APPLICATIONS OF AUTONOMOUS DATA COLLECTION AND ACTIVE LEARNING

Decision-Making Under Uncertainty for Multi-stage Pipelines: Simulation Studies to Benchmark Screening Strategies

KRISTOFER G. REYES [1,4] JIAQIAN LIU,[2]
and CARLOS JUAN DÍAZ VARGAS[3]

1.—Department of Materials Design and Innovation, University at Buffalo, Buffalo, NY 14260, USA. 2.—Department of Computer Science, University at Buffalo, Buffalo, NY 14260, USA. 3.—Department of Computer Science, University of Puerto Rico, San Juan, PR 00925, USA. 4.—e-mail: kreyes3@buffalo.edu

Multi-stage screening pipelines are ubiquitous throughout experimental and computational science. Much of the effort in developing screening pipelines focuses on improving generative methods or surrogate models in an attempt to make each screening step effective for a specific application. Little focus has been placed on characterizing a generic screening pipeline's performance with respect to the problem or problem parameters. Here, we develop methods to codify and simulate features and properties about the screening procedure in general. We outline and model common problem settings and identify potential opportunities to perform decision-making under uncertainty for optimizing the execution of screening pipelines. We then illustrate the developed methods through several simulation studies. We finally show how such studies can provide a quantification of the screening pipeline performance with respect to problem parameters, specifically identifying the significance of stage-wise covariance structure. We show how such structure can lead to qualitatively different screening behaviors and how screening can even perform worse than random in some cases.

INTRODUCTION

Multi-stage screening pipelines are ubiquitous throughout experimental and computational science. In such a pipeline, a set of initial candidates is passed through several evaluative stages. Based on the collective performance of the candidates at any particular stage of the pipeline, a subset of candidates is selected to advance to the next stage; the rest are "screened out." Typically, this screening is done to minimize how many candidates are evaluated throughout the pipeline, with the goal of identifying promising or optimal candidates with respect to some experimental objective. Often, later stages are more costly, in time, money or other measures of effort.

For example, experimental and virtual screenings are used in drug-design,[1–4] chemistry[5] and material science.[6–8] Here, candidate drugs, molecules or materials designs are first tested using preliminary computer models, and promising candidates are then tested in real-world experiments. Another example is multi-modal characterization of materials. In this context, a material sample is characterized using one of several techniques, from atomic force microscopy available in many labs to x-ray scattering at a synchrotron. Here, a materials scientist may employ an ad hoc screening procedure to identify which materials sample candidates are passed through this characterization pipeline. As autonomous materials platforms begin to incorporate more complex experimental or hybrid computational/experimental structure,[9–11] understanding how screening pipelines could operate in an autonomous manner could allow such platforms the ability to make optimal screening pipeline decisions as well.

The use of a screening procedure relies on a heuristic belief that the screening pipeline will effectively narrow down a search space to a small set of promising leads. However, there is no generic guarantee this will work. Consider, for example, the case of identifying promising drugs, where the "gold standard" measure for effectiveness occurs at the last stage of a screening pipeline, which could be performing physical trials of the drug. Imagine the first stage is a machine learning (ML) model meant to predict this effectiveness based on a candidate drug's descriptors, but its predictions are only slightly correlated or even anti-correlated with the gold standard measure. In this case, the first stage of the pipeline may aggressively and prematurely screen out good candidates.

To mitigate against this threat, much of the development around screening pipelines has been the development of stages and models that are more strongly correlated to the experimental objectives, either through better surrogate models[12–15] or through better generative models[16–18] that determine the initial candidate pool to begin with. Yet fundamental and generic questions persist: What aspects of a problem make the screening procedure effective? Can we quantify the effectiveness of the screening procedure? How do we deal with the uncertainties behind the relationships between stages? Are our current pipeline strategies or ad hoc decision-making policies sufficient in executing an effective pipeline? Coming back to the drug-design example above, we may wish to quantify how poorly the screening procedure performs as a function of the anti-correlation between the first and final stages.

To answer such questions, we must develop a theory where the general pipeline itself is the subject of study, specifically from a perspective of uncertainty quantification and decision-making under uncertainty. In this article, we outline a formalization of a family of screening pipeline problems and simulation-based policies that we use to make decisions before and during the screening procedure. To accomplish this, we build a model of statistical structure between candidates and stages and offer algorithms for efficient simulation-based decision-making based on the assumed statistical structure model. Lastly, we present some simulation results to illustrate how such models, simulations and policies can be used to characterize and optimize screening pipelines.

PROBLEM DEFINITIONS

We consider a screening procedure with m initial sampled candidates, $x_1, x_2, ..., x_m$. These candidates can represent drugs, molecules, materials or even synthesis recipes. Often, these initial candidates are obtained via generative models, ensemble methods or evolutionary techniques—methods capable of proposing a batch of candidates to test.

We will also consider n stages $f_1, f_2, .., f_n$, which we shall view as functions mapping candidates x_i to corresponding scalar quantities of interest $Y_{i,j} = f_j(x_i)$ as measured by the j-th stage. For example, in drug design, the stages could represent molecular property predictors, docking score calculations, density functional theory or molecular dynamics simulations, or physical trials. The outcomes for these stages could be computed binding efficacy, free energy calculations or measures toxicity and synthesizability. In manufacturing, the stages could include low- and high-fidelity finite element simulations, rapid physical prototyping via 3D printers and fabrication of a final deliverable. The outcomes for these stages could be measures of mechanical toughness or resiliency.

