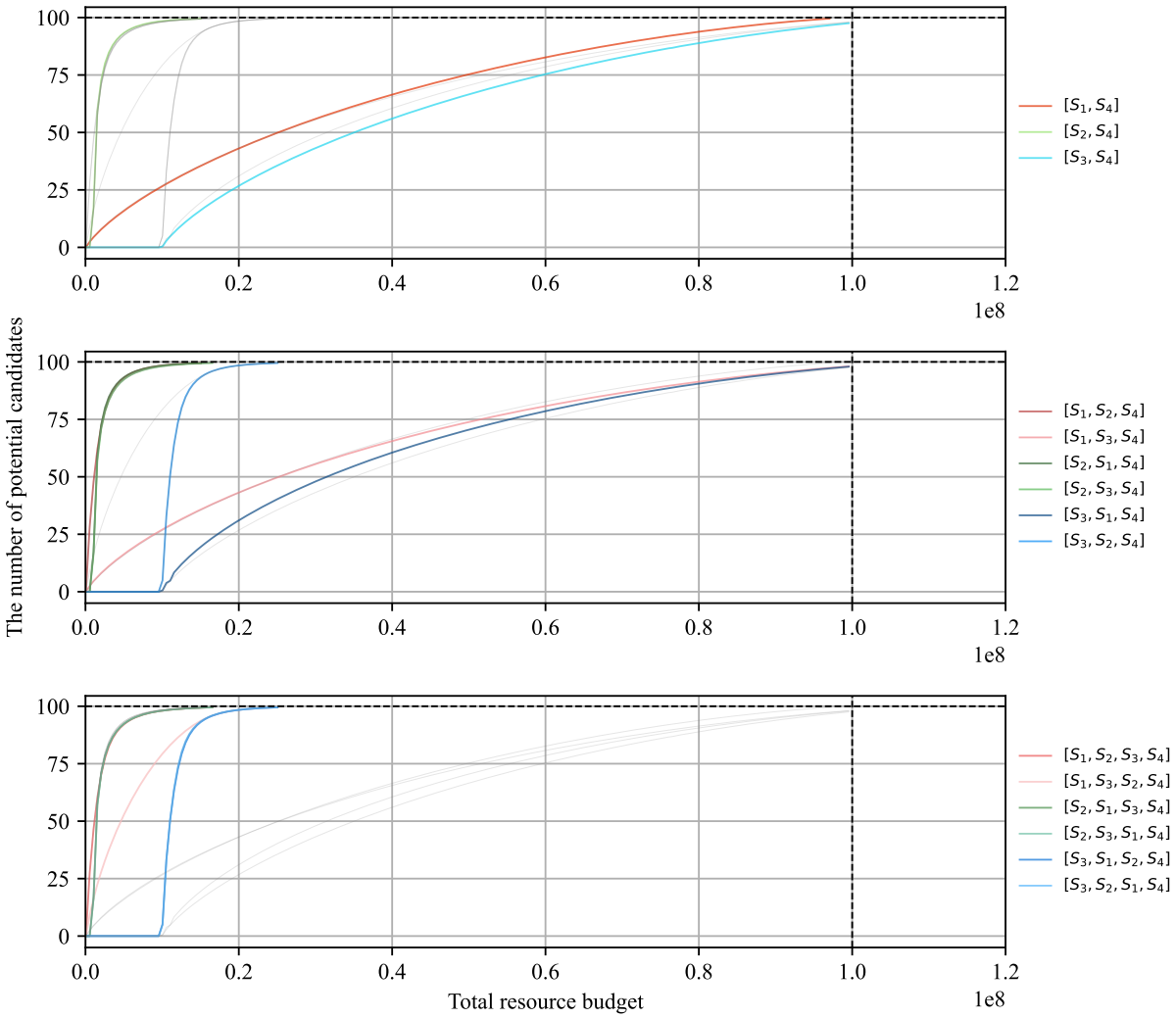


Figure D.17: Performance comparison of the optimized HTVS pipelines in terms of discovery capability in scenario 5.

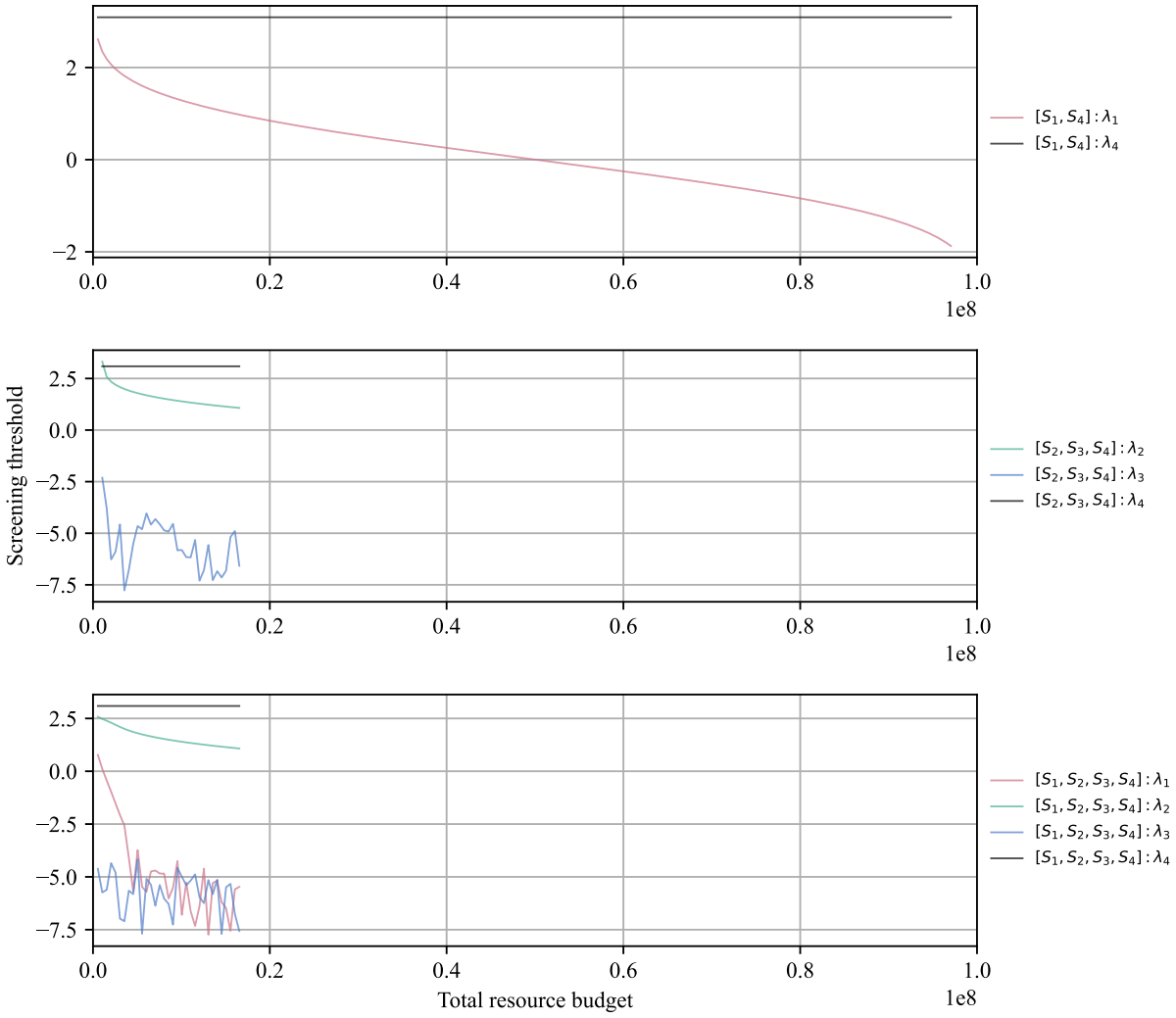


Figure D.18: Screening thresholds of the optimized pipelines in scenario 5.

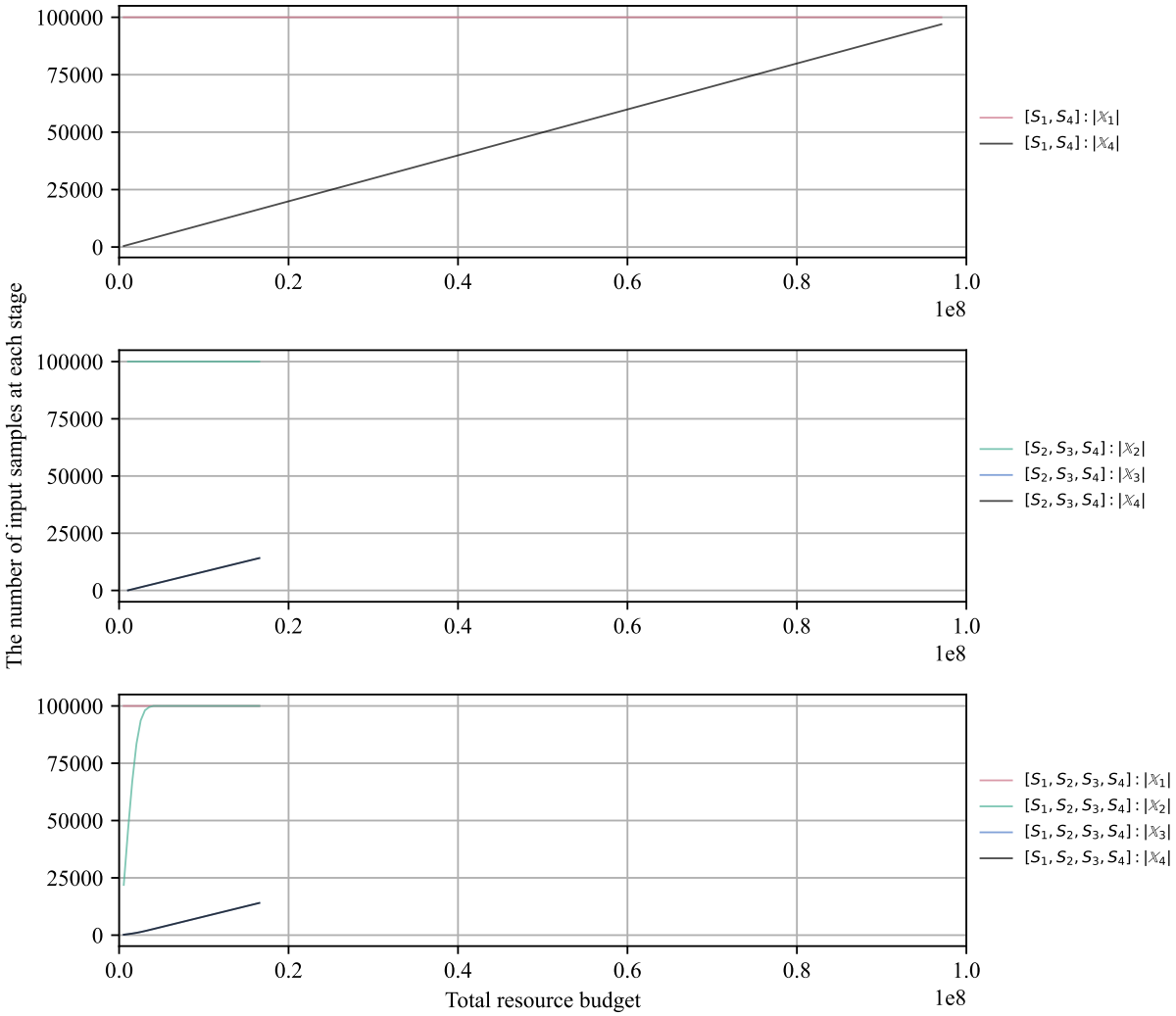


Figure D.19: The number of input samples at each stage in scenario 5.

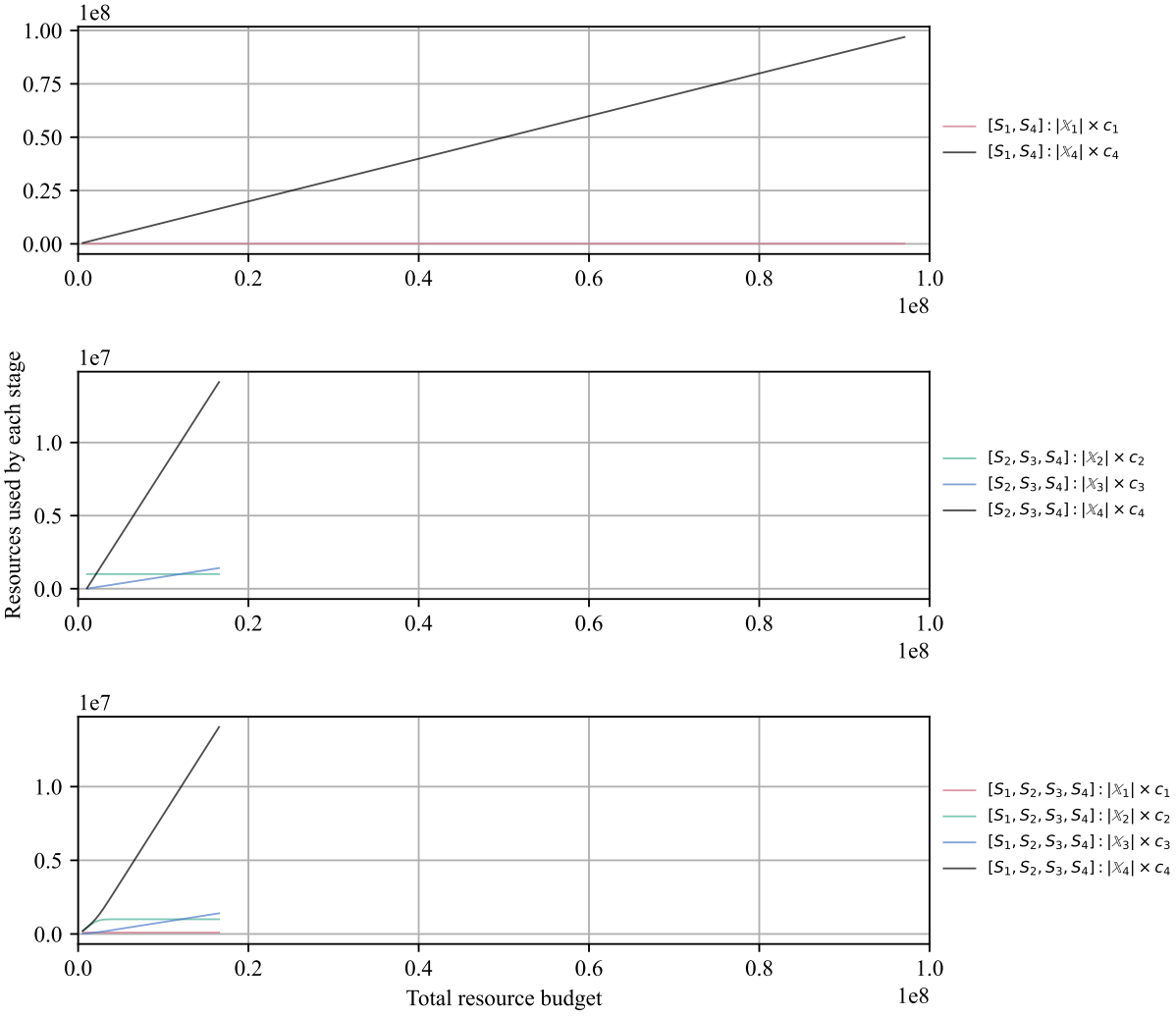


Figure D.20: Resources used by each stage in scenario 5.

$$p\left(y_1, y_2, y_3, y_4\right) \sim \mathcal{N}\left(\boldsymbol{\mu}=\mathbf{0}, \boldsymbol{\Sigma}=\begin{bmatrix} 1 & 0.2 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 & 0.2 \\ 0.2 & 0.2 & 1 & 0.8 \\ 0.2 & 0.2 & 0.8 & 1 \end{bmatrix}\right)$$

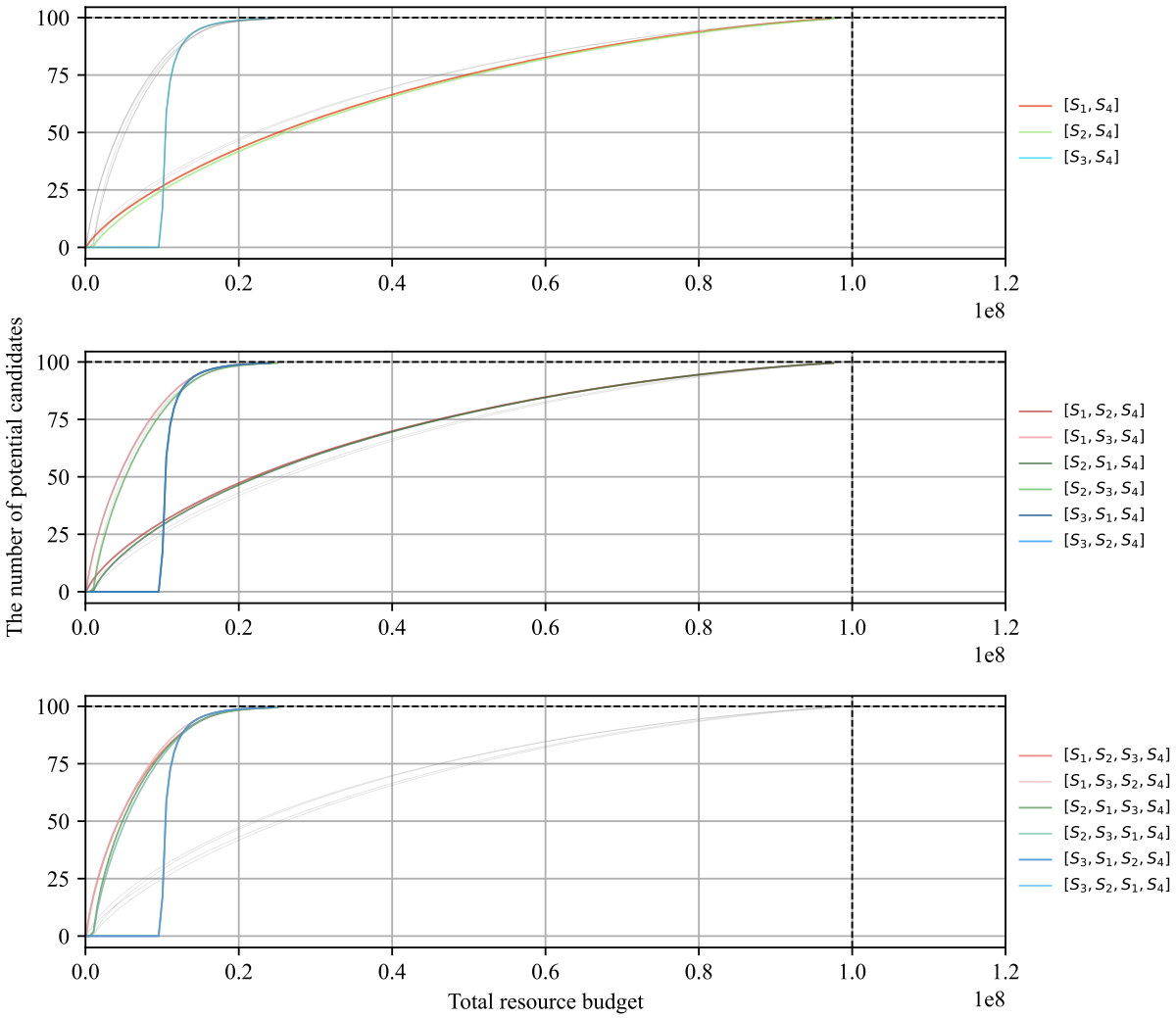


Figure D.21: Performance comparison of the optimized HTVS pipelines in terms of discovery capability in scenario 6.

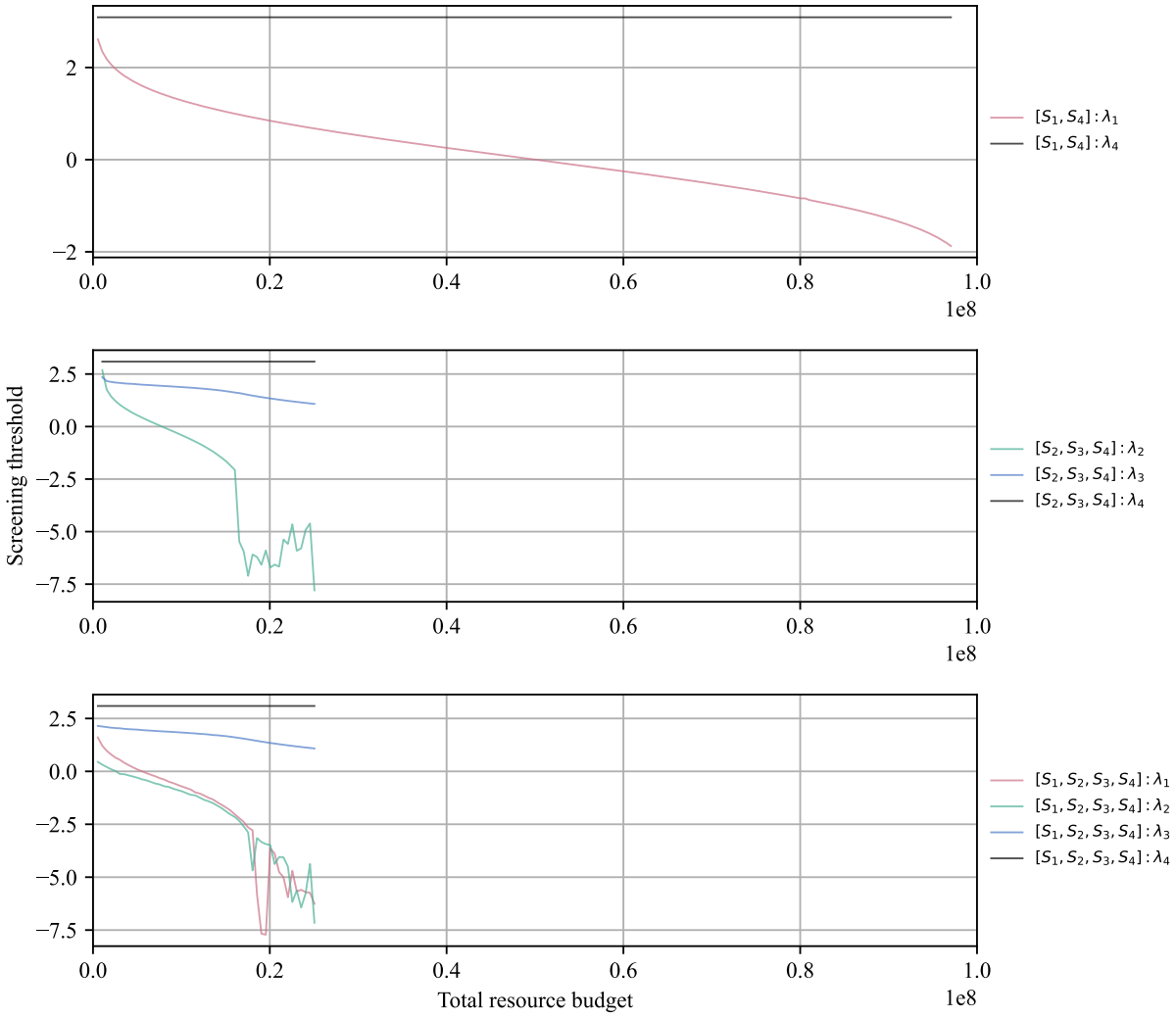


Figure D.22: Screening thresholds of the optimized pipelines in scenario 6.