In the abstract, we call these values *scores*, so that $Y_{i,j}$ is the score assigned to candidate x_i at stage j. We interpret these scores as the averaged ground truth values. That is, if we were to repeatedly evaluate a candidate x_i through the j-th stage, the average of the set of obtained scores will approach the number $Y_{i,j}$. Thus, the $Y_{i,j}$ represents the score of candidate x_i through the j-th stage, averaging over noise or any inherent randomness implicit in making the appropriate measurement and obtaining the quantity of interest.

Throughout, we assume we do not know the scores prior to evaluating the candidates through each stage. As *a priori* uncertain quantities, we treat the scores within a *Bayesian* context by modeling such scores using a multivariate probability distribution. That is, throughout the screening procedure, we can maintain a probability distribution for the scores $Y_{i,j}$ that captures predicted estimates of scores not yet evaluated as well as an uncertainty quantification of those predictions and an estimated correlation between different scores. As more data are obtained through the screening procedure, we can update this probability distribution through Bayes's law.

Objectives

These scores allow us to define two objectives. First is stage-wise multi-fidelity optimization, in which we view each stage as measuring the same quantity, but with variable accuracies (i.e., fidelities) and costs. In general, we will assume that the last stage f_n is the ground truth or gold standard that we ultimately care about and is the most expensive. The prior stages exist to perform cheaper screening of candidates. For example, f_n could be performing a real experiment, while prior stages could be a sequence of increasingly accurate (and increasingly costly) simulations. A common pipeline in this setting consists of three stages: f_1 is an ML model, f_2 is a physics-based simulation, while f_3 is a physical experiment. Each stage could produce as an output some figure of merit, such as some material property to optimize, and the goal is to

find the candidate that optimizes this property in the real-life experiment. Another similar setting is when stages correspond to different materials characterization methods, where the final such method could be a costly trip to a beamline. The stage-wise multi-fidelity optimization problem shall be the main focus of this article.

Another objective is stage-wise multi-objective optimization. In this setting, we view each stage as characterizing different quantities of interest. For any given candidate x_i, the scores $Y_{i,1}, Y_{i,2}, ..., Y_{i,n}$ encode a set of different characterizations of candidate x_i's material properties or structure, for example. Under the multi-objective objective, we want to identify those candidates x that either (1) optimize some scalar utility $u(x) = u(f_1(x), f_2(x), ..., f_n(x))$ or (2) lie on the n-dimensional Pareto frontier.[19]

Both objectives can be characterized by defining the *final reward* obtained at the end of the screening procedure. For any screening procedure, we obtain a subset $\mathcal{X}_S \subseteq \{x_i\}_{i=1}^m$ of candidates that "survived" the screening process. For the multi-fidelity objective, we may define the reward as the maximum of the final scores for all such screened candidates, $R = \max_{\mathcal{X}_S} f_m(x)$. For multi-objective optimization, the final score could be either the maximum scalar utility over all surviving candidates, $R = \max_{\mathcal{X}_S} u(x)$, or some measure of improvement to the Pareto front, such as a hypervolume improvement,[20] $R = H(\mathcal{X}_S) - H_0$. Here, $H(\mathcal{X}_S)$ and H_0 are respectively the hypervolumes defined by the Pareto front generated by the surviving points $\mathcal{X}_s$ and prior data points and the Pareto front obtained from just the prior data points.

Pipeline Policies for the Optimal Allocation Design Problem

Regardless of how we define the reward, the goal in executing the pipeline is to do so in a way that maximizes the realized reward on average. Throughout, we shall refer to how we execute the pipeline as a *pipeline policy*. The specific decisions we make during the execution of the pipeline will ultimately impact the rewards we gain at the end of the pipeline. Examples of such decisions include: (1) how many or what proportion of candidates may pass from one stage to the next, (2) how costly each stage is (to the extent that this cost is possible to tune) and (3) how many or which initial candidates we generate.

In this article, we shall decompose a pipeline policy into two policies. The *inter-stage policy* selects which candidates pass to the next stage based on the scores they receive at the current stage. In contrast, the pipeline *meta-policy* selects hyperparameters and constraints that the inter-stage policy must satisfy. The main example used throughout this article is selecting an *allocation* $\mathbf{m} = (m_1, ..., m_n)$ via a meta-policy. An allocation specifies how many candidates are evaluated at each stage. In this notation, $m_1 = m$ candidates are evaluated in the first stage, and then m_2 candidates are selected (by the inter-stage policy) for the second stage, and so on. Throughout, we require each stage to screen out at least one candidate and that some candidates survive to the end. That is, we require $m_1 = m > m_2 > \cdots > m_n \geq 1$.

The meta-policy selects allocations, while the inter-stage policy selects candidates given an allocation. This distinction is not fundamental. We can imagine an inter-stage policy that selects both the number of candidates to pass trough to the next stage and which specific candidates do so. However, for this article, we focus on the decomposition of decision variables described above. We have previously studied a similar nested-batch decision-making policy in the case of Bayesian optimization for the co-design of materials and devices[21] and have found the decomposition computationally favorable in practice.

We employ this nested decision policy to solve the optimal allocation problem. Here, we must select the allocation $\mathbf{m}^\star$ that is likely to result in maximal rewards under a specific inter-stage policy. The meta-policy performs this selection prior to the actual execution of the pipeline. Instead, it relies on limited prior knowledge about the scores to design an allocation. We encode this prior knowledge using a probability distribution, a *(Bayesian) prior* on the scores. This prior captures uncertainties about the scores and the assumed statistical relationship (the *covariance*) between them.

To design an allocation using a prior, we will *simulate* the pipeline under some inter-stage policy. That is, we can sample a ground truth $\mathbf{y} = (y_{i,j})$ from the prior and use this sample to execute the screening pipeline. That is, scores are obtained from the sampled ground truth, $f_j(x_i) = y_{i,j}$, perhaps with some noise added in. Using these scores, the inter-policy screens out candidates for the next stage, and so on. Executing the pipeline this way yields a sample of rewards for the given sampled score and allocation, $r(\mathbf{y}, \mathbf{m})$. Repeating the simulations under different samples of ground truth yields a representative sample of rewards. We treat this sample as an empirical estimate of the distribution of the reward $R(\mathbf{m})$ for a given allocation $\mathbf{m}$. Given this distribution for each allocation $\mathbf{m}$, the meta-policy selects an allocation $\mathbf{m}^\star$.

Meta-policies for Closed-loop Versus Single-shot Settings

The distribution on rewards $R(\mathbf{m})$ is a quantification of uncertainty. The meta-policy may consider this uncertainty, especially when in a closed-loop setting. In this setting, the meta-policy selects an allocation $\mathbf{m}_1^\star$ based the reward distribution that the prior distribution induces. We then implement this decision, running the pipeline in real-life subject to

this allocation. Doing so yields observations of the true scores, which we use to update the Bayesian prior. The updated prior induces a new reward distribution, and so the meta-policy can select a second allocation $\mathbf{m}_2^\star$. This procedure is iterated until some total budget is expended. If the meta-policy is effective, by the end of the campaign, it will have identified a near-optimal allocation.

In this closed-loop setting, we can use relevant policies from Bayesian optimization.[22] They acknowledge the uncertainties present in the reward distribution $R(\mathbf{m})$ by balancing between exploration (suggesting random $\mathbf{m}$ or the one corresponding to the distribution $R(\mathbf{m})$ with the most variance) and exploitation (suggesting the $\mathbf{m}$ that maximizes the average, or expected, value of $R(\mathbf{m})$). For example, the upper confidence bound (UCB) policy[23] explicitly encodes this balance by selecting the allocation $\mathbf{m}$ that maximizes the value:

$$A_{\mathrm{UCB}}(\mathbf{m}) = \mathbb{E}[R(\mathbf{m})] + c \cdot \sqrt{\mathrm{Var}[R(\mathbf{m})]},$$

where $\mathbb{E}[R(\mathbf{m})]$ is the expected value of the reward distribution and $\mathrm{Var}[R(\mathbf{m})]$ is the variance of the distribution. The square root of the variance is the standard deviation, which measures the spread of the distribution. The expectation term favors allocations that have a high expected value (exploitation), while the variance term favors allocations with more reward uncertainty (exploration). The parameter c balances the two terms.

Such policies are truly effective if we can execute several iterations of the closed loop. They are limited in the case where a budget reduces the number of times the entire pipeline can be executed to a few iterations. In the extreme one-shot setting, the pipeline can be executed once. In this case, we do not have the luxury to be explorative in our selection of $\mathbf{m}$. Hence, the only reasonable meta-policy in the one-shot setting is pure exploitation (XPLT):

$$A_{\mathrm{XPLT}}(\mathbf{m}) = \mathbb{E}[R(\mathbf{m})].$$

While this limits our choice of the meta-policy, we may still consider different inter-stage policies to guide the single execution of the pipeline. However, for this article, we shall focus explicitly on the one-shot setting and the XPLT meta-policy. However, we note that the development and analysis of proper inter-stage policies specifically for multi-stage pipelines could be a fruitful topic of study.

The Cost-Constrained Problem

In practice, later stages of the pipeline can be significantly more expensive than earlier ones. They represent real experiments, high-fidelity simulations or material characterization at a high-demand user facility. Let $\mathbf{c} = (c_1, c_2, , c_n)$ be a cost vector so that c_j is the cost to evaluate a candidate at stage j.

We assume the costs are additive and that we cannot amortize costs or parallelize evaluations. Therefore, the costs to evaluate m_j candidates in stage j is simply $m_j c_j$, and the total cost $C(\mathbf{m})$ for an allocation is $C(\mathbf{m}) = \mathbf{m} \cdot \mathbf{c}$, which we require to be less than some budget $\mathbf{C}max$.

We frame the XPLT meta-policy via the (linearly) constrained integer program:

$$\underset{m \in \mathbb{Z}^n}{\text{maximize}} \quad A_{\mathrm{XPLT}}(m)$$
$$\text{Subject to}: \quad m_1 = m > m_2 > \ldots m_n \geq 1$$
$$m \cdot c \leq C_{\max}$$

The meta-policy selects the allocation that solves this program. While the function A_{XPLT} is non-linear in $\mathbf{m}$, it is monotone with respect to the dominance partial order on points in $\mathbb{Z}^n$. That is, for two allocations $\mathbf{m} = (m_1, , m_n)$ and $\mathbf{m}' = (m_1', , m_n')$ such that $m_j' \geq m_j$ for all $j = 1, 2, , n$, we have $A_{\mathrm{XPLT}}(\mathbf{m}') \geq A_{\mathrm{XPLT}}(\mathbf{m})$. In other words, it is never optimal to consider fewer candidates than what we have the capacity for. This monotone property means we only have to consider extremal points in the feasible set defined by the linear constraints above to solve the integer program.

A Rich Problem Landscape

Multi-stage screening is a rich and multi-faceted problem that can be studied in various contexts. Above, we outlined a few dimensions of interest. To summarize, we identified two objectives: multi-objective and multi-fidelity optimizations. We must specify what aspects of the pipeline can be optimized or controlled. There are a variety of such pipeline design or decision variables, but above we identified two: the stage-wise allocations and, given a fixed allocation, which candidates to pass between stages. We outlined a few policies in which each selects specific settings of the design variables, including the XPLT policy that selects allocations based on maximizing the expected reward. We then identified some constraints. First, we distinguished between a closed-loop setting versus a one-shot scenario, which is a constraint on the number of times we can execute the full real-world pipeline. We also discussed budget constraints to limit the cost of one execution of the pipeline.