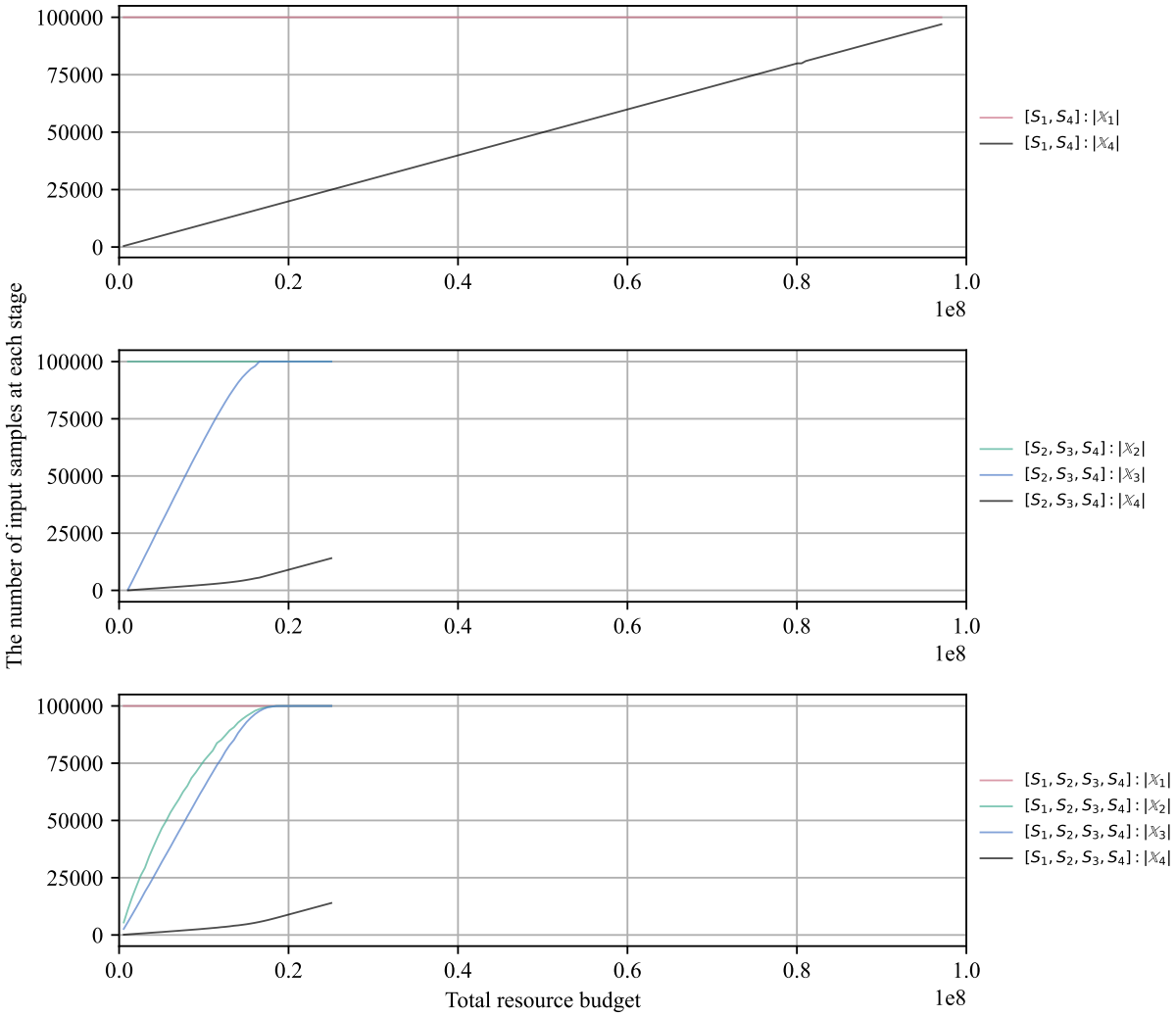


Figure D.23: The number of input samples at each stage in scenario 6.

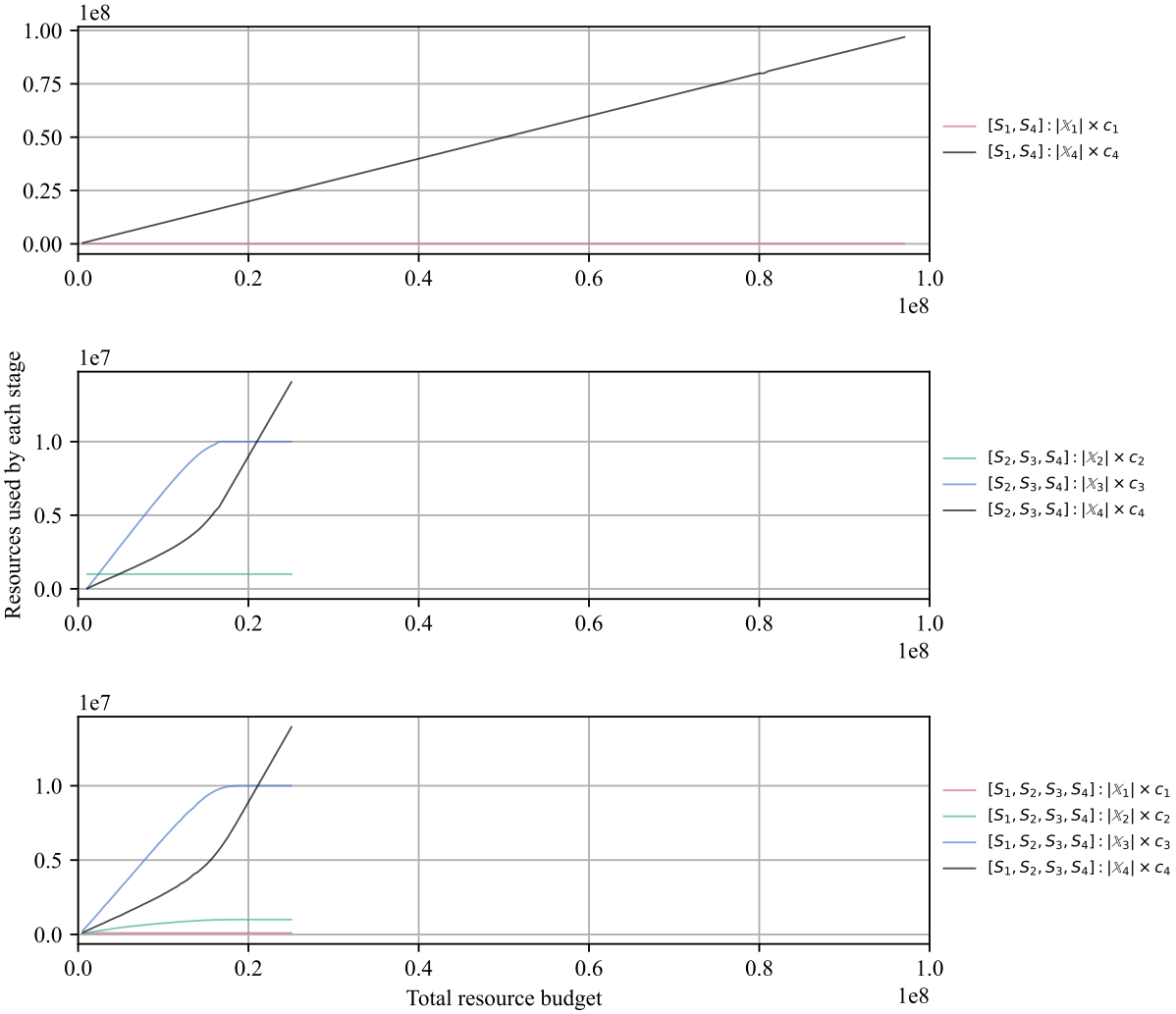


Figure D.24: Resources used by each stage in scenario 6.

$$p\left(y_1, y_2, y_3, y_4\right) \sim \mathcal{N}\left(\boldsymbol{\mu}=\mathbf{0}, \boldsymbol{\Sigma}=\begin{bmatrix} 1 & 0.2 & 0.2 & 0.2 \\ 0.2 & 1 & 0.8 & 0.8 \\ 0.2 & 0.8 & 1 & 0.8 \\ 0.2 & 0.8 & 0.8 & 1 \end{bmatrix}\right)$$

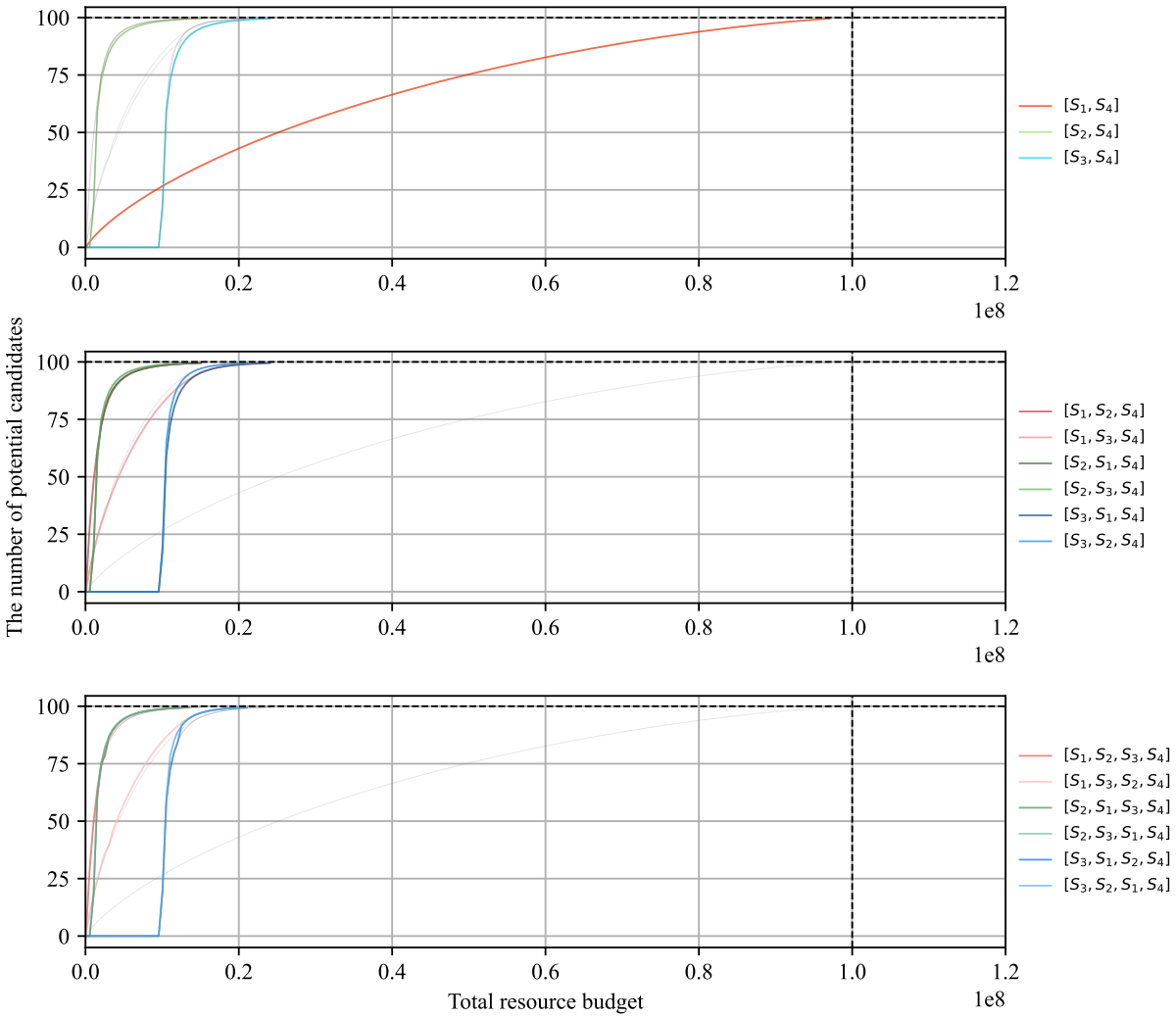


Figure D.25: Performance comparison of the optimized HTVS pipelines in terms of discovery capability in scenario 7.

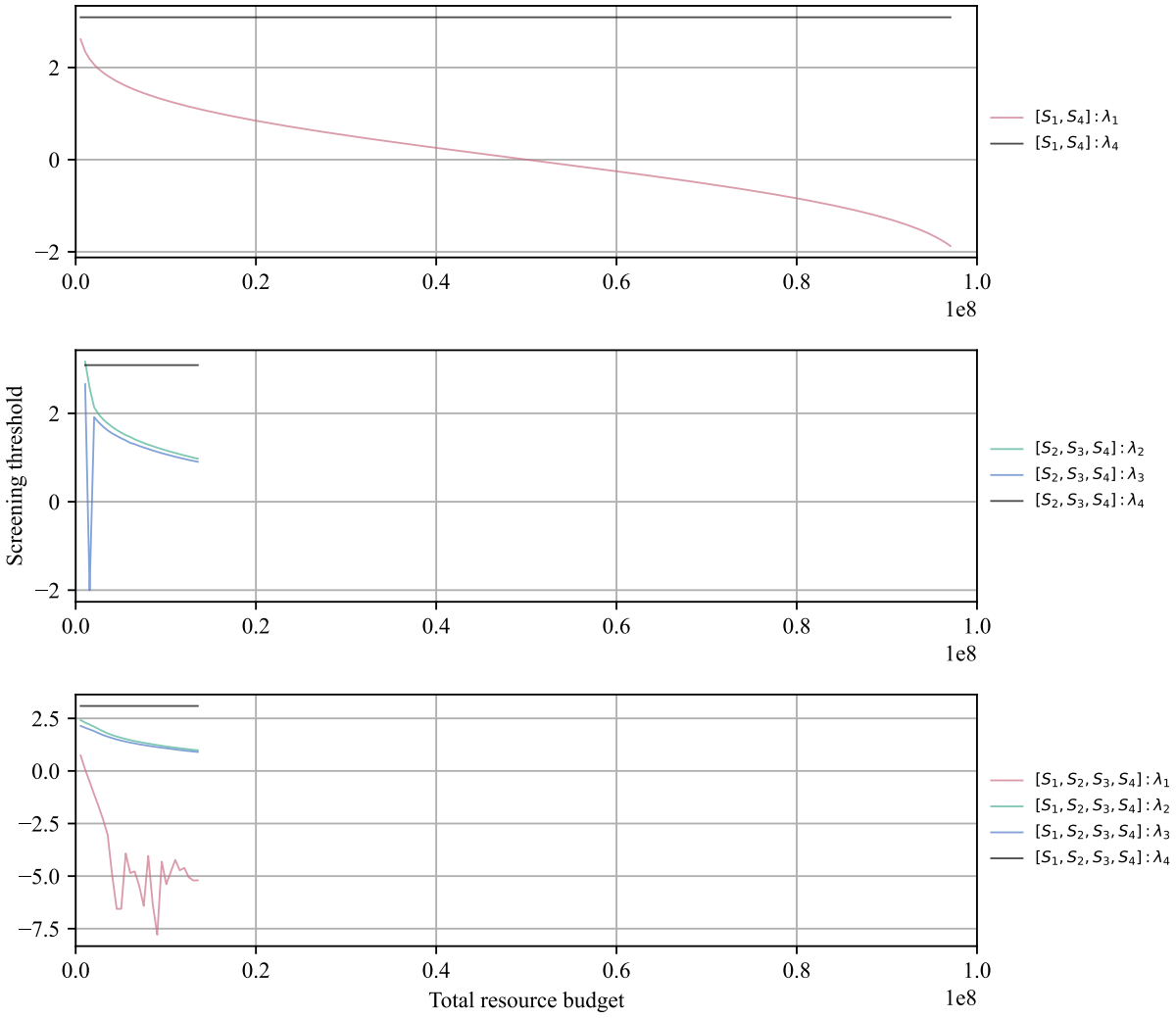


Figure D.26: Screening thresholds of the optimized pipelines in scenario 7.

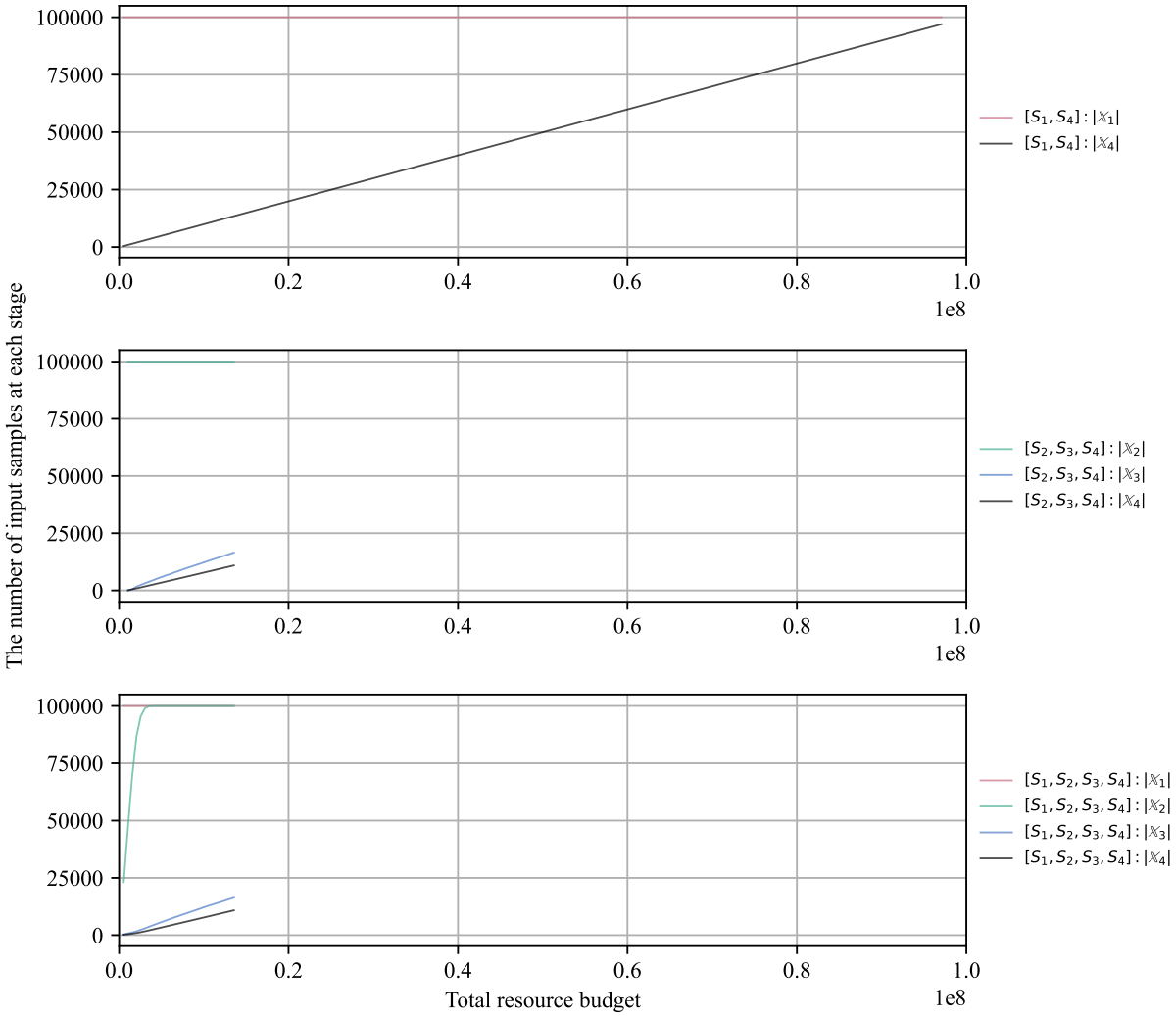


Figure D.27: The number of input samples at each stage in scenario 7.

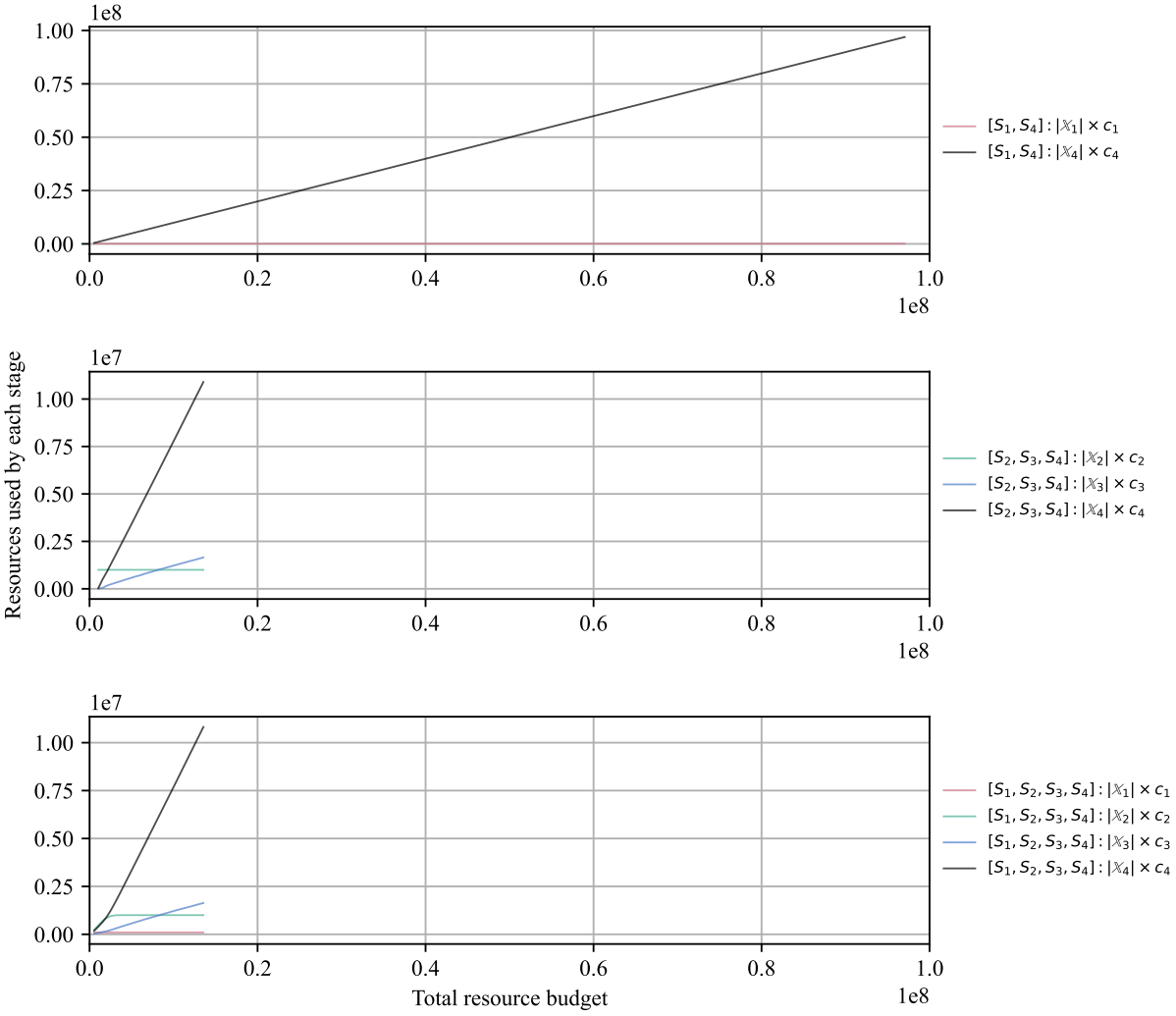


Figure D.28: Resources used by each stage in scenario 7.

$$p\left(y_1, y_2, y_3, y_4\right) \sim \mathcal{N}\left(\boldsymbol{\mu}=\mathbf{0}, \boldsymbol{\Sigma}=\begin{bmatrix} 1 & 0.2 & 0.8 & 0.8 \\ 0.2 & 1 & 0.2 & 0.2 \\ 0.8 & 0.2 & 1 & 0.8 \\ 0.8 & 0.2 & 0.8 & 1 \end{bmatrix}\right)$$

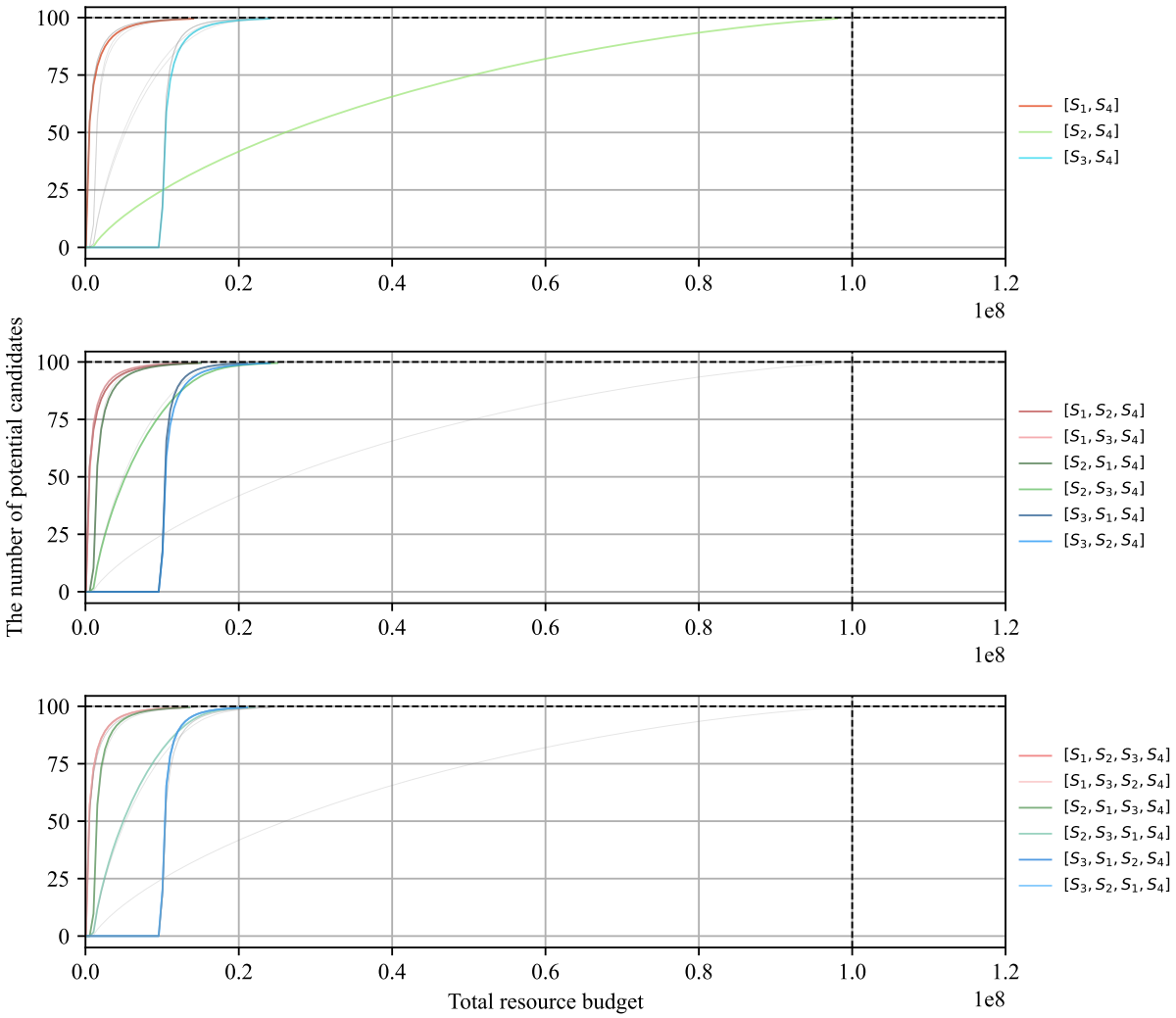


Figure D.29: Performance comparison of the optimized HTVS pipelines in terms of discovery capability in scenario 8.

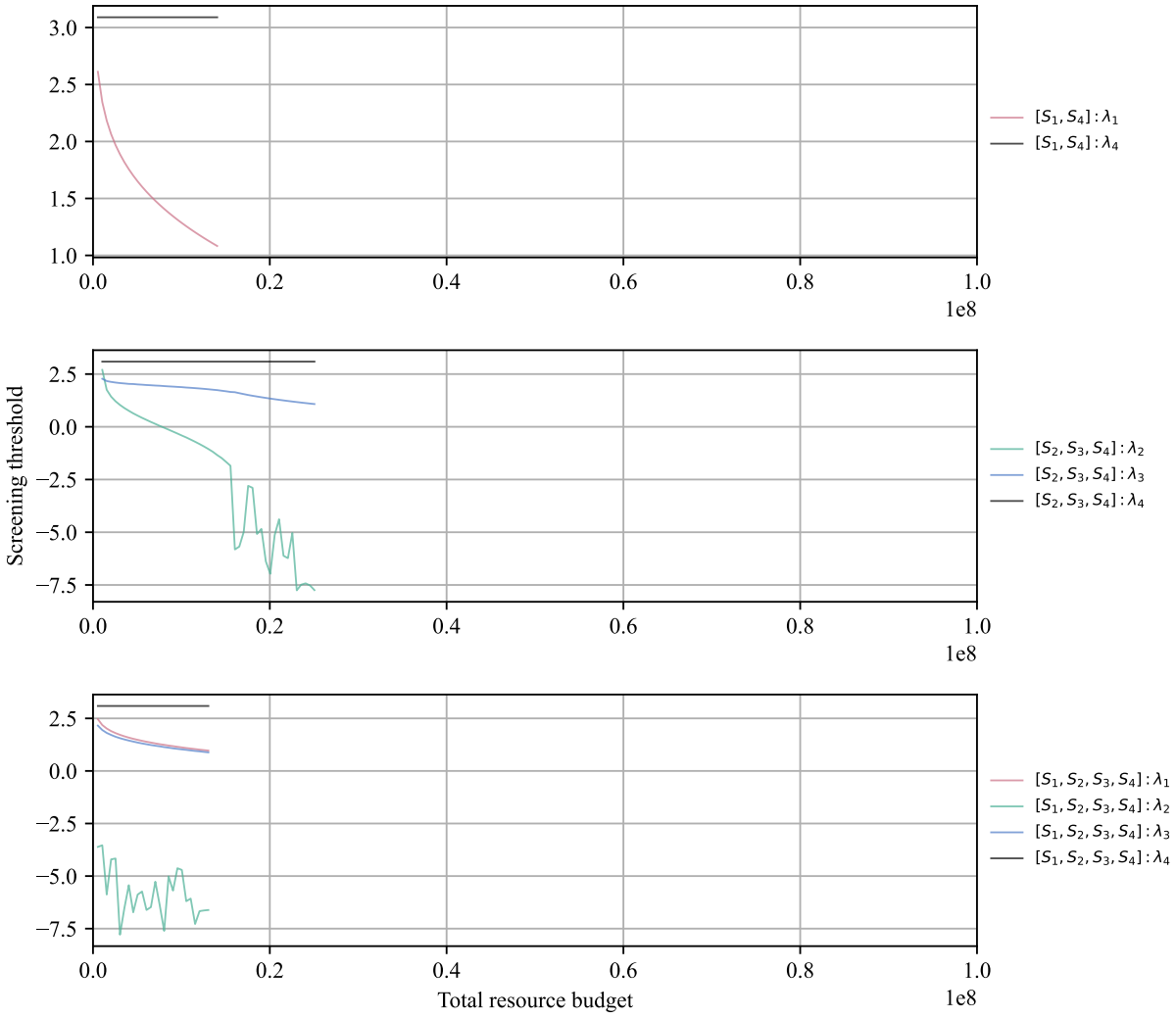


Figure D.30: Screening thresholds of the optimized pipelines in scenario 8.

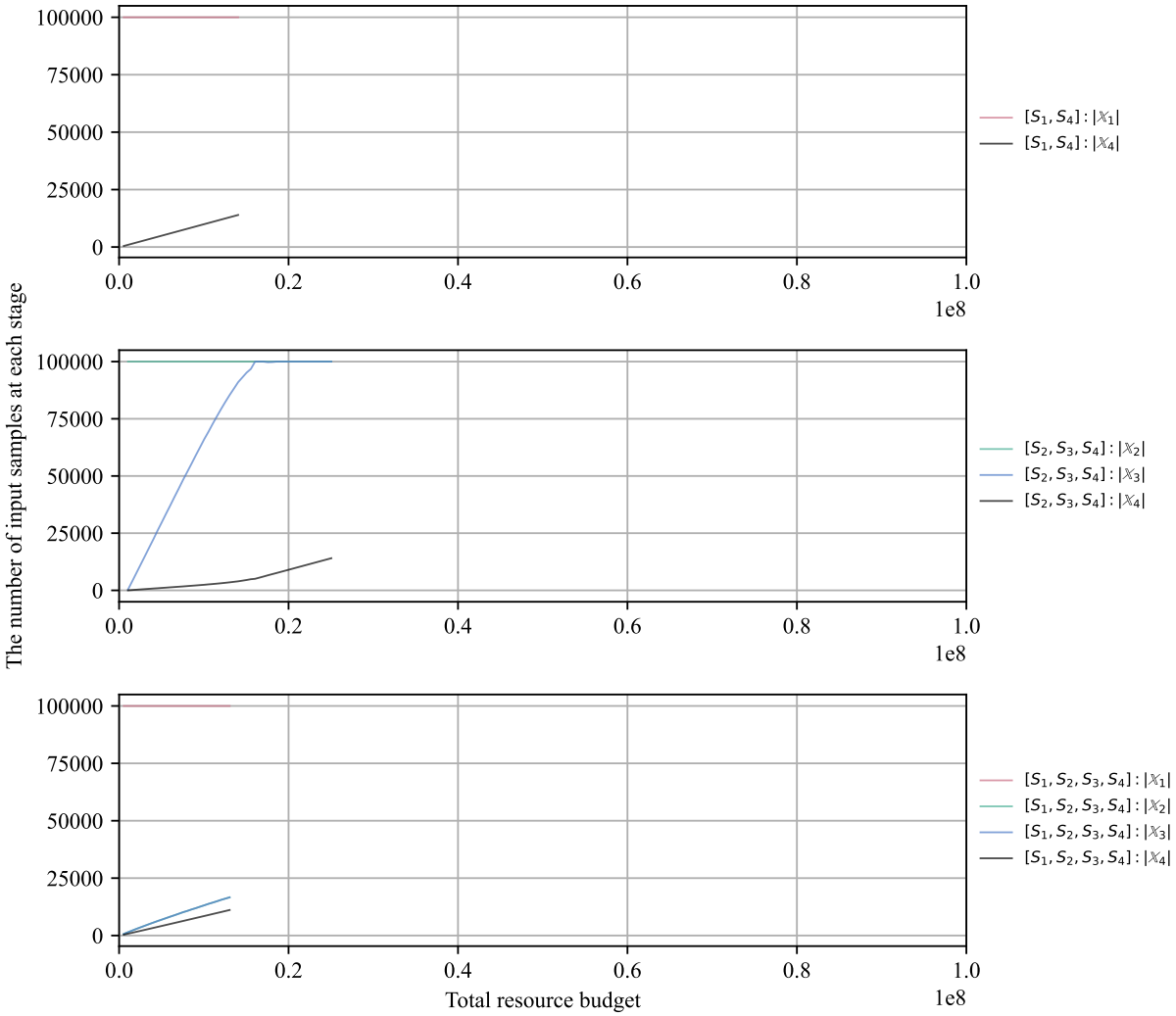


Figure D.31: The number of input samples at each stage in scenario 8.

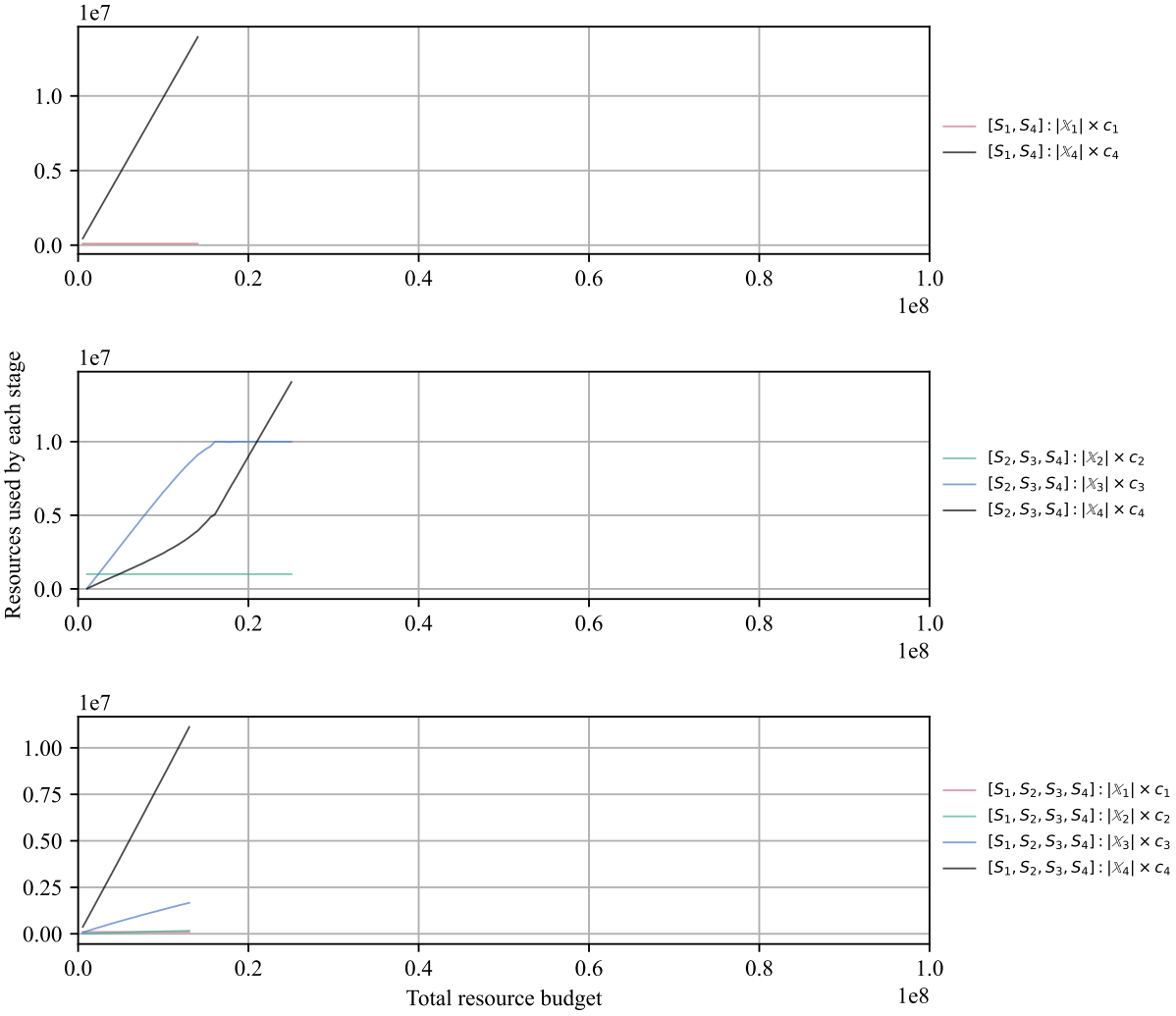


Figure D.32: Resources used by each stage in scenario 8.

$$p\left(y_1, y_2, y_3, y_4\right) \sim \mathcal{N}\left(\boldsymbol{\mu}=\mathbf{0}, \boldsymbol{\Sigma}=\begin{bmatrix} 1 & 0.8 & 0.2 & 0.8 \\ 0.8 & 1 & 0.2 & 0.8 \\ 0.2 & 0.2 & 1 & 0.2 \\ 0.8 & 0.8 & 0.2 & 1 \end{bmatrix}\right)$$

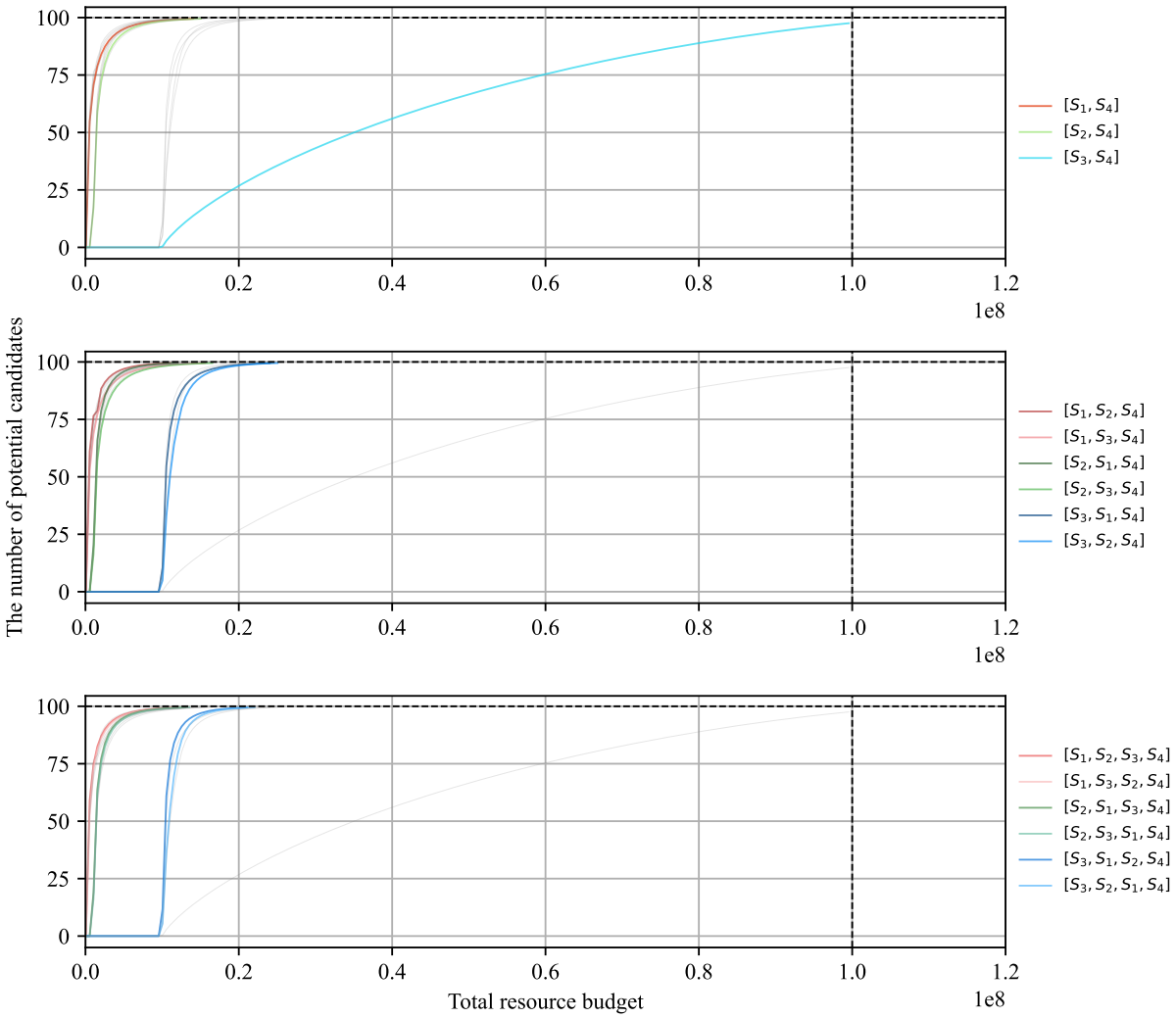


Figure D.33: Performance comparison of the optimized HTVS pipelines in terms of discovery capability in scenario 9.