By varying objectives, design variables, policies and constraints, we obtain a rich problem landscape. In what follows, we shall focus on a small portion of this landscape. Specifically, we turn our attention to selecting optimal allocations which maximize the multi-fidelity reward function in the one-shot setting, using the XPLT policy. In "Results and Discussion" section, we detail some simulation studies that measure the impact of various pipeline parameters on the overall effectiveness of the screening procedure in this setting.

A PRIOR MODEL FOR STAGE-WISE SCORES

As described above, we shall model the scores within a Bayesian context by maintaining a multivariate probability distribution of the scores. This distribution captures predictions for the scores, an uncertainty quantification for such predictions and estimated correlation between different scores. Starting from a prior distribution of such scores, we can integrate observed scores for candidates selected by a policy using Bayes's law. This results in an updated posterior distribution on scores. Key to this procedure is the prior distribution, which reflects our initial assumptions about the scores. Also essential is the ability to efficiently sample scores from this prior distribution, which we use to simulate the pipeline. Below, we detail these two aspects.

We shall consider a flattened representation of the scores $\mathbf{Y} = (Y_{i,j})$ so that $\mathbf{Y}$ is an nm-long vector whose $(jm + i)$-th entry is $Y_{i,j} = f_j(x_i)$. For our simulation studies, we shall assume that the scores follow a multivariate normal prior distribution, $\mathbf{Y} \sim \mathcal{N}(0, \Gamma)$. That is, we assume that the scores on average are centered at 0, which we view as a nominal baseline value. Statistically significant deviations away from this baseline value will represent "good" candidates. The covariance matrix Γ is an $nm \times nm$ matrix, whose entries describe the statistical relationship between the scores.

For our problem, we assume that the covariance is *separable*. That is, we can decompose Γ into the Kronecker product $\Gamma = \Sigma \otimes X$, where Σ and X are $n \times n$ and $m \times m$ covariance matrices, respectively. Here, Σ represents the relationship between stages and X represents the relationship between candidates. Namely, the covariance $\mathrm{Cov}(Y_{i,j}, Y_{k,\ell})$ is given by the product $\Sigma_{j,\ell} \cdot X_{i,k}$ of the (j, ℓ)-th and (i, k)-th entries of Σ and X, respectively. That is, the relationship between the scores of different candidates at different stages "factors through" assumed inherent statistical relationships between stages and candidates.

The assumption of separability is not a fundamental one, and we may imagine problem settings where the covariance relationship between different scores is non-separable. However, we choose this setting to perform our simulation studies for several reasons. First, it is computationally efficient (see section "Lazy Stage-wise Sampling"). The assumption also allows us to study the impact of stages and candidates covariance structure independently. Lastly, in contrast to other methods that consider similar separable structure (such as fitting a multi-task Gaussian process[24] to model several responses or tasks at once as a function of input variables), the assumption of separability is more natural in our multi-fidelity setting. Specifically, the different tasks we are fitting a multivariate distribution to are different models predicting the same response, albeit at different levels of fidelity. In this setting, it

is more natural to assume correlations between scores of two different candidates persist to some degree between different fidelity levels, i.e., the covariance is separable.

The optimal allocation problem is, in part, defined by this distribution. It serves as the prior distribution from which we sample ground truths to estimate expected rewards. Different distributions will result in varying effectiveness of the screening procedure. We wish to characterize the connection between the distribution and the screening procedure's efficacy. To do this, we generate distributions as follows.

Pick dimensions d_x and d_s. Then, independently and uniformly sample m vectors $\{\mathbf{x}_1, ..., \mathbf{x}_m\}$ from the d_x-dimensional unit hypercube, $\mathbf{x}_i \sim \mathrm{Unif}[0, 1]^{d_x}$. With these samples, define the candidate covariance matrix X

$$X_{i,j} = \sigma_x^2 \exp\left[-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\ell_x^2} \right], \quad i,j = 1, 2, , m. \quad (1)$$

Here, σ_x, ℓ_x are free parameters. Similarly define the stage covariance Σ matrix using n independent samples $\{\mathbf{s}_1, \mathbf{s}_n\}$, where $\mathbf{s}_j \sim \mathrm{Unif}[0, 1]^{d_s}$:

$$\Sigma_{i,j} = \sigma_s^2 \exp\left[-\frac{\|\mathbf{s}_i - \mathbf{s}_j\|_2^2}{2\ell_s^2} \right], \quad i,j = 1, 2, , n, \quad (2)$$

for free parameters σ_s, ℓ_s. We call the samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^m$ and $\mathbf{S} = \{\mathbf{s}_j\}_{j=1}^n$ the *latent representations* of candidates and stages. Different latent samples yield different covariance matrices and hence different prior distributions. The parameters σ_x and σ_s capture the magnitude of the scores assumed by the prior model. Without loss of generality, we shall assume both are equal to 1 for our simulation studies.

In reality, the covariance structure is problem-specific, yet the procedure above provides a method for systematically varying the covariance structure in a parametric way to study the impact of such structure on the effectiveness of the screening pipeline. Implicit in this generation procedure is the assumption that the covariances are *stationary* in the latent spaces. That is, scores between different candidates or stages are considered statistically similar if their latent representations are close together in the latent space, and the notion of closeness is the same wherever the latent representations are. This assumption of closeness in latent space as a proxy for similarity between candidates is used in many generative design methods (e.g.,[17]). Below, we study the impact of the latent dimensions d_s, d_x and length scales ℓ_s, ℓ_x, which impact this measure of closeness. There are other ways to sample or otherwise construct the covariance matrices, such as sampling such matrices from an inverse Wishart distribution[25] or considering a non-

stationary covariance structure over the latent space. Regardless of the sampling procedure, we can perform a similar simulation analysis below given the covariance matrices, Σ, X.

Lazy Stage-wise Sampling

Once the prior $\mathcal{N}(0, \Gamma)$ is defined, we can use it to sample ground truth for simulations. Naïvely, we may sample the entire set of nm scores at once. This is wasteful because many candidates are screened out of the pipeline, obviating the need for many of these scores in later stages. In reality, only $M = \sum_{j=1}^{n} m_j$ scores are needed to simulate the pipeline, and this sum is typically much smaller than nm. Indeed, in many applications, the stage costs increase exponentially, necessitating that allocations decrease by an order of magnitude at each stage. In this case, M is essentially km, for some constant k.

Computationally, naïvely sampling from a fixed distribution $\mathcal{N}(0, \Gamma)$ incurs a one-time $\mathcal{O}((nm)^3)$ computational cost and a per-sample costs of (1) nm samples from a univariate standard normal distribution and (2) an $\mathcal{O}((nm)^2)$ matrix-vector multiplication. Such per-sample costs are incurred for every simulation we run to obtain a reward distribution for a specific allocation. These costs are further multiplied by the number of allocations considered when solving the integer program to identify the optimal allocation. This is all to say that naïve sampling can increase the costs of our simulation-based analysis drastically. Improving how we sample scores is therefore important for efficient simulation studies.

We can gain significant speed-ups in a straightforward way using lazy stage-wise sampling. We sample scores for stage j only when the simulation of the pipeline arrives at this stage. At this point, we will know the subset of candidates that had survived up to this stage, and hence we shall only sample those scores. Suppose candidates $x_{i_1}, x_{i_{m_j}}$ have made it to stage j. Then, we can sample the scores $Y_{i_1 j}, , Y_{i_{m_j} j}$ *conditional* on all the previously sampled scores from prior stages. By properties of the multivariate normal distribution, these conditional scores are also multivariate normally distributed. By additionally taking advantage of the separable structure of the covariance matrix, we can perform the sampling in an efficient manner. See supplementary Sect. SI.1 for details (refer to online supplementary material).

Simulating a Pipeline with an Inter-stage Policy

With the ability to sample ground truth scores, we can simulate the execution of the pipeline under some pre-specified inter-stage policy and some fixed allocation $\mathbf{m}$. For the first stage, we evaluate all $m_1 = m$ initial candidates x_i by sampling scores $y_{i,1}$ for $i = 1, ..., m$. Using the sampled scores, an inter-stage policy selects m_2 candidates to advance to stage 2. For example, the *pure-exploitation inter-stage policy* simply selects the m_2 candidates with highest stage-1 scores.

Other inter-stage policies may consider the conditional distribution of the *final-stage* scores in light of the observed data. That is, conditioned on the observations $\{y_{i,1}\}$, we can consider the distribution of final scores $Y_{i,n}$ and select candidates for the second round based on this distribution. This distribution is also multivariate normally distributed. We may, for example, use a delayed exploitation policy, where we select candidates corresponding to the largest expected final scores under this distribution. We can attempt to balance between exploration and exploitation as well, using polices such as UCB. We do not dwell on other inter-stage policies in this article. Instead, we will focus solely on the pure-exploitation inter-stage policy.

Once the inter-stage policy selects the m_2 candidates for the second stage, their scores for this stage are sampled, conditioned on the sampled scores $\{y_{i,1}\}$ of the previous stage. Then, the process repeats: based on these scores, the appropriate number of candidates is selected for the next stage per the pre-specified allocation. For these candidates, scores from the next stage are sampled conditionally on the previously sampled scores, and so on. This is repeated until the evaluation of the last stage. After simulating the pipeline through the last stage, we compute the reward obtained for this simulation. Then, the procedure is repeated several times more, each simulation using a different ground truth sample drawn from the prior. These simulations yield an empirical distribution of the reward $R(\mathbf{m})$ for the given allocation $\mathbf{m}$. As described above, the meta-policy will use this reward distribution to select an allocation $\mathbf{m}^\star$.

In practice, this decision $\mathbf{m}^\star$ would be then implemented, running the real-life pipeline with this allocation. In the closed-loop design setting, we would use the data obtained from this real-life execution of the pipeline to update the Bayesian prior, which we would use in turn to select another allocation to try. In the one-shot setting, we receive a single *realized reward* $r(\mathbf{m}^\star)$ from the sole execution of the pipeline. Assuming the prior is representative of the real ground truth, this realized reward should be consistent with the distribution of rewards obtained via simulation. That is, under the *truth-from-prior* assumption (that the real ground truth is sampled from the prior distribution), the distribution of realized reward $r(\mathbf{m}^\star)$ is exactly that of $R(\mathbf{m}^\star)$, the simulated reward distribution used to select $\mathbf{m}^\star$ in the first place. In the simulations below, we explicitly enforce this truth-from-prior assumption.

Table I. Base parameter values used throughout the simulation studies

Parameter	Description	Value
m	Number of initial candidates	500
$C_{\max}$	Total budget	2500
d_x	Dimensionality of latent candidate representation	8
ℓ_x, σ_x	Parameters to candidate covariance function (Eq. (1))	1, 1
d_s	Dimensionality of latent stage representation	1
ℓ_s, σ_s	Parameters to stage covariance function (Eq. (2))	0.2, 1

RESULTS AND DISCUSSION

In this section, we report results of simulation studies where we simulate the entire pipeline definition and optimization procedure:

1. Sample latent representations of candidates and stages to define a prior distribution on scores.
2. Given the prior distribution, use an inter-stage policy to simulate the pipeline via lazy stage-wise sampling and obtain samples of reward.
3. Solve the integer program defining the meta-policy to select an optimal allocation $\mathbf{m}^\star$, performing (2) above as an inner loop.
4. Characterize how well the meta-policy selects optimal allocations through further out-of-sample simulations, here serving as a proxy for real-world implementation of the selected allocation $\mathbf{m}^\star$.