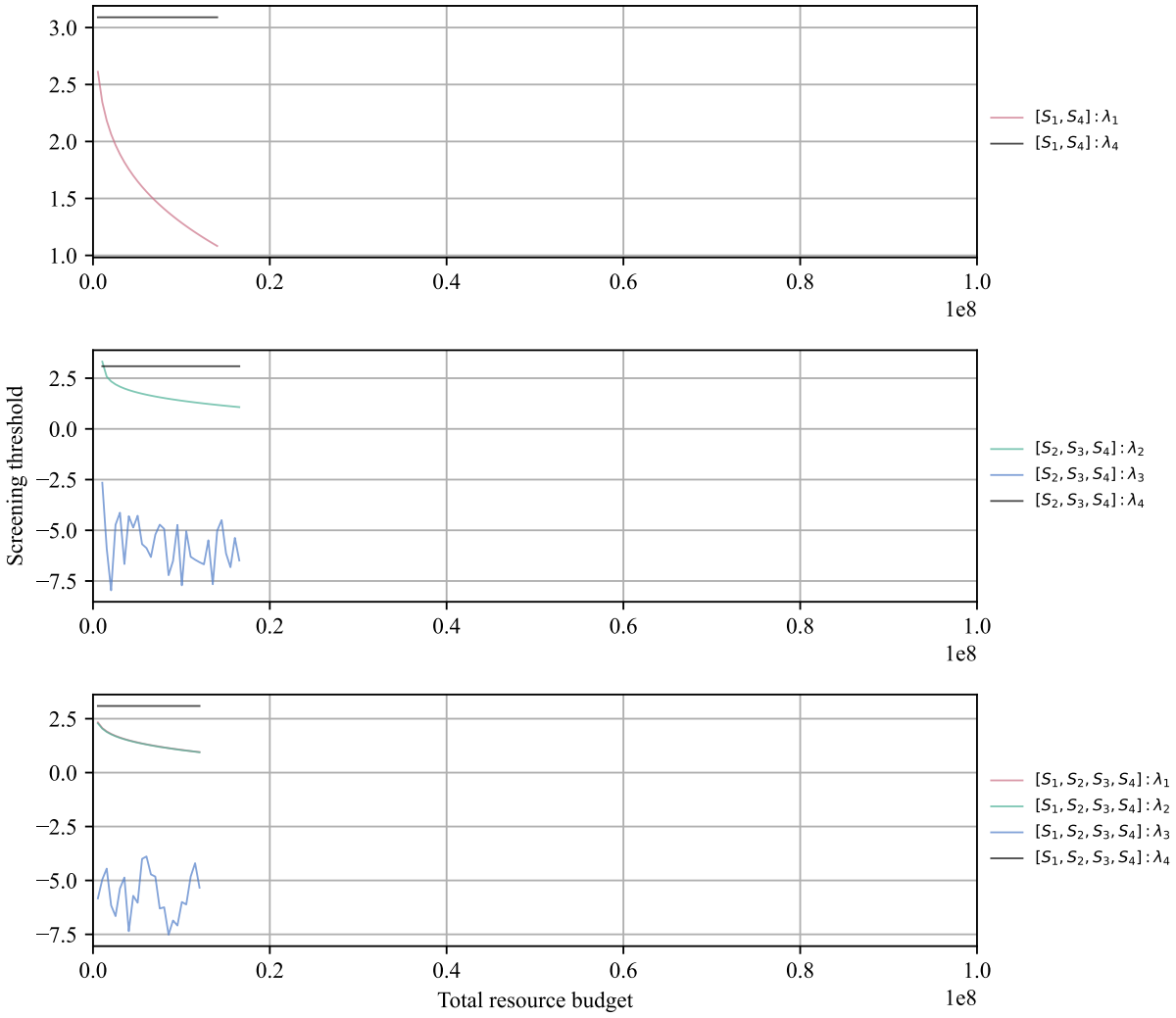


Figure D.34: Screening thresholds of the optimized pipelines in scenario 9.

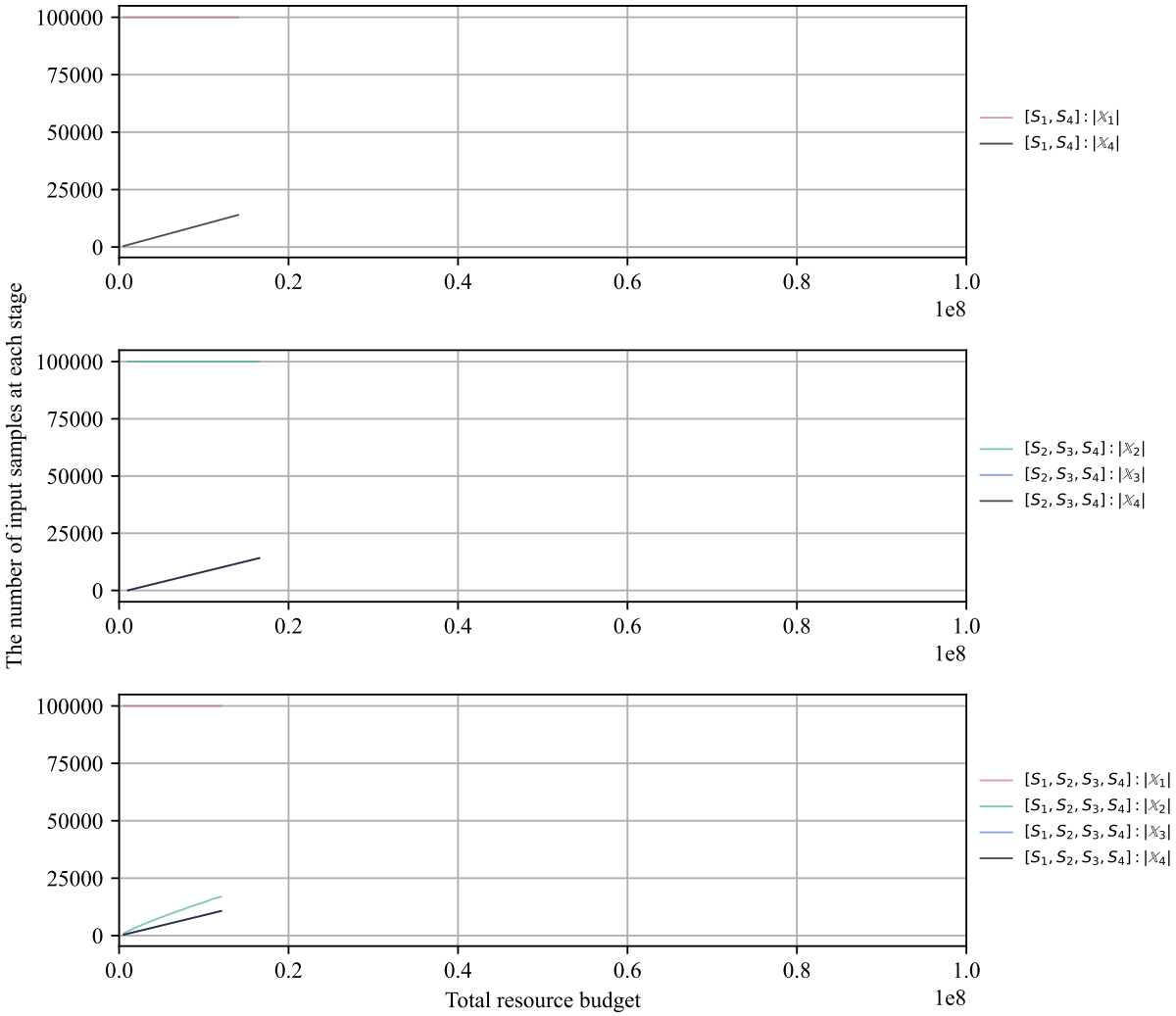


Figure D.35: The number of input samples at each stage in scenario 9.

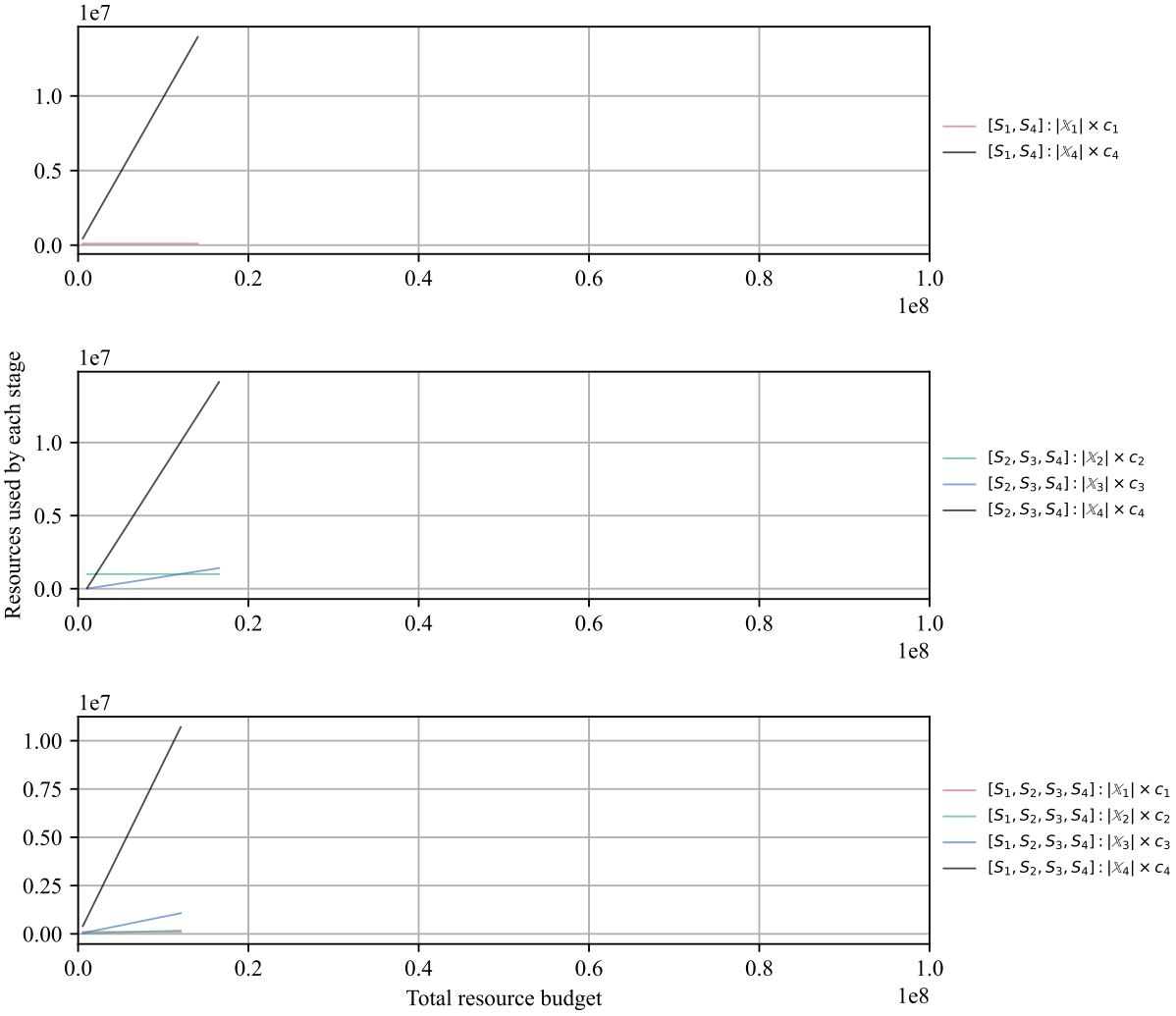


Figure D.36: Resources used by each stage in scenario 9.

$$p\left(y_1, y_2, y_3, y_4\right) \sim \mathcal{N}\left(\boldsymbol{\mu}=\mathbf{0}, \boldsymbol{\Sigma}=\begin{bmatrix} 1 & 0.3 & 0.2 & 0.1 \\ 0.3 & 1 & 0.3 & 0.2 \\ 0.2 & 0.3 & 1 & 0.3 \\ 0.1 & 0.2 & 0.3 & 1 \end{bmatrix}\right)$$

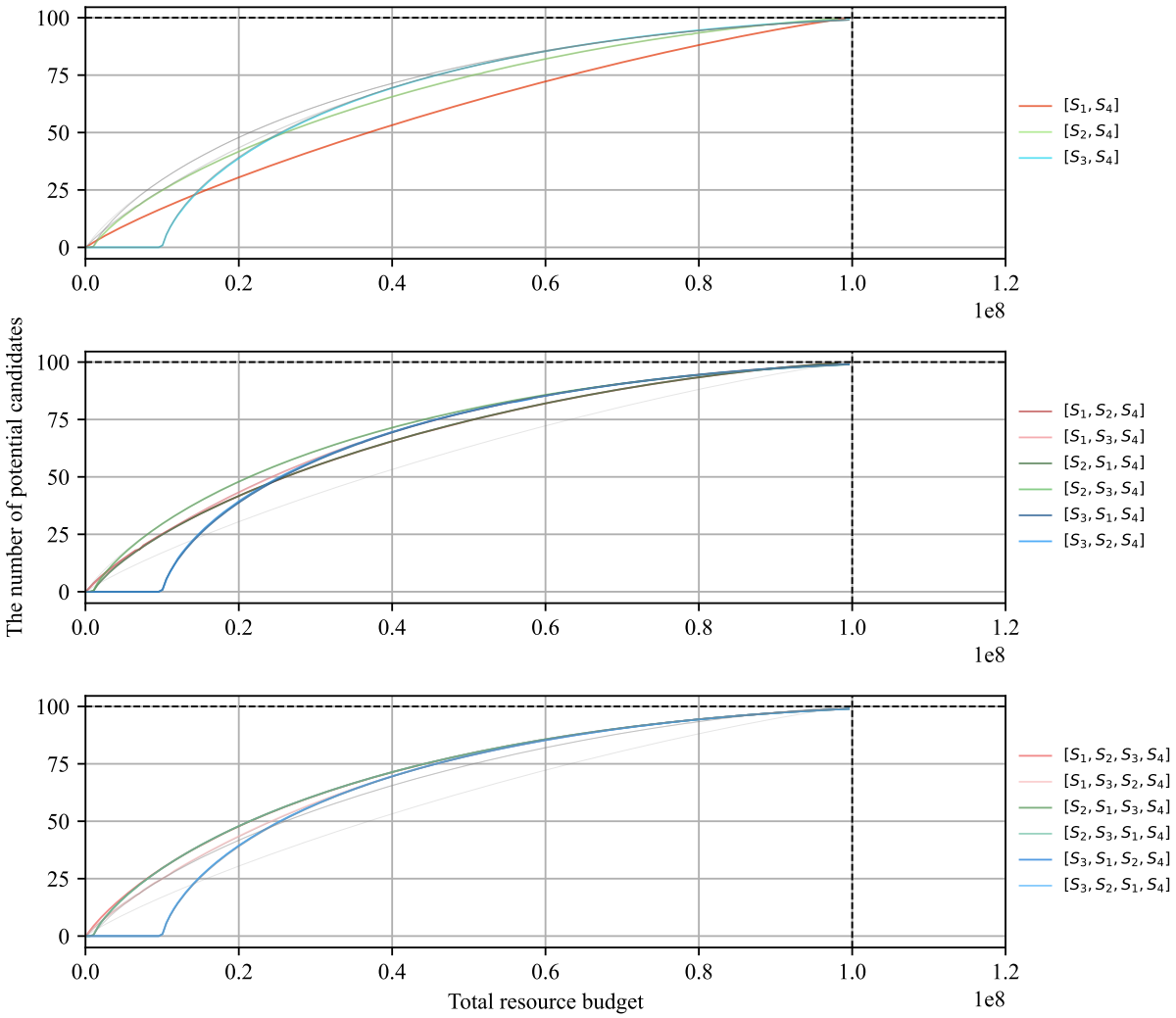


Figure D.37: Performance comparison of the optimized HTVS pipelines in terms of discovery capability in scenario 10.

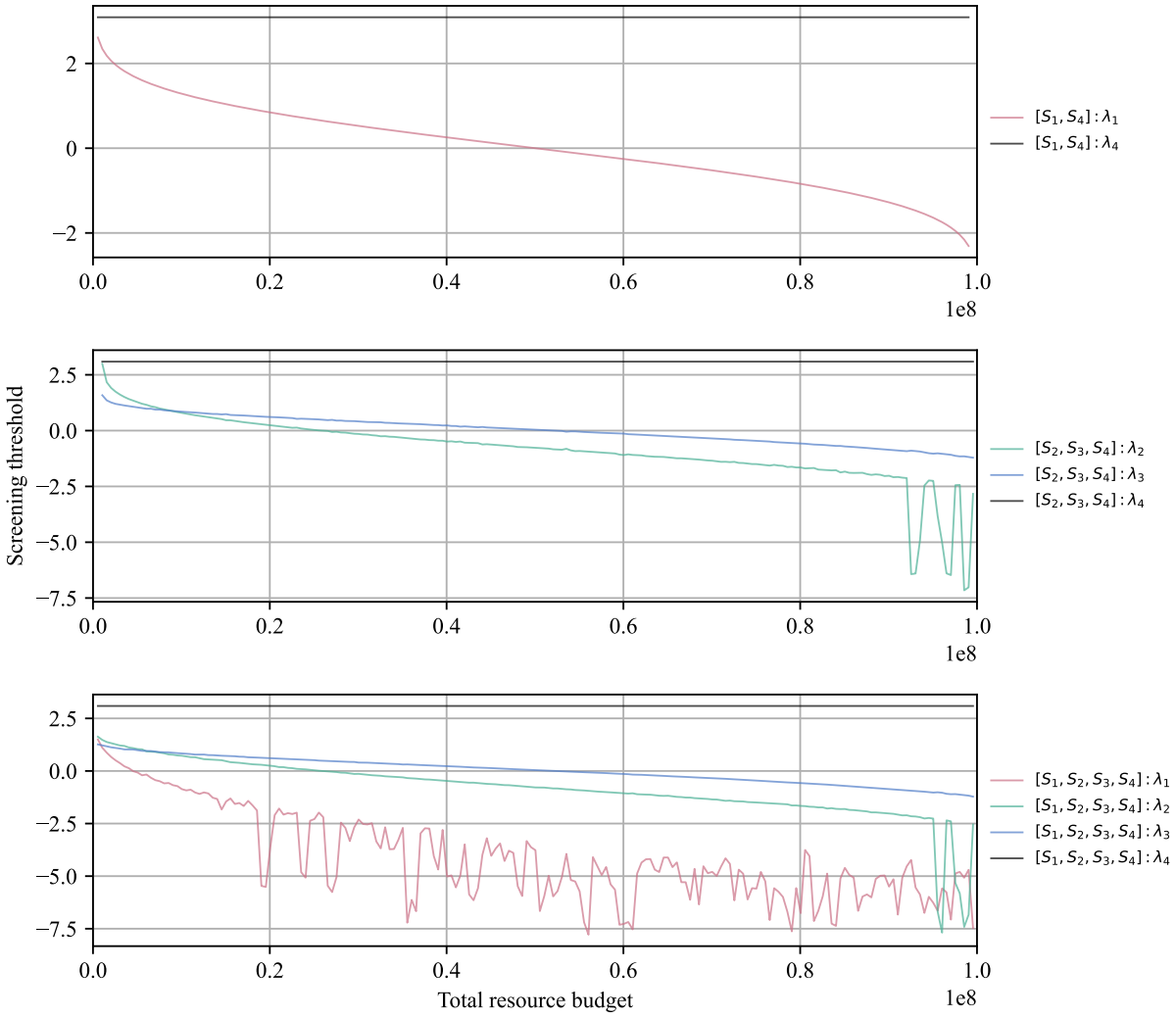


Figure D.38: Screening thresholds of the optimized pipelines in scenario 10.

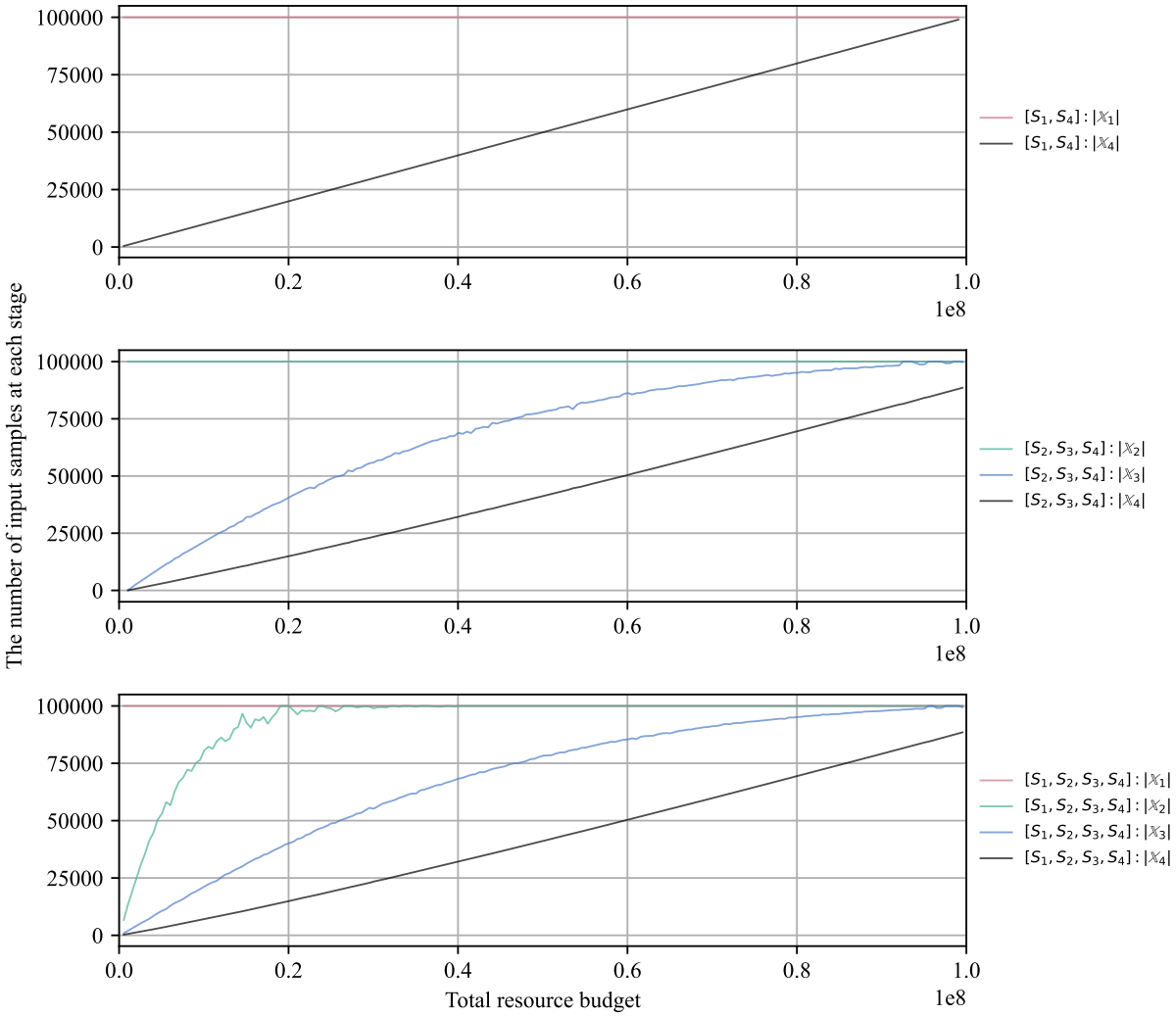


Figure D.39: The number of input samples at each stage in scenario 10.