The outcome of step (4) above is a distribution for realized reward for the allocation selected by the meta-policy. Under the truth-from-prior assumptions, the distribution of the realized reward is the same as the distribution of the simulated rewards, and so we shall use the two interchangeably. Specifically, below we will simply refer to the expected optimal rewards,

$$\overline{R}(\mathbf{m}^\star) = \mathbb{E}[R(\mathbf{m}^\star)]$$
$$= \mathbb{E}[r(\mathbf{m}^\star)], \text{(under truth} - \text{from} - \text{prior assumption.)}$$

In the simulation studies below, we characterize how different pipeline parameters impact the screening procedure's efficacy, as measured by the expected optimal rewards obtained. We compare against a baseline policy, which we refer to as the *random* policy. Under the random policy, we expend as much of the budget $Cmax$ by testing candidates solely on the final stage, without any screening. We select candidates uniformly at random, with replacement. Once the budget is spent on these random final-stage trials, we calculate the reward in the same manner as the screening procedure: select the maximum score from among the randomly selected candidates. Repeating this exhaustive random selection several times results in an expected reward, $\overline{R}$, for the random policy.

We consider a three-stage pipeline, with the three stages representing a cheap machine-learning material property predictor, an expensive physics-based model and a physical experiment, respectively. We assign nominal exponentially increasing costs $\mathbf{c} = (1, 10, 100)$ for the stages. Unless otherwise stated, the simulations use a set of base parameters listed in Table I.

Effect of Budget $C_{\mathbf{max}}$ and Number of Initial Candidates m

One reason to carry out simulation studies is to assess the sensitivity of the screening procedure on specific parameters. The supplemental information (supplementary Sect. SI.2) provides results for several such simulation results, in which specific parameters were varied. Here, we focus on two for illustration.

Figure 1a shows the expected reward obtained as a function of the budget $Cmax$, which is varied from 1000 to 50,000. We note that a budget of 50,000 is exactly the cost to exhaustively evaluate the $m = 500$ candidates once during the final stage, each evaluation incurring a cost of $c_3 = 100$. The solid blue lines plot the expected optimal reward under the XPLT meta-policy, while the dashed black line shows the expected rewards under the random policy. The four plots each correspond to four different choices of priors and hence represent four different problems, each with unique statistical relations between candidates and stages. We note that, in general, the performance of both policies increases as the computational budget increases. Yet, depending on the distribution, the marginal returns obtained from increasing the budget varies. For example, in panel C, there is little marginal improvement to the expected reward even as we increase the budget by an order of magnitude. In panel B, we obtain good marginal improvement only up to $Cmax = 3000$. The bottom-right panel (panel D) shows it is possible for the screening procedure to consistently perform worse than random for some problem settings.

Figure 1b shows the expected reward obtained versus the number of initial candidates m, varying this number between 500 and 2000. We note that the performance of the random policy is not
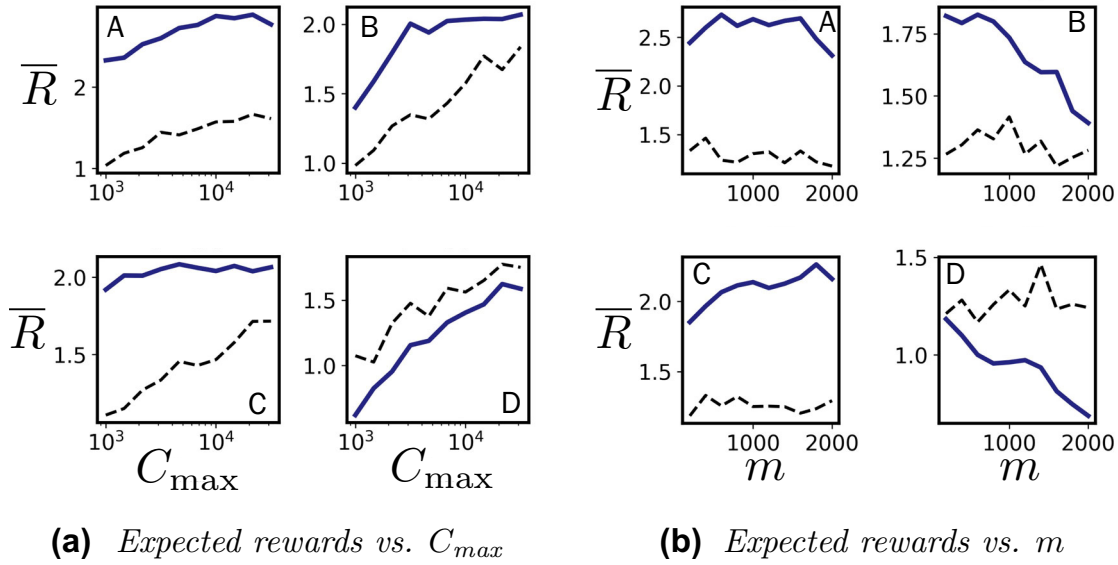
Fig. 1. (a) Expected rewards vs. *Cmax*. (b) Expected rewards vs. *m* Expected rewards vs. specific pipeline parameters. Solid blue lines are the expected optimal rewards obtained from the XPLT-based screening procedure, while the dashed black line shows the expected rewards obtained under the random (non-screening) policy. Each sub-panel (A, B, C and D) corresponds to a different prior distribution and hence a different problem. Specifically, the prior distributions differ with respect to stage-wise similarities, as described in "Effect of Prior Distribution" section (Color figure online).

sensitive to this value. With the fixed budget $Cmax = 2500$ and a final-stage cost $c_3 = 100$, the random policy can try 25 out of m total candidates. As the simulations suggest, the chance that 1 of 25 selected candidates yields a large reward is relatively stable (and low) for m between 500 and 2000. In contrast, the expected optimal rewards obtained from the XPLT policy show varying behavior depending on the prior distribution. In panel A, the expected optimal rewards are relatively insensitive to m, while in panel C, these rewards increase with m. Panels B and D show the opposite behavior, where the performance decreases with respect to m. As with the previous study, we see that the screening procedure performs worse than random in panel D.

It is interesting to note the different trends in expected optimal reward versus different problem parameters or distributions of the ground truth and how such trends compare to random. The above simulations and those detailed in supplementary Sect. SI.2 make clear that the analysis is not straightforward and that there is no one general "rule of thumb" when it comes to setting these parameters. Therefore, in practice, it is important to assess the problem setting (in the above: which of A, B, C or D) prior to executing a screening pipeline or even selecting policy parameters using ad hoc or heuristic methods. Simulations offer a systematic way of studying such effects.

Effect of Prior Distribution

As we see above, the trends of expected rewards could depend on the prior distribution, since the real ground truth is sampled from this distribution in these truth-from-prior studies. It is therefore desirable to characterize this dependence further and see whether there are any features of a distribution or problem that predict effectiveness of the screening procedure. The prior distribution is characterized by the latent samples of candidates, $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^m$, of stages $\mathbf{S} = \{\mathbf{s}_j\}_{j=1}^n$, and parameters such at the dimensionality of such samples d_x and d_s, and the covariance function hyperparameters ℓ_x, ℓ_s, σ_x and σ_s. The SI details simulation studies for these parameters. Here, we study the impact of the samples $\mathbf{X}$ and $\mathbf{S}$.

By considering several such latent samples, we obtain distributions of expected rewards, which are shown in Fig. 2. The left panel shows the induced distribution on expected reward as we vary the latent samples on candidates $\mathbf{X}$ for both the XPLT policy and the random baseline policy. As we see, the distributions for both policies are insensitive to the specific choice of latent samples, typified by the tight induced distributions. In contrast, the right panel shows these distributions as we vary stage latent samples $\mathbf{S}$. Here, we see that the expected reward is highly sensitive to the actual latent samples of stages, implying that the stage-wise covariance structure significantly impacts the overall effectiveness of the screening procedure. We even see that a significant proportion of sampled
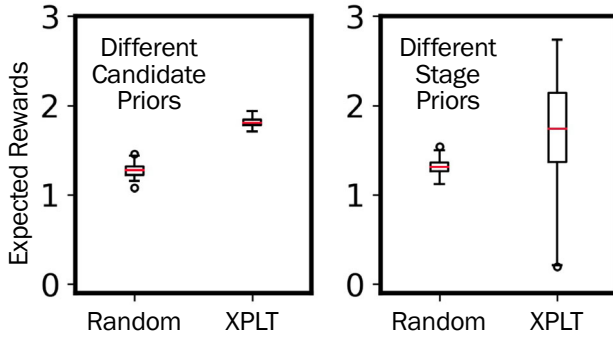
Fig. 2. Expected reward distributions over different priors.
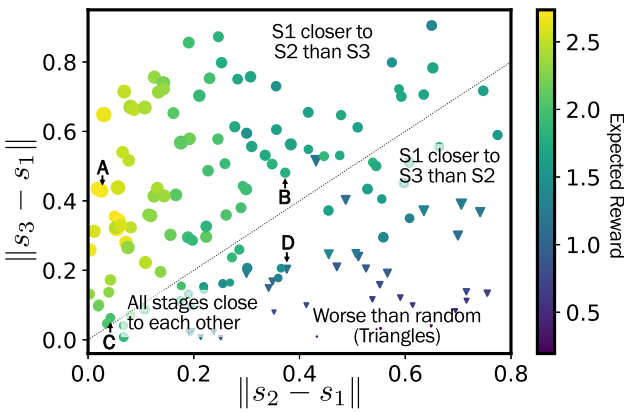

Fig. 3. Similarities between stages dictate expected rewards. This plot shows expected realized rewards for different prior distributions. Locations of the plotted points indicate the distance between first and second stages and first and third stages. Color indicates the expected realized rewards, while size is inversely proportional to the variance of the realized rewards. Inverted triangles indicate cases where the expected reward obtained via screening were less than that obtained via random sampling (Color figure online).

priors results in the screening procedure performing worse than random.

To further explore this phenomenon, we consider the optimal rewards obtained by the XPLT policy for several samples of $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)$ and hence for several different prior distributions. We characterize these rewards with respect to two geometric features derived from the latent stage representation: the distance between the latent representations of Stage 1 and Stage 2 $\|\mathbf{s}_2 - \mathbf{s}_1\|$ and the distance between Stage 1 and Stage 3 $\|\mathbf{s}_3 - \mathbf{s}_1\|$. These distances typify the statistical relationship between the stages. These data are displayed in Fig. 3.

Each dot in the figure corresponds to a specific sample of latent stages, i.e., a specific distribution for ground truth. Dots are located at coordinates

$(\|\mathbf{s}_2 - \mathbf{s}_1\|, \|\mathbf{s}_3 - \mathbf{s}_1\|)$. For example, points in the lower-left corner correspond to the case where the three stages are statistically similar, while dots in the top left corner correspond to the case where Stages 1 and 2 are similar, but Stages 1 and 3 are not. The color of a dot indicates the expected optimal reward given the specific prior distribution corresponding to that dot. The size of the dot is inversely proportional to the variance of the reward distribution. Smaller dots are cases where there is a wide distribution of rewards associated to the selected optimal allocation. Circles indicate the cases where the XPLT policy performs better than random given the same prior distribution, while inverted triangles show the cases where it performs worse than random. The points labeled A, B, C and D correspond to the specific prior distributions used in the single-parameter simulation studies in section "Effect of Budget C_{max} and Number of Initial Candidates m" above and in the SI.