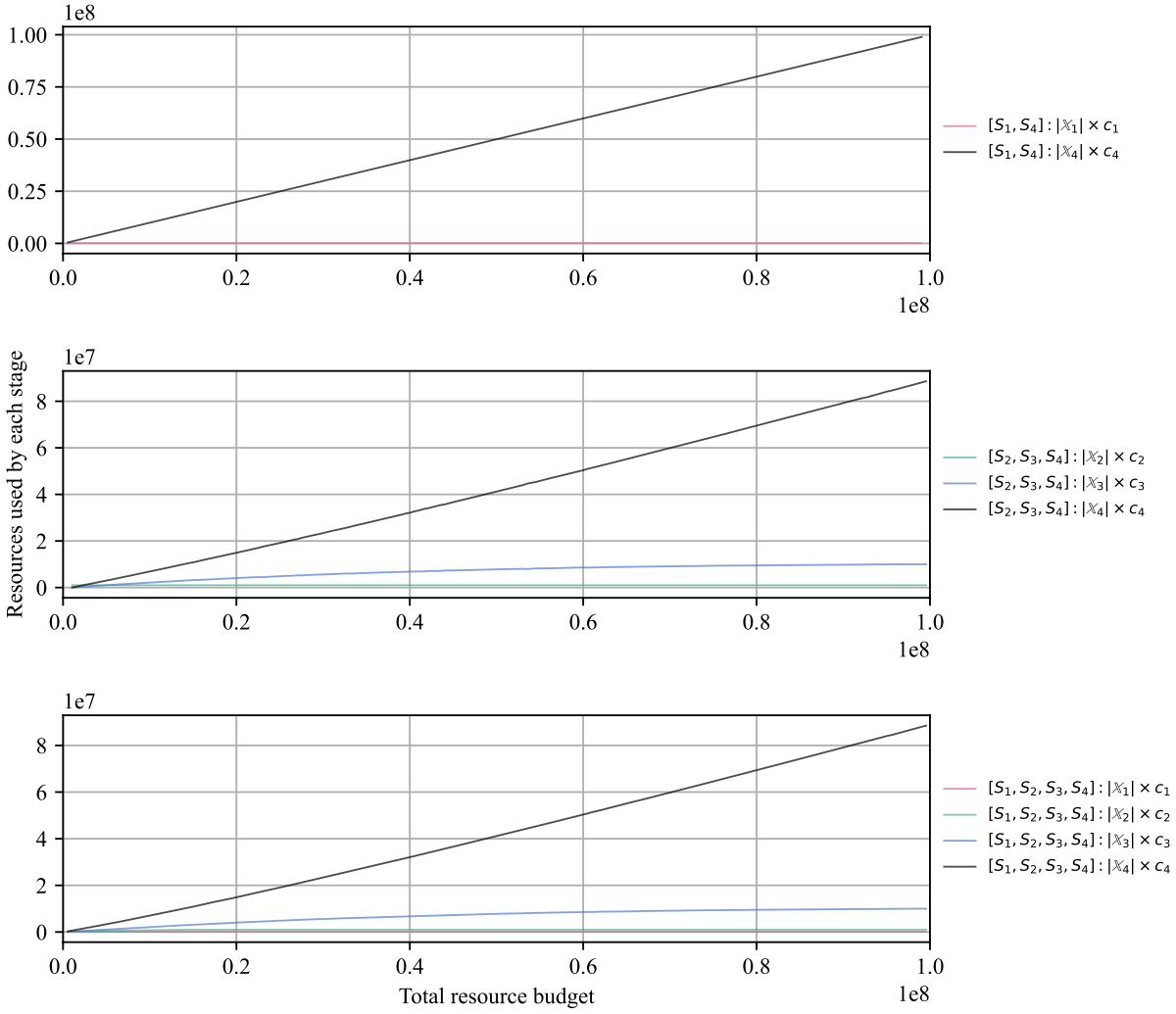


Figure D.40: Resources used by each stage in scenario 10.

$$p\left(y_1, y_2, y_3, y_4\right) \sim \mathcal{N}\left(\boldsymbol{\mu}=\mathbf{0}, \boldsymbol{\Sigma}=\begin{bmatrix} 1 & 0.8 & 0.7 & 0.6 \\ 0.8 & 1 & 0.8 & 0.7 \\ 0.7 & 0.8 & 1 & 0.8 \\ 0.6 & 0.7 & 0.8 & 1 \end{bmatrix}\right)$$

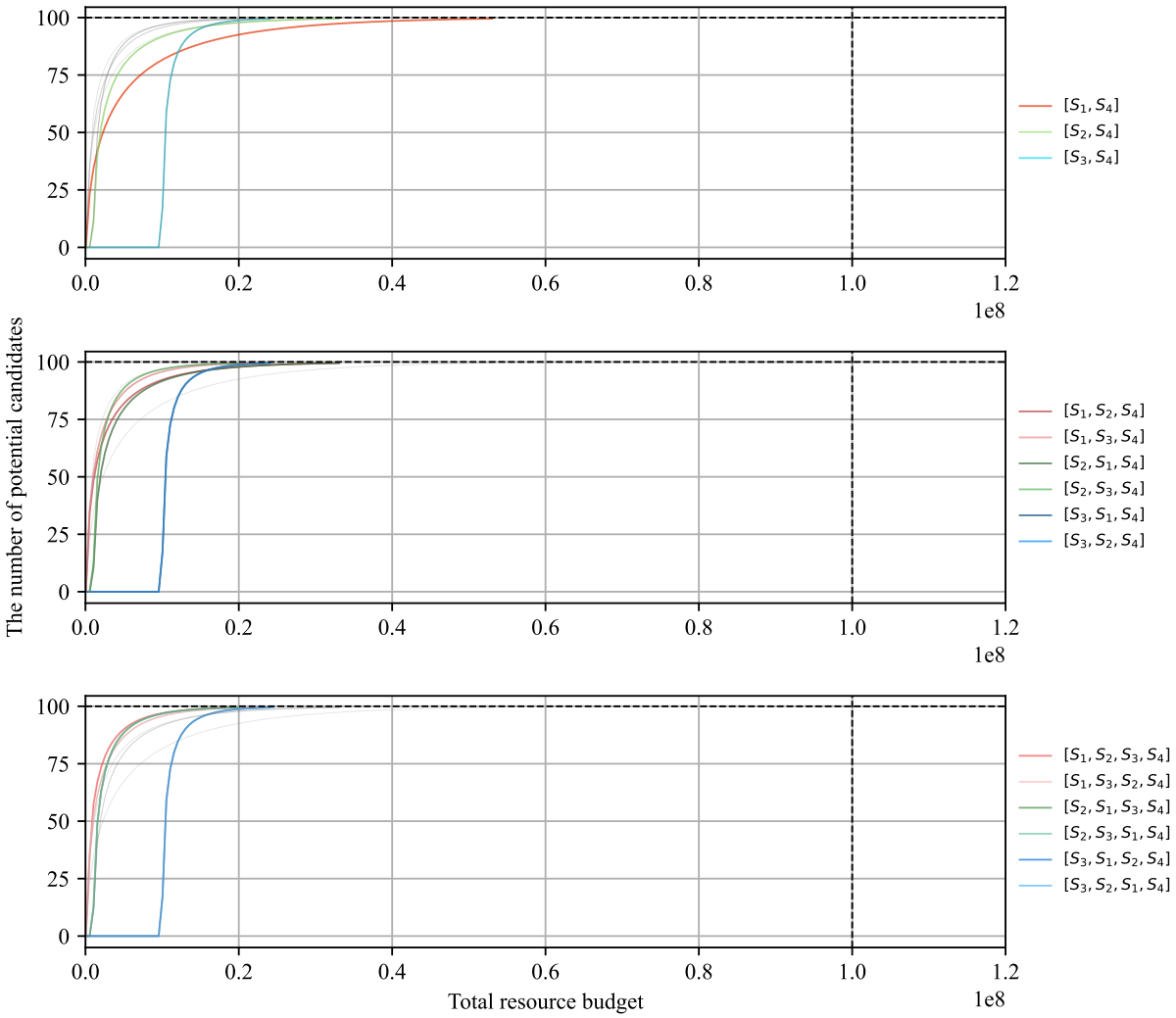


Figure D.41: Performance comparison of the optimized HTVS pipelines in terms of discovery capability in scenario 11.

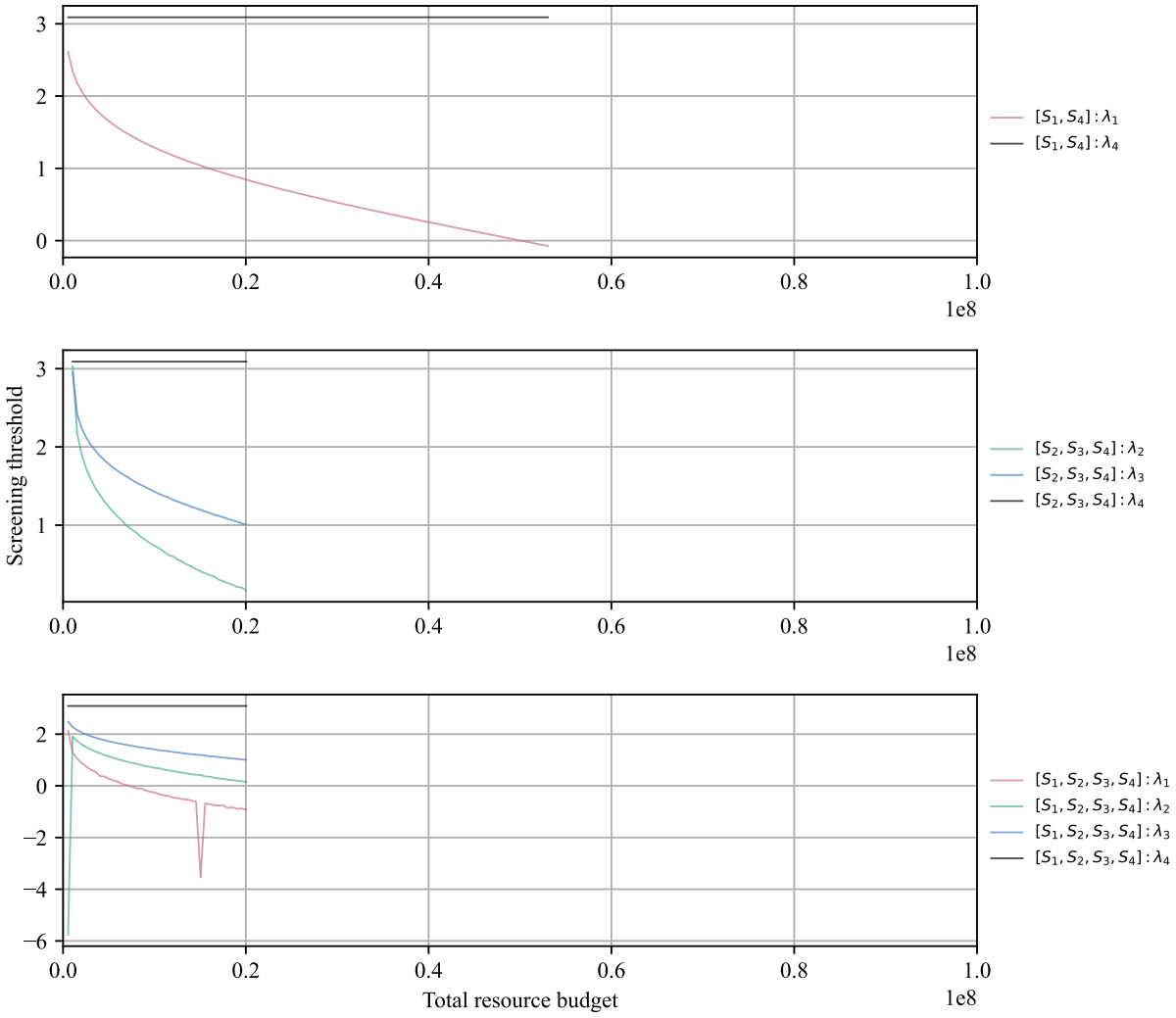


Figure D.42: Screening thresholds of the optimized pipelines in scenario 11.

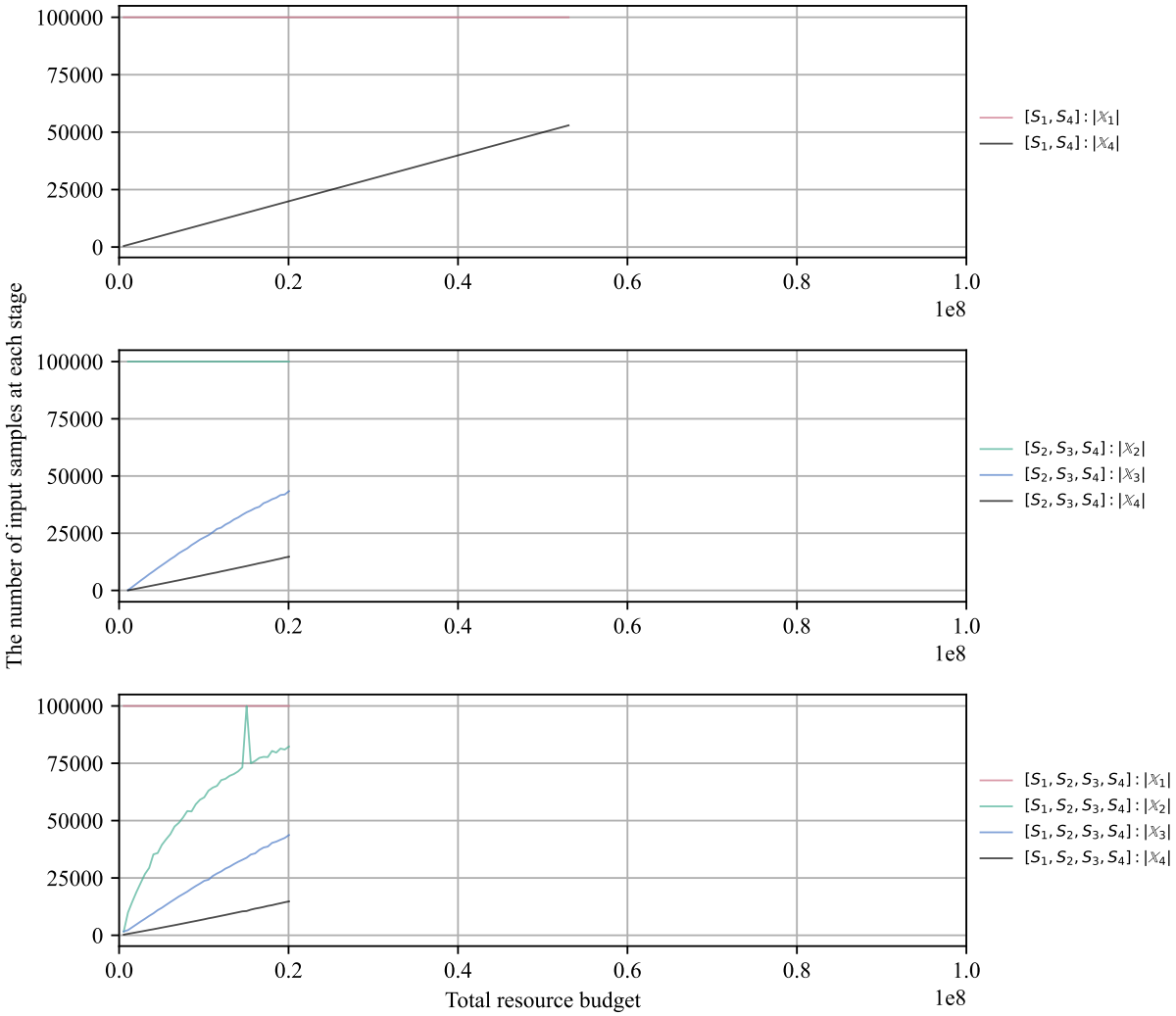


Figure D.43: The number of input samples at each stage in scenario 11.

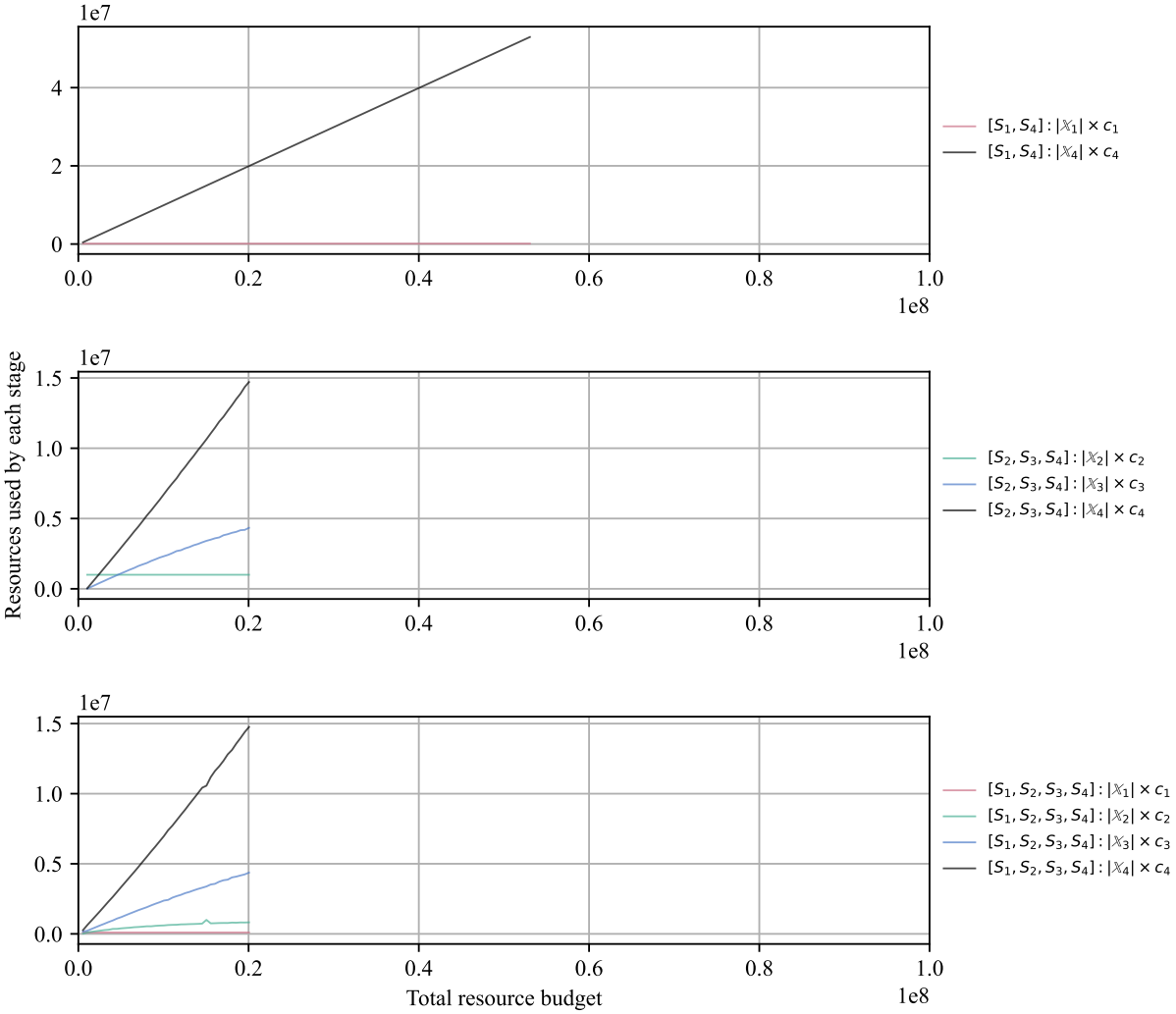


Figure D.44: Resources used by each stage in scenario 11.

APPENDIX E

PERFORMANCE OF OPTIMIZED HTVS PIPELINES FOR LONG NON-CODING RNA DETECTION

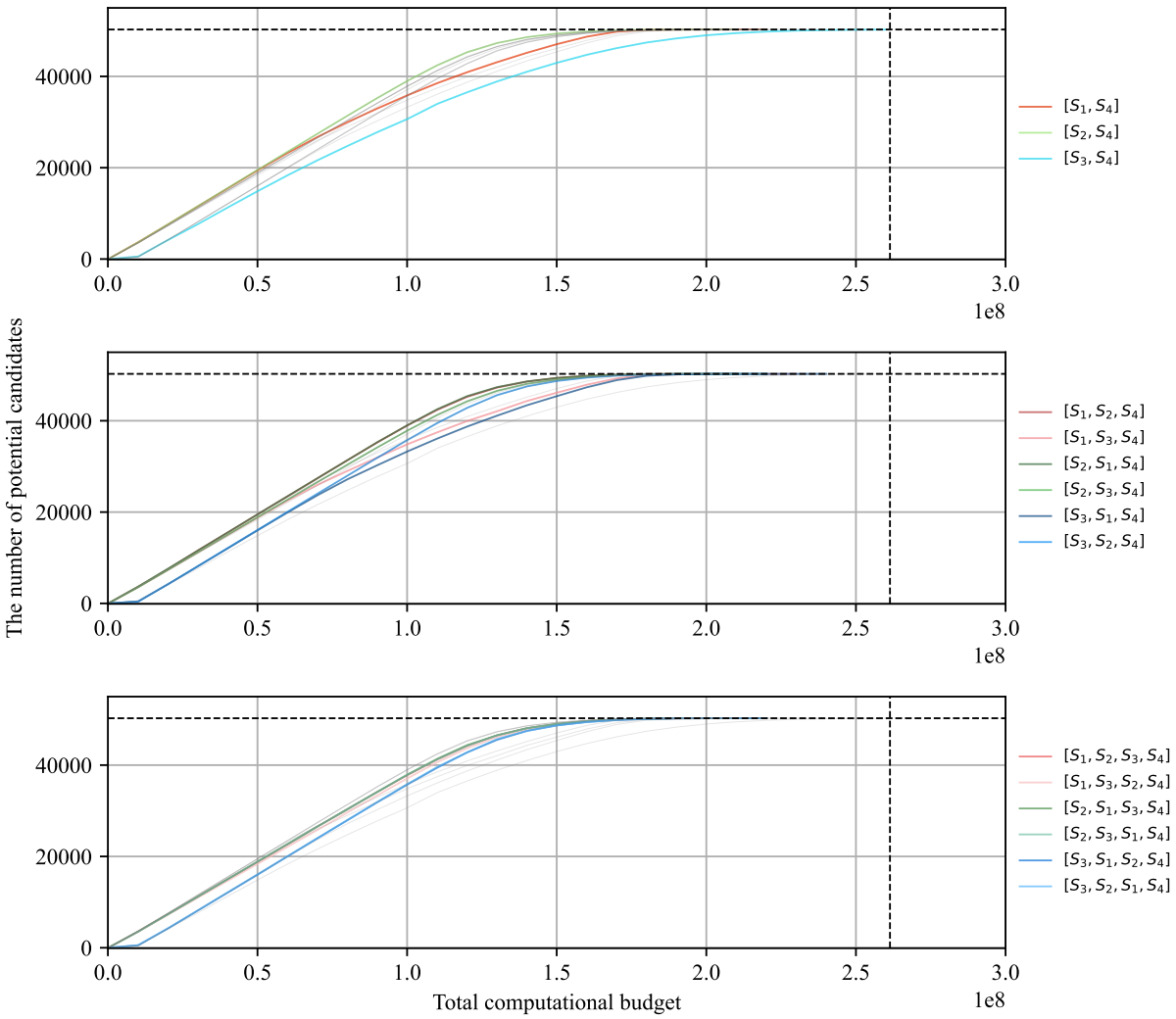


Figure E.1: Performance comparison of the optimized HTVS pipelines in terms of discovery capability.

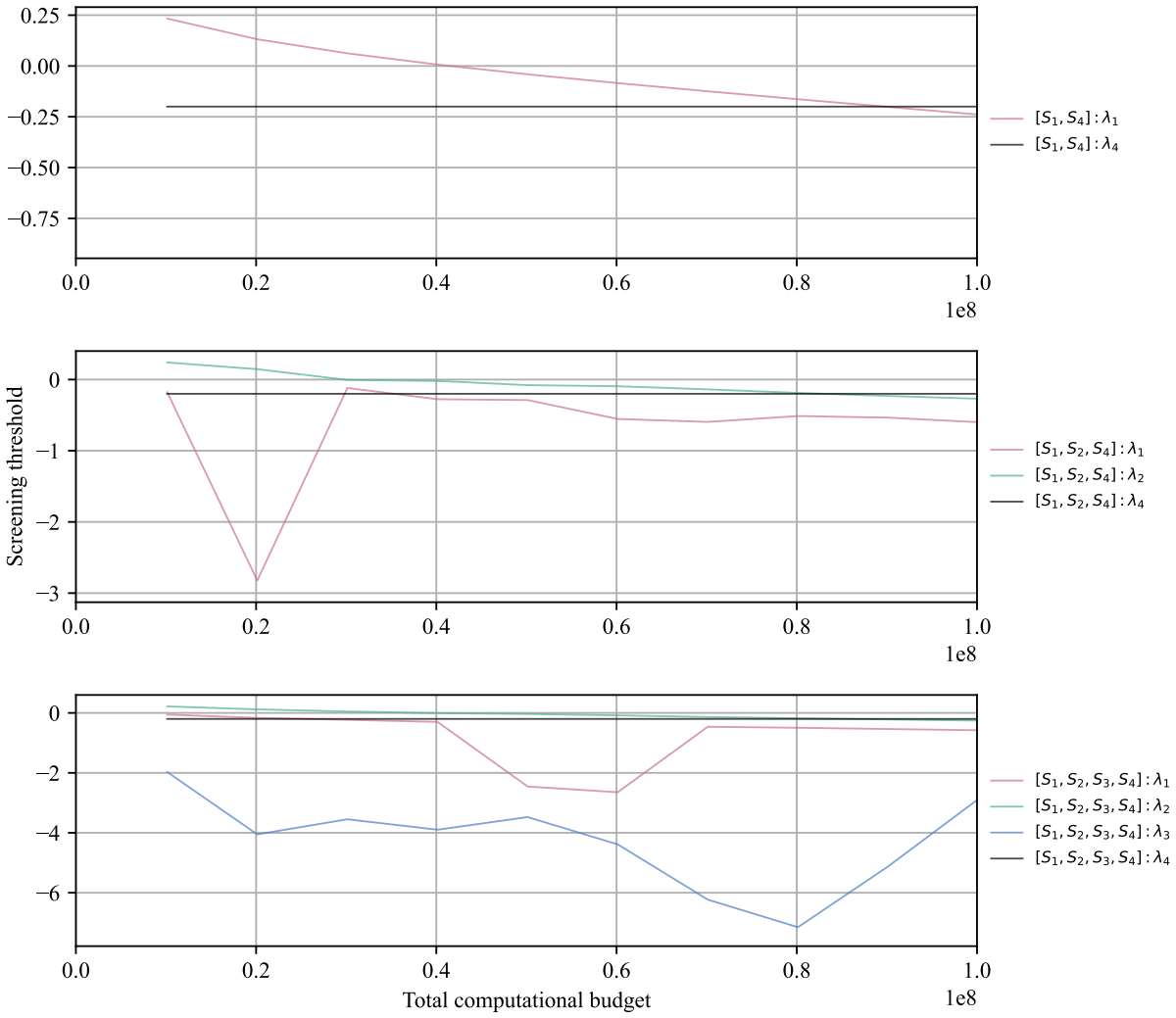


Figure E.2: Screening thresholds of the optimized pipelines for lncRNA detection.

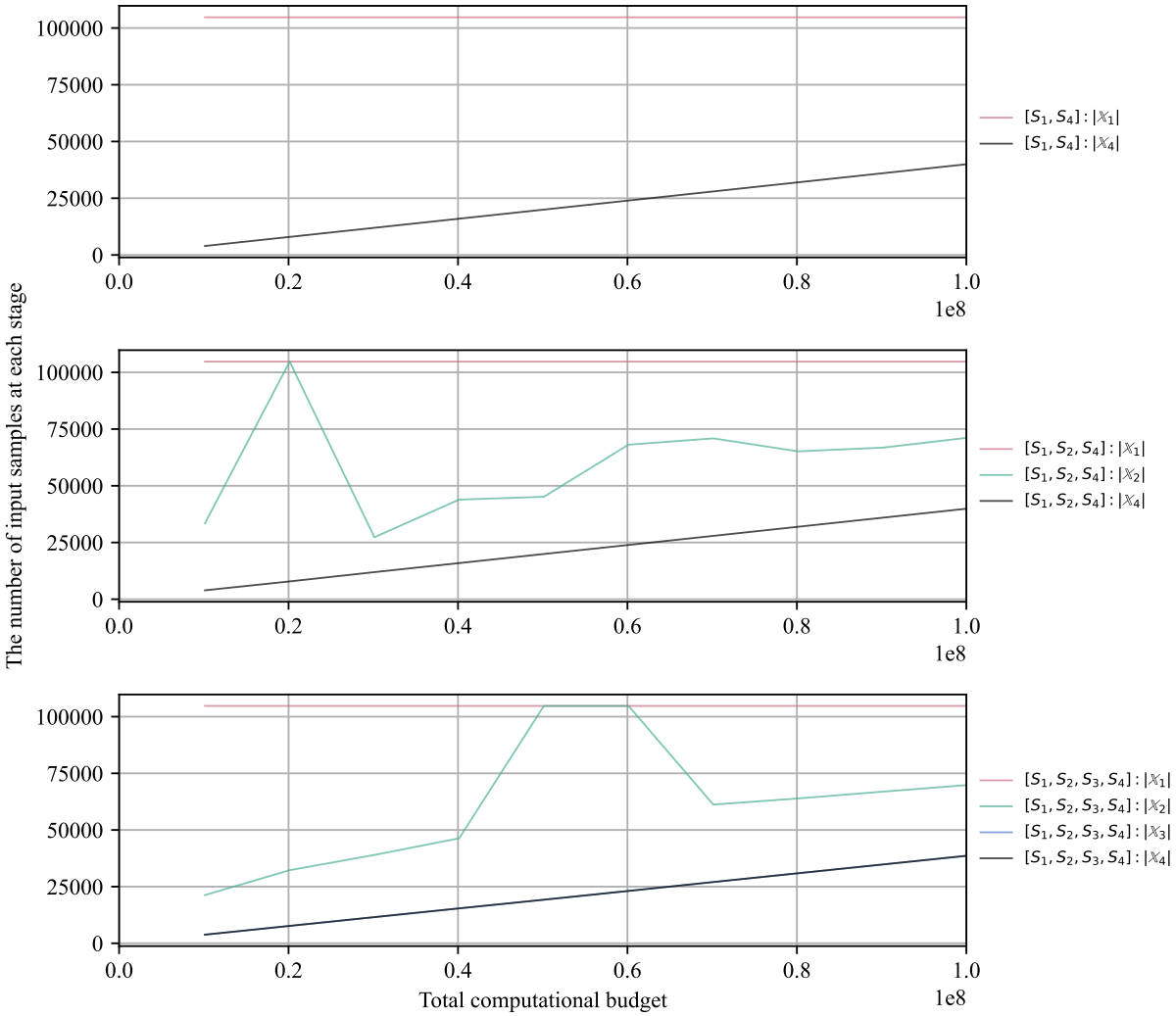


Figure E.3: The number of input samples at each stage for lncRNA detection.

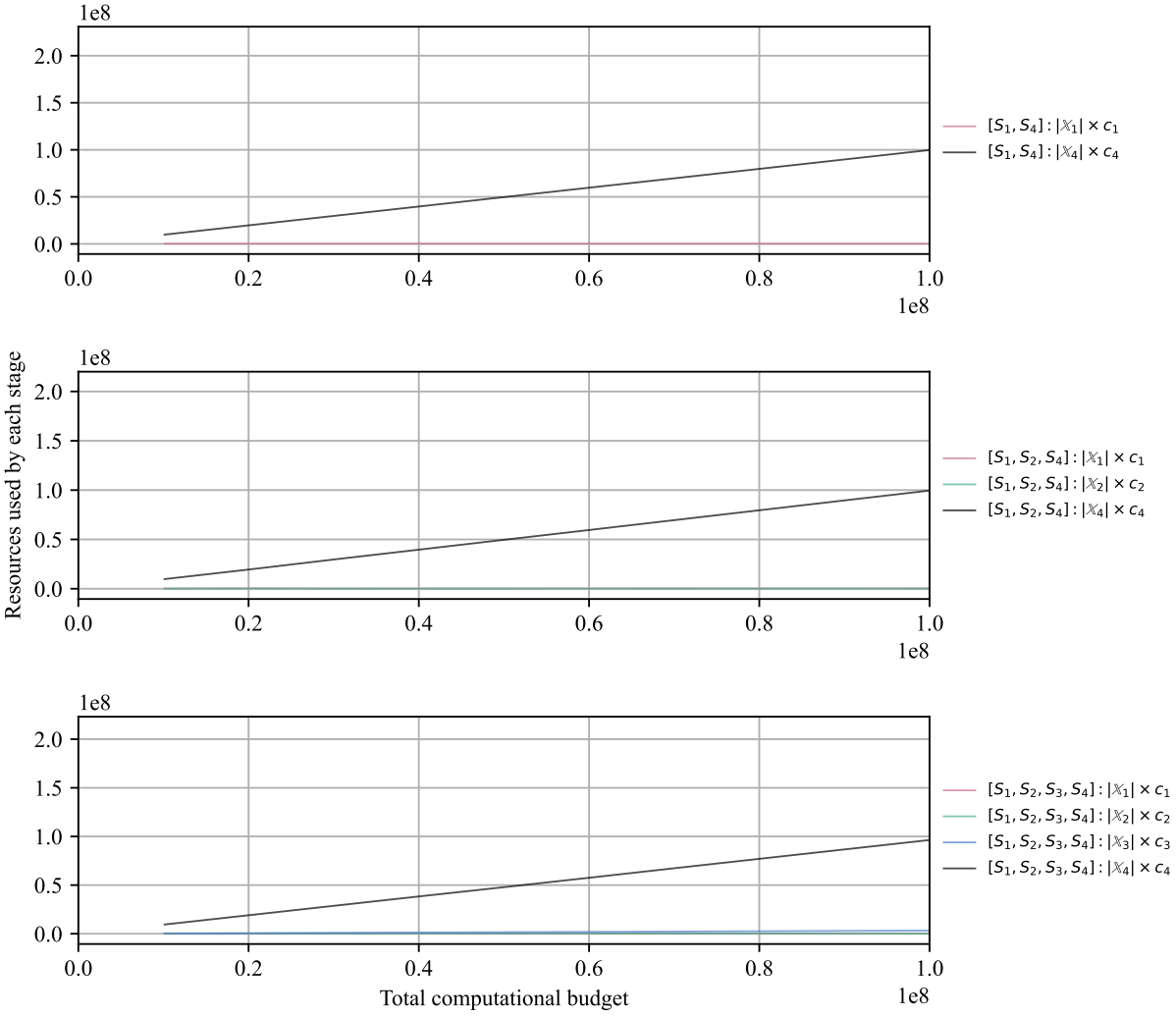


Figure E.4: Resources used by each stage for lncRNA detection.

APPENDIX F

REDOX POTENTIAL COMPUTATION

To compute redox potential (RP) of the materials, we first calculate density functional theory (DFT) using Schrödinger Jaguar [137], with PBE0 [138] functional and $6-31+G(d,p)$ basis set [139]. After geometry optimization using DFT, we compute the electronic features, such as HOMO, LUMO, HOMO-LUMO gap, and RP.

Then, we used the thermodynamic cycle suggested by Truhlar [140, 141] to calculate the RP. To evaluate the reduction free energies at $298K$ in the gas phase $\Delta G^{\text{red}}_{\text{gas}}$, the vibrational frequencies were analyzed for both the anionic and neutral states for all the organic species. To evaluate the solvation-free energies of the anionic and neutral states ($\Delta G_{\text{sol}}\left(R^{-}\right)$ and $\Delta G_{\text{sol}}\left(R\right)$, respectively) in the mixture of ethylene carbonate and dimethyl carbonate, the Poisson–Boltzmann implicit solvation model was used with a dielectric constant of 16.14. Using the thermodynamic cycle, the reduction free energy in solution phase ($\Delta G^{\text{red}}_{\text{sol}}\left(R\right)$) was calculated by:

$$\Delta G^{\text{red}}_{\text{sol}}\left(R\right) = \Delta G^{\text{red}}_{\text{gas}}\left(R\right) + \Delta G_{\text{sol}}\left(R^{-}\right) - \Delta G_{\text{sol}}\left(R\right). \tag{F.1}$$

Finally, the RP in solution phase with respect to Li/Li^{+} electrode was calculated based on the free energy change for reduction in solution phase using,

$$E^{0}_{\text{w.r.t. Li}} = \left(-\frac{\Delta G^{\text{red}}_{\text{sol}}\left(R\right)}{nF} + E_{H}\right) - E_{\text{Li}}, \tag{F.2}$$

where n and F denote the number of electrons transferred and the Faraday constant ($96,485$ C mol^{-1}), respectively. E_{H} and E_{Li} correspond to the absolute potential of the hydrogen electrode ($4.44V$), and the potential of Li electrode with respect to the standard hydrogen electrode ($-3.05V$) [142], respectively. In the previous studies, we showed that this computational strategy produced RPs with a staggering accuracy, within $0.3V$ vs. Li/Li^{+} relative to experimental results [95, 92, 91, 93, 96, 99, 94, 97, 98]. In addition to the RP, the adiabatic electron affinity was

calculated from the difference in energy between the organic molecules in their neutral state and in their anionic state. Additional details of the DFT calculations used to predict the RP are found in the previous studies [95, 92, 91, 93, 96, 99, 94, 97, 98].

APPENDIX G

PERFORMANCE COMPARISON OF VARIOUS ML SURROGATE MODELS IN PREDICTING ELECTRONIC PROPERTIES

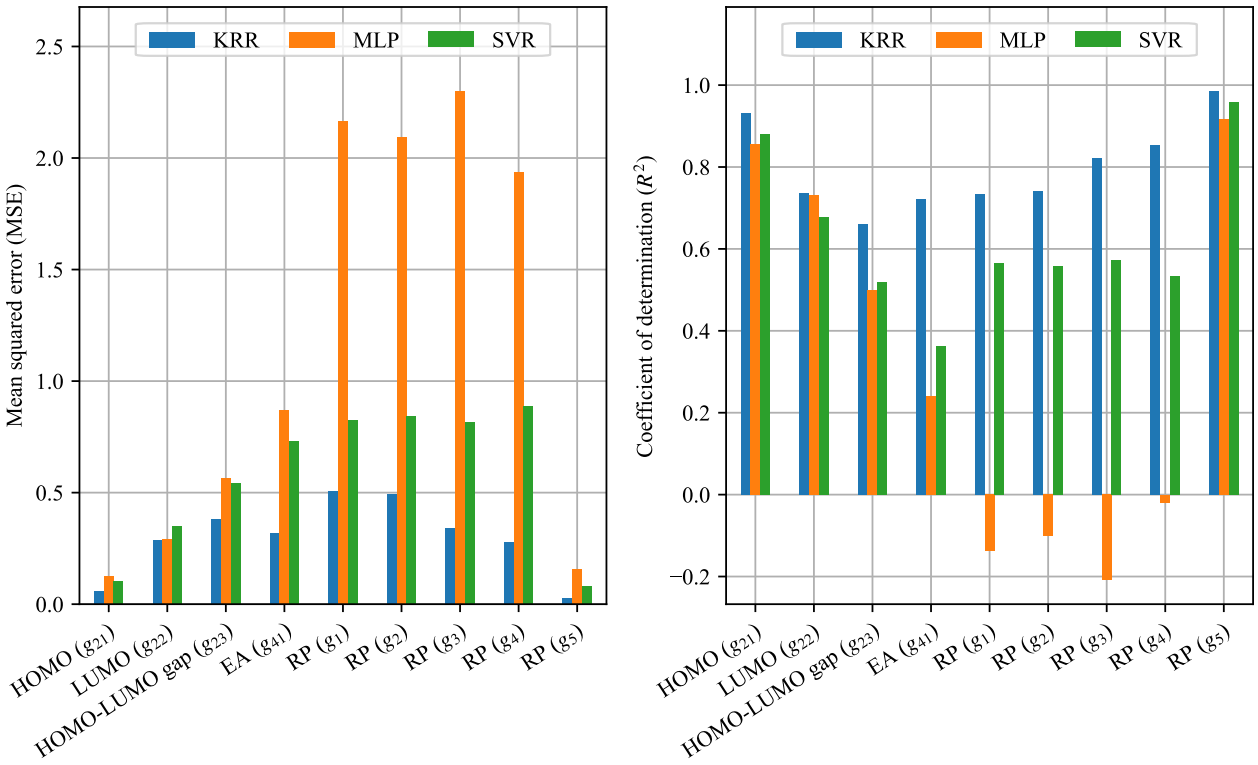


Figure G.1: Performance comparison of fundamental machine learning (ML) models in terms of mean squared error (MSE) and coefficient of determination. For the kernel ridge regression (KRR) model, we used the hyperparameters summarized in Table H.1. For multilayer perceptron (MLP) and support vector regression (SVR) models, we optimized hyperparameters based on a 5-fold cross-validation.

APPENDIX H

SPECIFICATION OF THE OPTIMIZED ML SURROGATE MODELS OF THE HTVS PIPELINE

Surrogate symbol	Machine learning model	Kernel	Hyperparameter (α)	Mean squared error	R^2 score
g_1	Kernel ridge regression	Radial basis function	0.1	0.5046	0.7346
g_2			0.1	0.4907	0.7419
g_3			0.1	0.3408	0.8208
g_4			0.1	0.2781	0.8538
g_5			0.1	0.0256	0.9865
$g_{2,1}$	Kernel ridge regression	Radial basis function	0.1	0.0595	0.9307
$g_{2,2}$			0.1	0.2870	0.7351
$g_{2,3}$			0.1	0.3808	0.6616
$g_{4,1}$			0.1	0.3194	0.7212

Table H.1: Specification of the optimized machine learning (ML) surrogate models utilized to construct the HTVS pipeline. The hyperparameters–kernel function and α–were optimized via 5-fold cross-validation.

APPENDIX I

PERFORMANCE EVALUATION OF THE OPTIMIZED HTVS PIPELINE BASED ON A STRICT 5-FOLD CROSS-VALIDATION

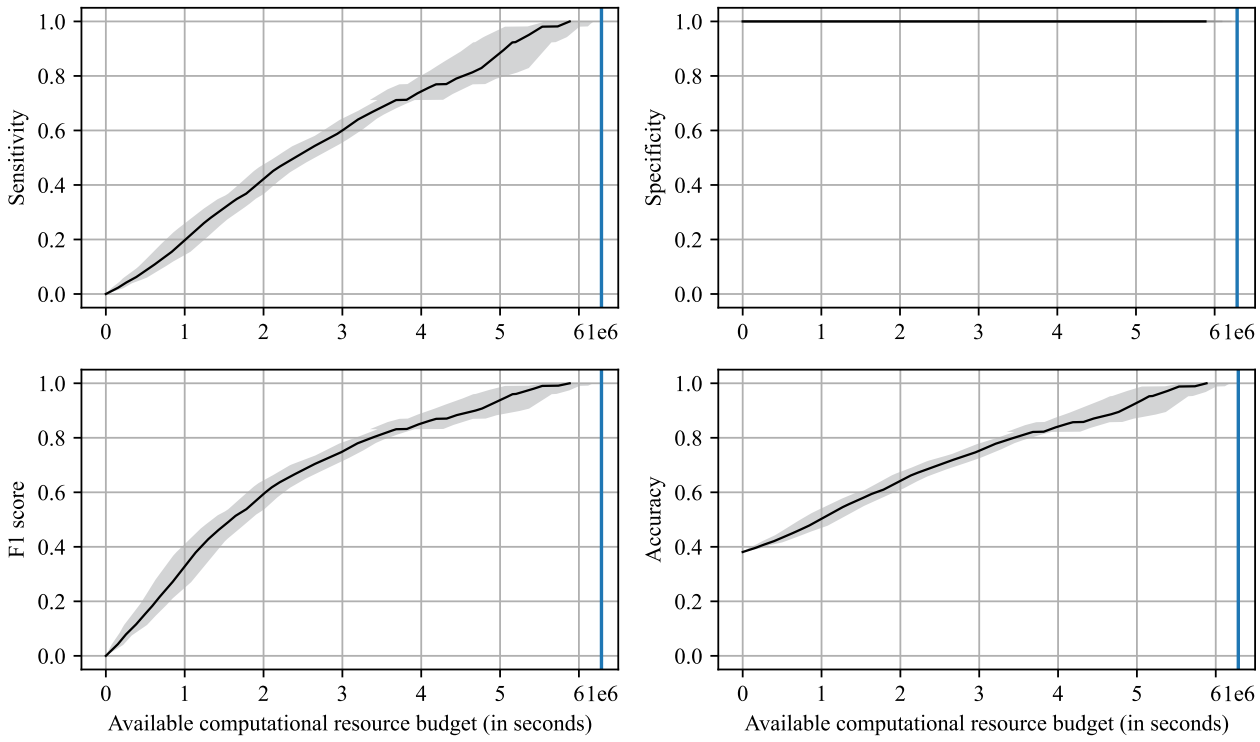


Figure I.1: Performance evaluation of the optimized HTVS pipeline with minimum target redox potential (RP) 2.5 V under a computational resource budget constraint (x-axis) based on a strict 5-fold cross-validation.

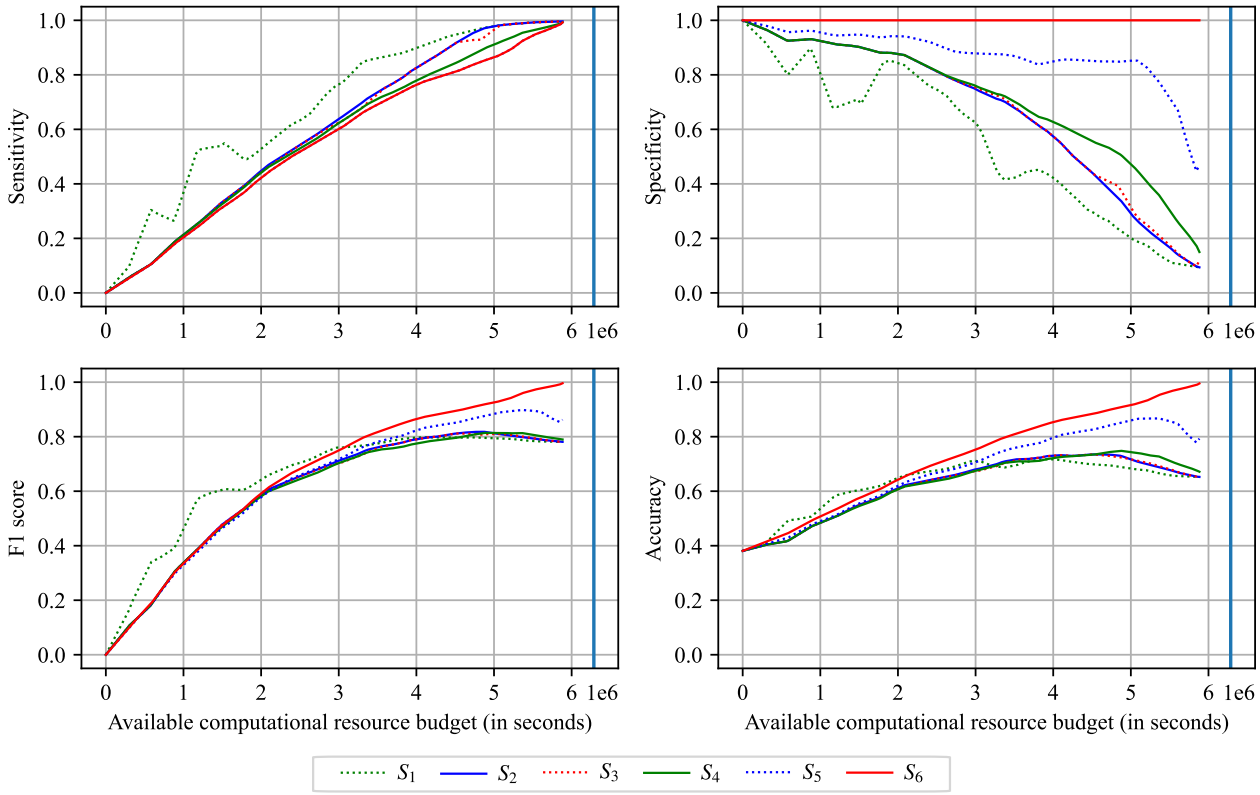


Figure I.2: Performance evaluation at each stage in the optimized HTVS pipeline with minimum target RP 2.5 V under a computational resource budget constraint (x-axis) based on a strict 5-fold cross-validation.

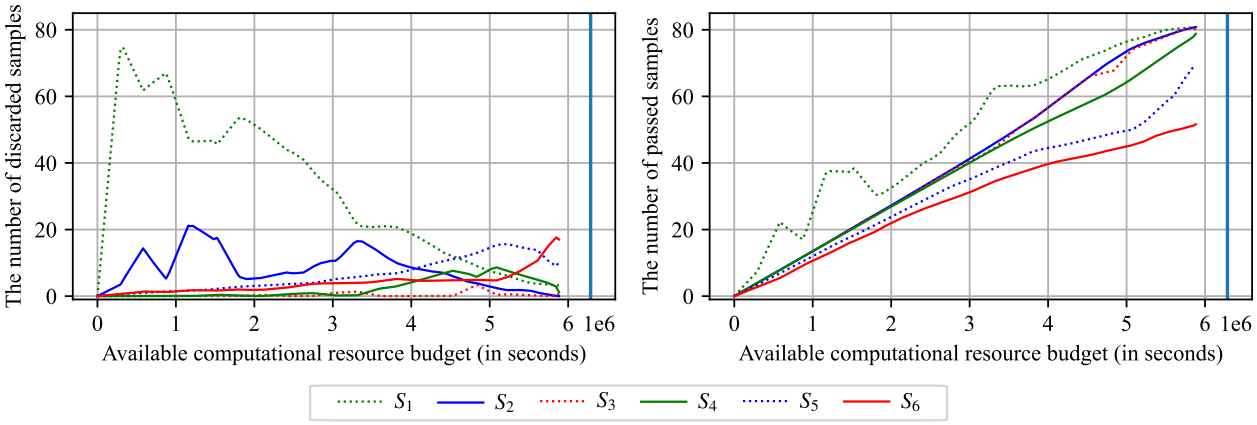


Figure I.3: The number of samples discarded (left) or passed to the next stage (right) at each stage in the HTVS pipeline with minimum target RP 2.5 V under a computational resource budget constraint (x-axis) based on a strict 5-fold cross-validation.

α	Selected materials	Total cost (seconds)	Effective cost (seconds)	Sensitivity	Specificity	F1 score	Accuracy
0.25	36.6	$3,506,408.8$	$95,803.5$	0.7038	1	0.8233	0.8167
0.5	43.8	$4,425,191.2$	$101,031.8$	0.8423	1	0.9134	0.9024
0.75	45.4	$4,605,138.8$	$101,434.8$	0.8731	1	0.9316	0.9214

Table I.1: Performance evaluation of the jointly optimized HTVS pipeline with minimum target RP 2.5 V based on a strict 5-fold cross-validation.

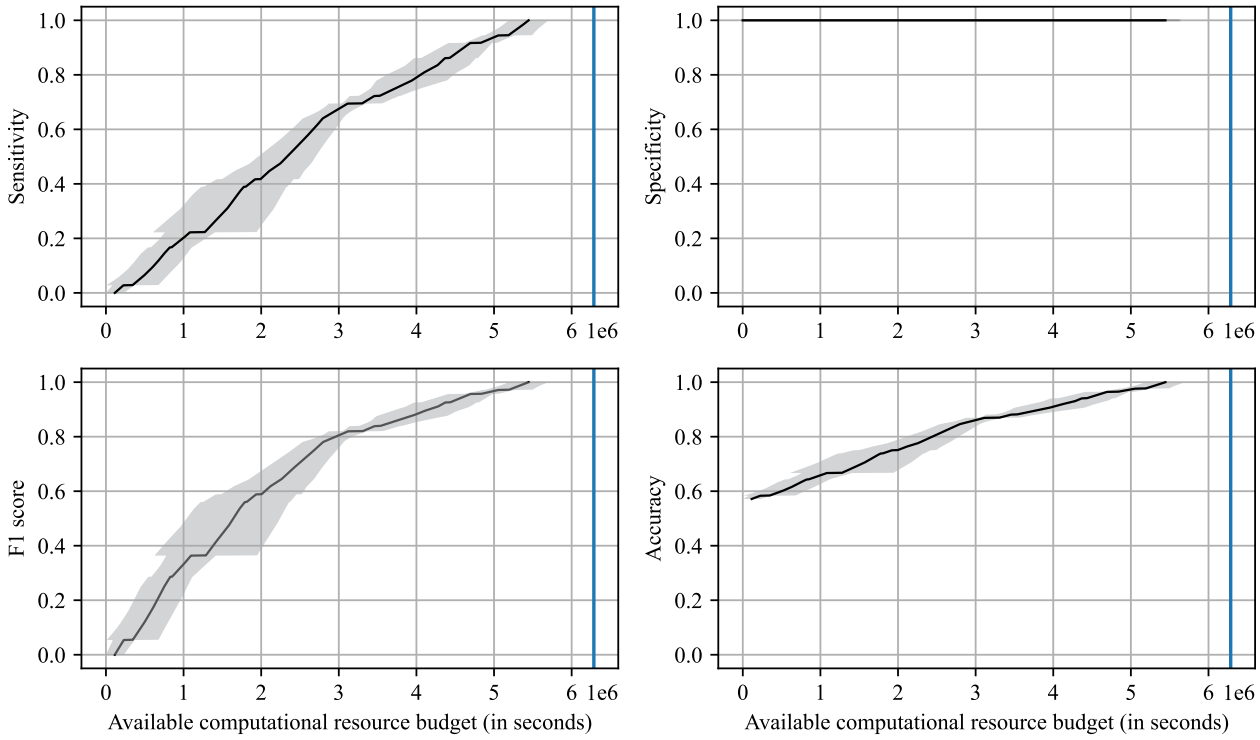


Figure I.4: Performance evaluation of the optimized HTVS pipeline with target RP range $[2.5\ \text{V}, 3.2\ \text{V}]$ under a computational resource budget constraint (x-axis) based on a strict 5-fold cross-validation.

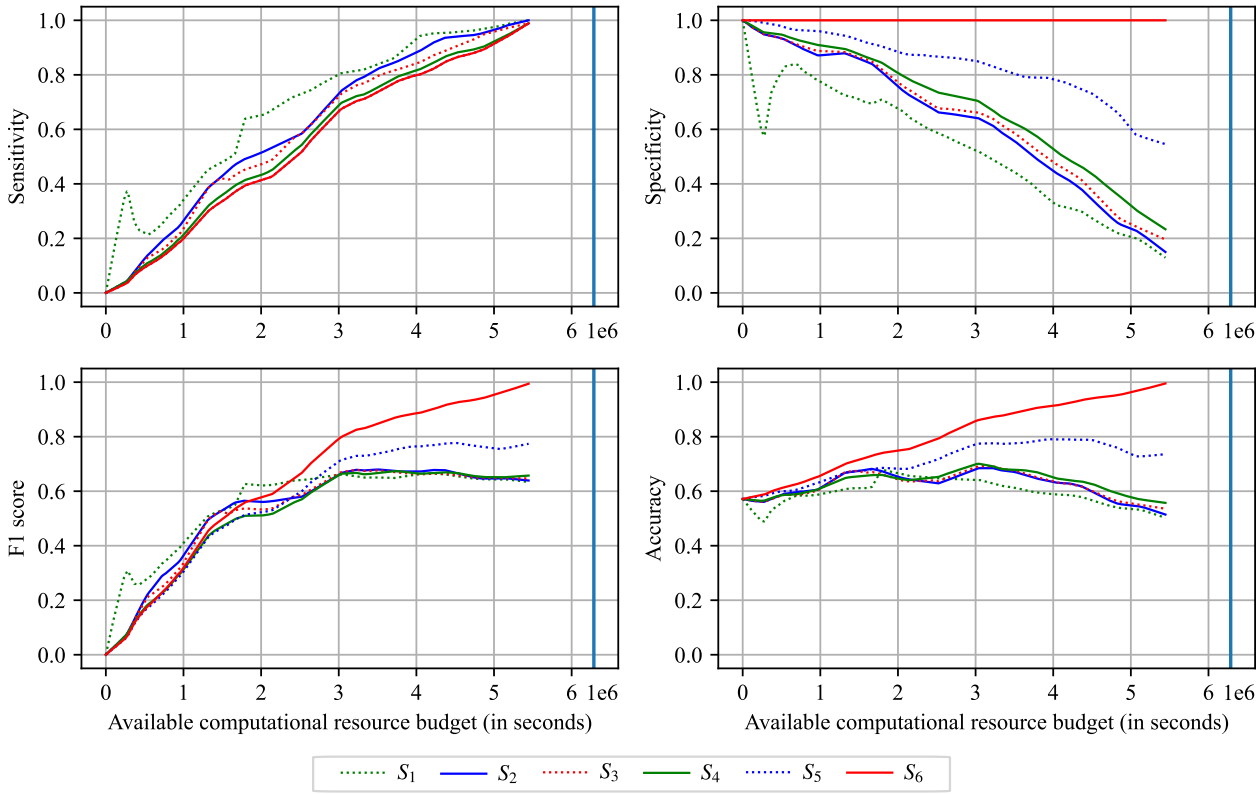


Figure I.5: Performance evaluation of each stage in the optimized HTVS pipeline with target RP range $[2.5\ \text{V}, 3.2\ \text{V}]$ under a computational resource budget constraint (x-axis) based on a strict 5-fold cross-validation.

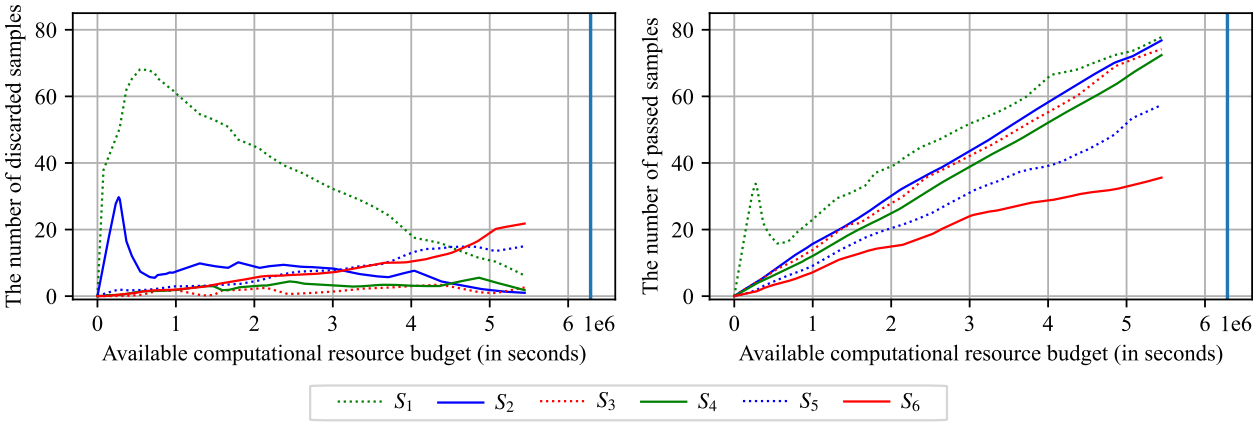


Figure I.6: The number of samples discarded (left) or passed to the next stage (right) at each stage in the HTVS pipeline with target RP range $[2.5\text{ V}, 3.2\text{ V}]$ under a computational resource budget constraint (x-axis) based on a strict 5-fold cross-validation.

α	Selected materials	Total cost (seconds)	Effective cost (seconds)	Sensitivity	Specificity	F1	Accuracy
0.25	25	$3,026,165.6$	$121,046.6$	0.6944	1	0.8155	0.8690
0.5	31.6	$4,490,452.4$	$142,102.9$	0.8778	1	0.9336	0.9476
0.75	33.2	$5,112,623.8$	$153,994.7$	0.9222	1	0.9576	0.9667

Table I.2: Performance evaluation of the jointly optimized HTVS pipeline with target RP range $[2.5\text{ V}, 3.2\text{ V}]$ based on a strict 5-fold cross-validation.

APPENDIX J

PERFORMANCE EVALUATION OF THE OPTIMIZED HTVS PIPELINE WITH STRUCTURE $[S_2, S_4, S_5, S_6]$

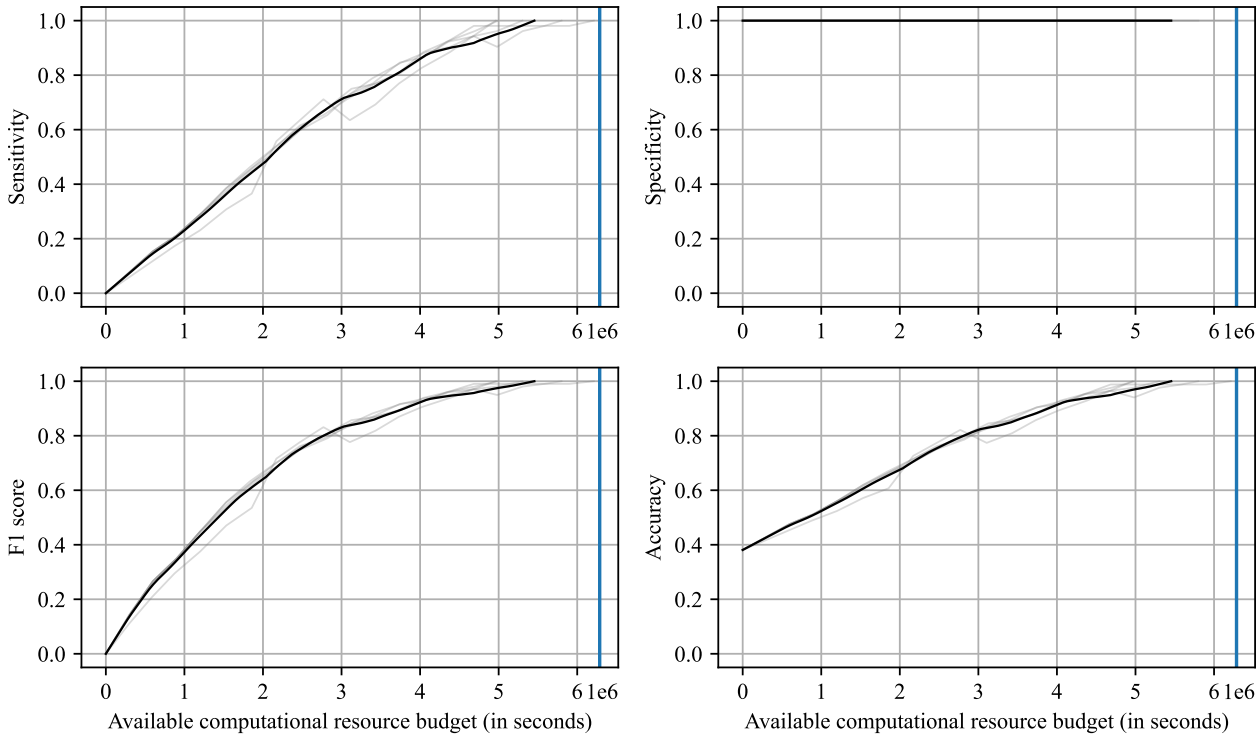


Figure J.1: Performance evaluation of the optimized high-throughput virtual screening (HTVS) pipeline $[S_2, S_4, S_5, S_6]$ with minimum target redox potential (RP) 2.5 V under a computational resource budget constraint (x-axis) based on a 5-fold cross-validation.

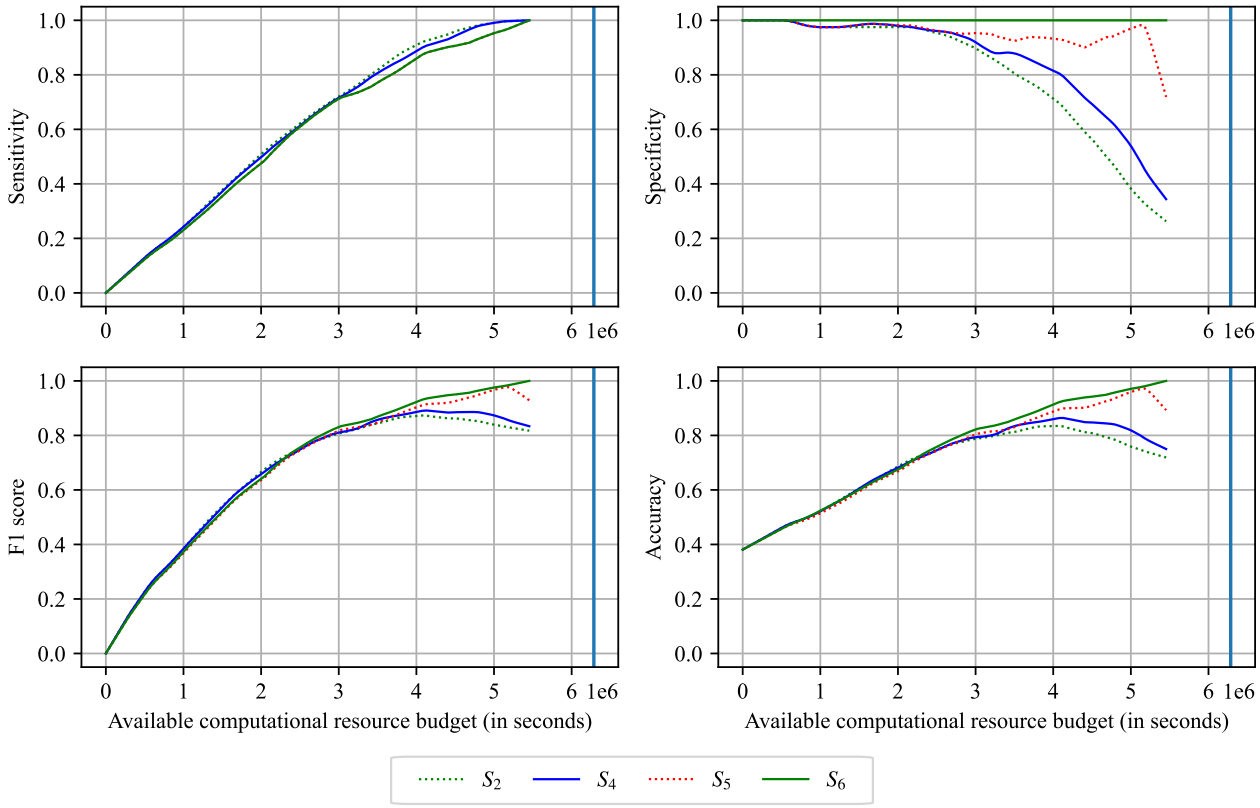


Figure J.2: Performance evaluation of each stage in the optimized HTVS pipeline $[S_2, S_4, S_5, S_6]$ with minimum target RP 2.5 V under a computational resource budget constraint (x-axis) based on a 5-fold cross-validation.

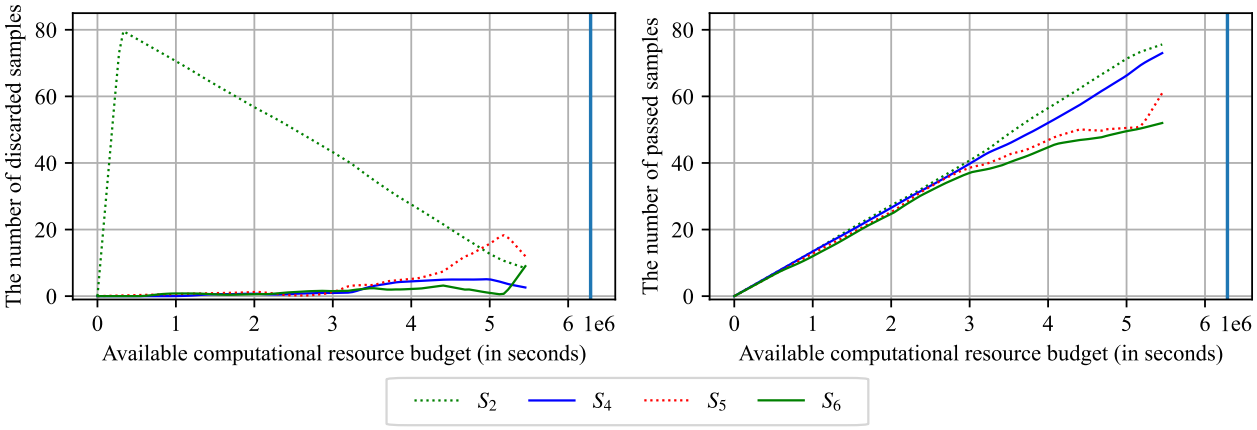


Figure J.3: The number of samples discarded (left) or passed to the next stage (right) at each stage in the HTVS pipeline $[S_2, S_4, S_5, S_6]$ with minimum target RP 2.5 V under a computational resource budget constraint (x-axis) based on a 5-fold cross-validation.

α	Selected materials	Total cost (seconds)	Effective cost (seconds)	Sensitivity	Specificity	F1 score	Accuracy
0.25	21.2	$1,697,310.6$	$80,061.8$	0.4077	1	0.5562	0.6333
0.5	46.6	$4,211,703.2$	$90,379.9$	0.8962	1	0.9443	0.9357
0.75	48.2	$4,540,007.4$	$9,419.2$	0.9269	1	0.9616	0.9548

Table J.1: Performance evaluation of the jointly optimized HTVS pipeline $[S_2, S_4, S_5, S_6]$ with minimum target RP 2.5 V based on a 5-fold cross-validation.

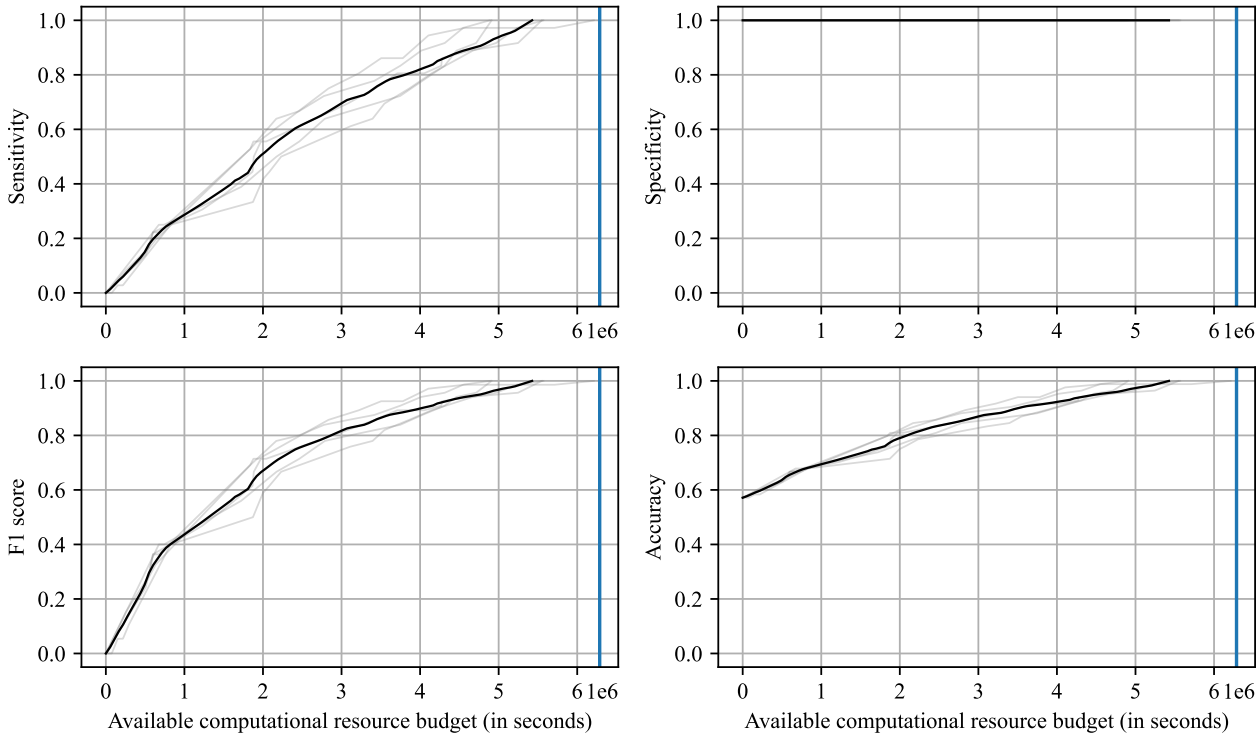


Figure J.4: Performance evaluation of the optimized HTVS pipeline $[S_2, S_4, S_5, S_6]$ with target RP range $[2.5\ \text{V}, 3.2\ \text{V}]$ under a computational resource budget constraint (x-axis) based on a 5-fold cross-validation.

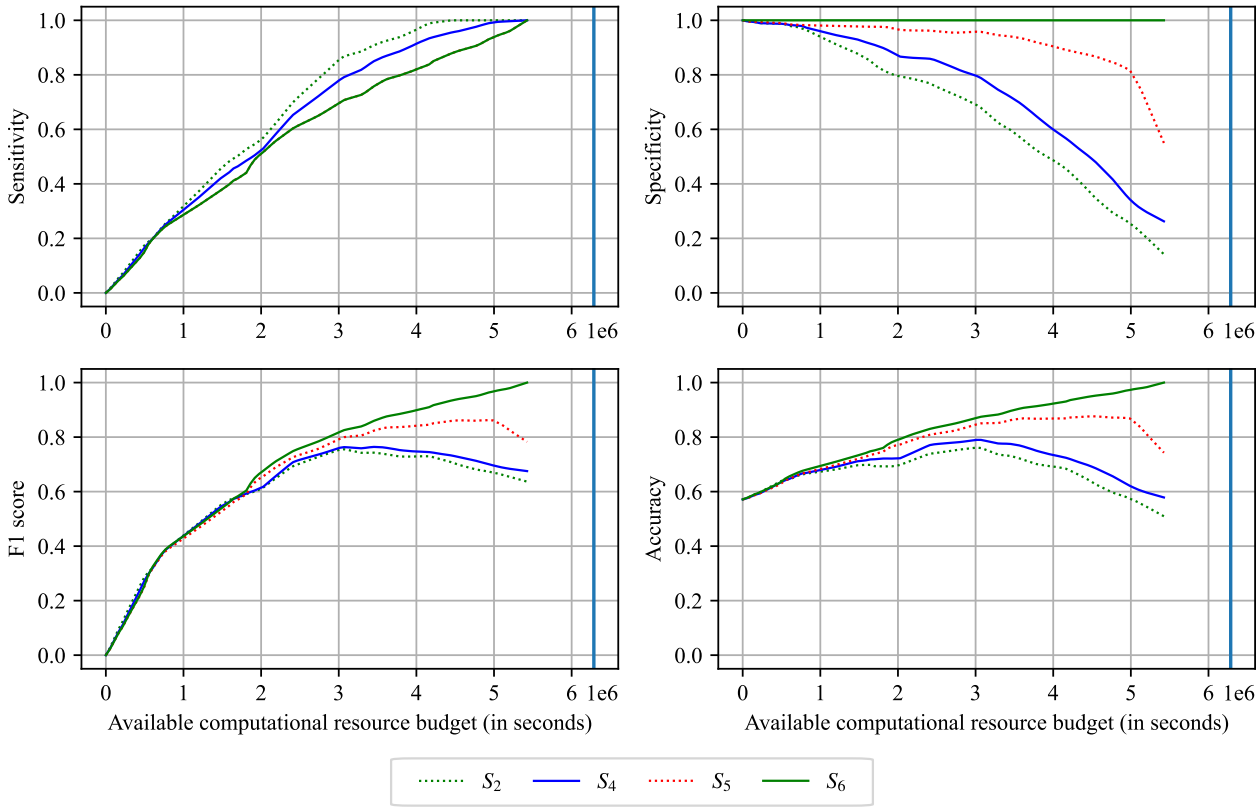


Figure J.5: Performance evaluation of each stage in the optimized HTVS pipeline $[S_2, S_4, S_5, S_6]$ with target RP range $[2.5\ \text{V}, 3.2\ \text{V}]$ under a computational resource budget constraint (x-axis) based on a 5-fold cross-validation.

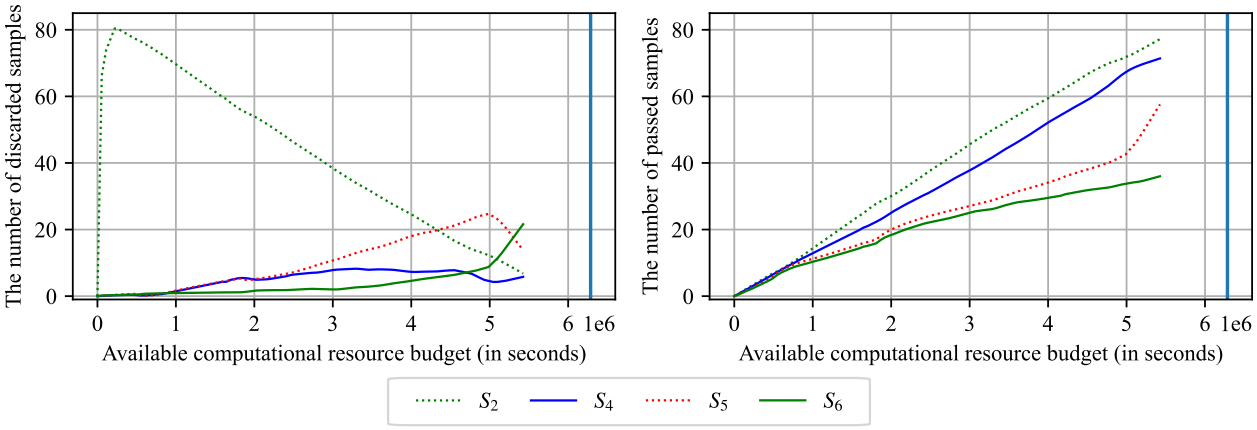


Figure J.6: The number of samples discarded (left) or passed to the next stage (right) at each stage in the HTVS pipeline $[S_2, S_4, S_5, S_6]$ with target RP range $[2.5\ \text{V}, 3.2\ \text{V}]$ under a computational resource budget constraint (x-axis) based on a 5-fold cross-validation.

α	Selected materials	Total cost (seconds)	Effective cost (seconds)	Sensitivity	Specificity	F1	Accuracy
0.25	12.2	$1,350,211.2$	$110,673$	0.3389	1	0.4732	0.7167
0.5	30.6	$3,645,767.6$	$119,142.7$	0.85	1	0.9054	0.9357
0.75	31.6	$4,307,546.6$	$136,314.8$	0.8778	1	0.9303	0.9476

Table J.2: Performance evaluation of the jointly optimized HTVS pipeline $[S_2, S_4, S_5, S_6]$ with target RP range $[2.5\ \text{V}, 3.2\ \text{V}]$ based on a 5-fold cross-validation.

APPENDIX K

PERFORMANCE EVALUATION OF THE OPTIMIZED HTVS PIPELINE WITH MINIMUM TARGET REDOX POTENTIAL 4.3 V

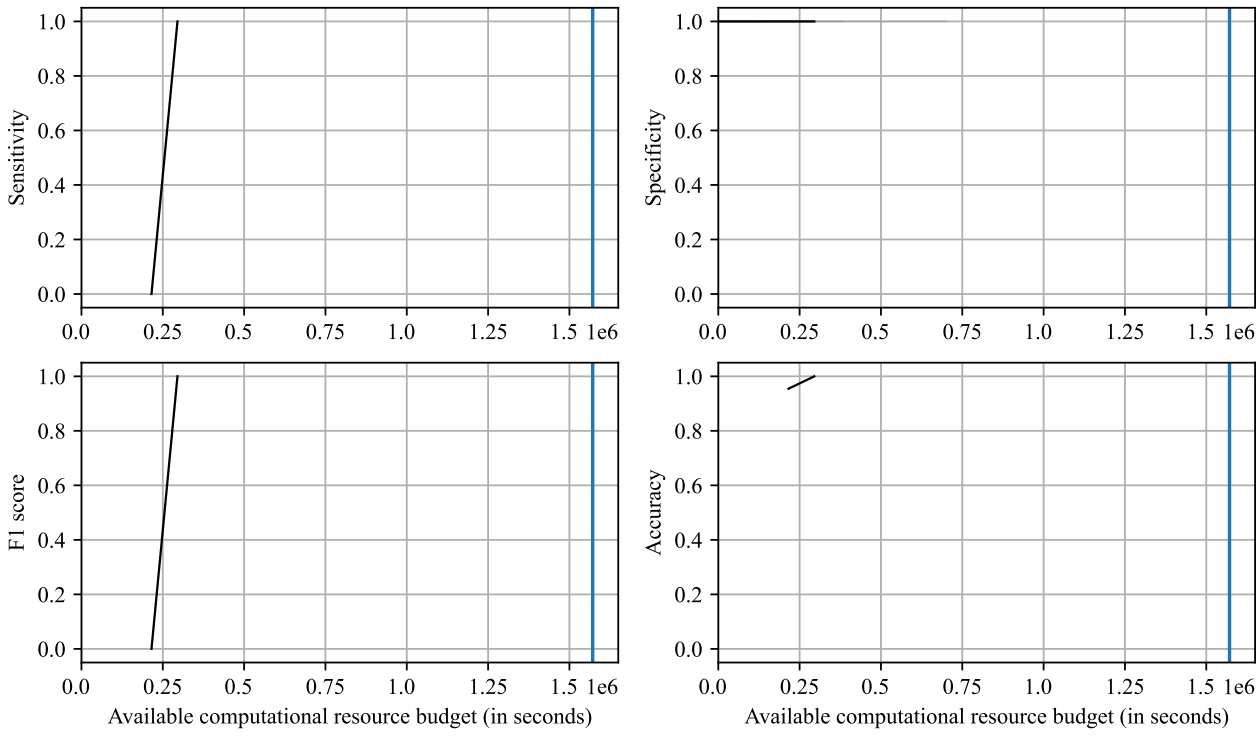


Figure K.1: Performance evaluation of the optimized high-throughput virtual screening (HTVS) pipeline with minimum target redox potential (RP) 4.3 V under a computational resource budget constraint (x-axis) based on a 5-fold cross-validation.

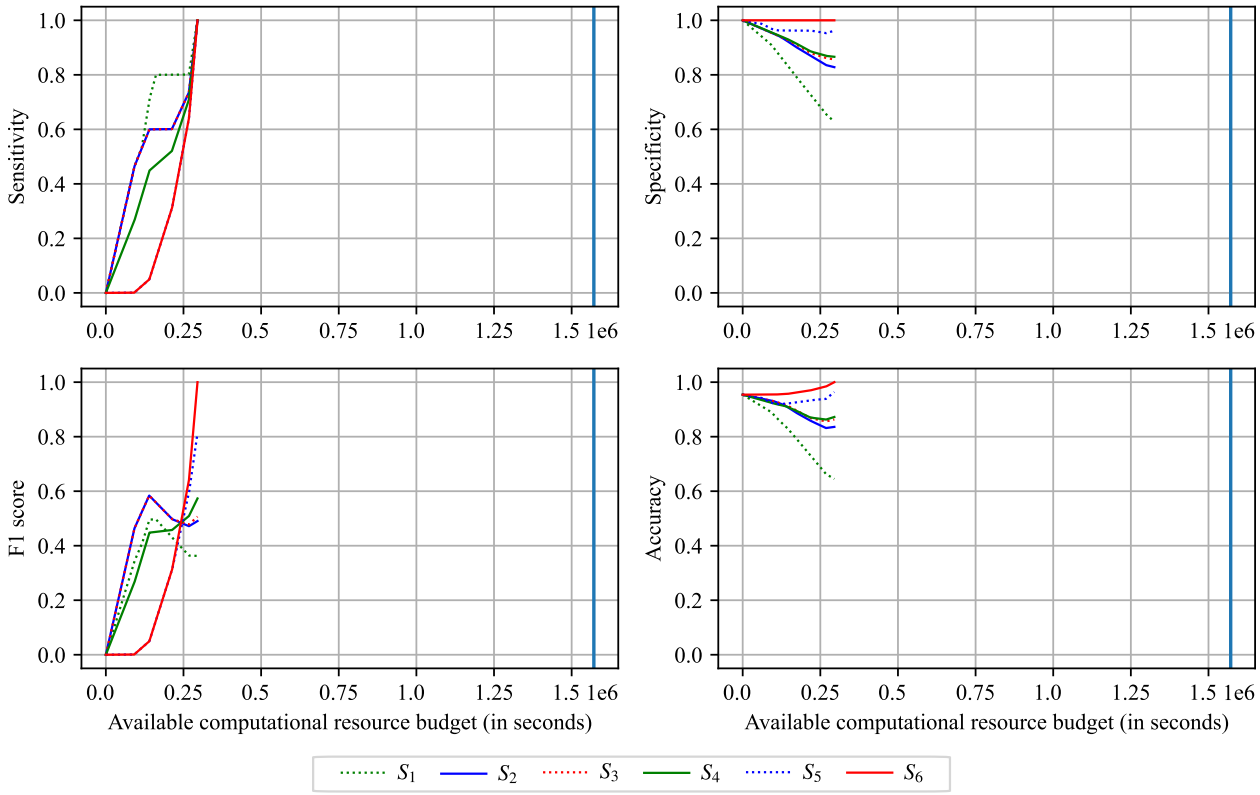


Figure K.2: Performance evaluation at each stage in the optimized HTVS pipeline with minimum target RP 4.3 V under a computational resource budget constraint (x-axis) based on a 5-fold cross-validation.

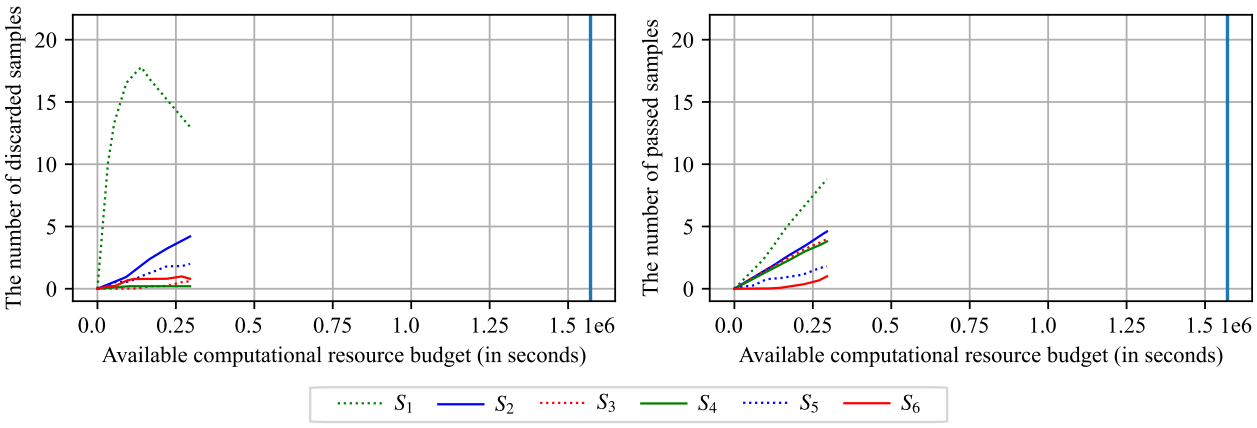


Figure K.3: The number of samples discarded (left) or passed to the next stage (right) at each stage in the HTVS pipeline with minimum target RP 4.3 V under a computational resource budget constraint (x-axis) based on a 5-fold cross-validation.

APPENDIX L

SOFTWARE AVAILABILITY

Chapter	GitHub Link
2	https://github.com/bjyoontamu/Kuramoto-Model-OED-acceleration
3	https://github.com/bjyoontamu/OCC
4	https://github.com/bjyoontamu/occ-rp

Table L.1: List of software developed in this dissertation.