The dashed line indicates the points where Stage 3 and Stage 2 are both equidistant to Stage 1. Points above this line refer to cases where Stage 1 is more statistically similar to Stage 2 than to Stage 3. Points below this line correspond to cases where Stage 1 is more similar to Stage 3 than to Stage 2. In this region, the statistical similarities between stages do not respect the order in which the stages are executed in the pipeline. From the plot, the screening procedure is not effective in this region, often performing worse than random. There is a general improvement as Stage 2 and Stage 1 become more similar, i.e., as we move right to left along the horizontal axis. Interestingly, this trend does not hold with respect to Stage 3. From the plot, there appears to be an optimal *dissimilarity* between Stages 1 and 3 that yield maximal expected rewards. For example, among the samples, Point A corresponds to the largest expected reward. For this point the distance between $\mathbf{s}_3$ and $\mathbf{s}_1$ is around 0.42.

This optimal dissimilarity for this model system again indicates the need to perform such simulations for real systems in practice. In this specific case, tuning similarity between stages to match the optimal settings identified through such simulation studies could mean tuning the accuracy of a ML model, increasing or decreasing simulation run times or resolutions, or introducing randomness (i.e., exploration) in selecting candidates to advance between stages. Having a proper assessment of the impact of such pipeline parameters could then result in more efficient and effective screening.

Throughput and Optimal Reward

Another pipeline study[26] examined the pipeline problem under different design variables and

objectives. There, authors considered score thresholds (rather than explicit allocations) as the design variables. They also considered the objective of optimizing pipeline *throughput*—the number of candidates that satisfy all thresholds to make it past the final stage, under similar cost constraints as those detailed above. In this setting, the pipeline is used to maximize the number of good candidates passing through the screening procedure. While not the same problem as considered here, we are nevertheless able to draw some connections between our work and this optimal throughput problem.

Figure 4 plots the expected optimal reward $\overline{R}(\mathbf{m}^\star)$ versus the final stage allocation $m_3^\star$, for different sampled prior distributions. Each dot corresponds to the expected optimal reward and final stage allocation obtained for a specific prior distribution. As before, the size of the dot is inversely proportional to the variance of the reward distribution. The red rectangles call attention to settings with high expected rewards (top) and those that have large final allocation (right). The final allocation is the measure for throughput. While in our study we optimized for expected rewards rather than throughput, we observe from the plot that the two quantities are correlated.

Namely, we note that for the settings resulting in the highest expected reward (top box), the final stage allocation $m_3^\star$ was around 10 or larger. Given the final stage cost $c_3 = 100$ and budget $Cmax = 2500$, this means that optimal allocations yielding highest expected rewards were expending
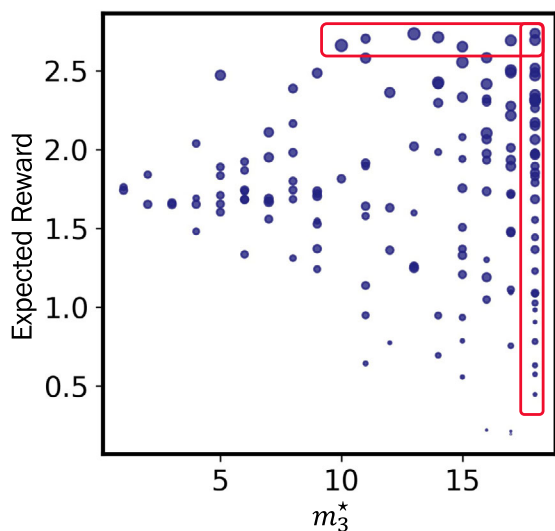
around 40 to 60% of the budget on final stage trials. We note that for cases where the optimal final allocation is large (right box), there is a spread of both high and low expected rewards. Yet, the concentration is biased toward the high expected reward regime. Moreover the high-throughput / low expected reward cases are associated with large variance of reward distribution. These results show that while the relationship between throughput and expected rewards may be complex, there are broad agreements between the two optimization objectives, though there are cases where this agreement may not hold.

CONCLUSION

In this article, we defined a model and algorithm for simulating pipelines and identified places where such methods can be used to perform decision-making under uncertainty via simulation-based policies. We codified how various different pipeline objectives fit inside the common framework of optimizing expected rewards under uncertainty and alluded to a multi-faceted, rich problem set that belies the ubiquitous experimental protocol of multi-stage screening.

Examining one part of this problem landscape, we focused on analyzing a popular mode for screening: simply selecting the top m_j candidates from the previous stage to pass onto the next stage. We showed how to optimize expected rewards with respect to a specific design parameter choice, allocation, in a one-shot setting. In this specific context, our simulation studies showed a variety of trends as we attempted to characterize the effectiveness of the screening procedure with respect to pipeline or problem parameters. We most importantly described how stage-wise covariance structure played a significant role in determining this efficacy. We demonstrated how simulated studies allowed us to *quantify* the performance, its dependence on parameters, and in comparison with a random baseline policy.

We hope to have demonstrated the need for proper modeling and analysis in multi-stage screening pipeline optimization, decision-making and parameter calibration. We believe that the models, policies and simulation methods presented in this paper provide important tools in this analysis.

ACKNOWLEDGEMENTS

Fig. 4. Throughput correlates with expected optimal reward. This figure plots expected optimal reward versus final-stage allocation $m_3^\star$ of the optimal allocation $\mathbf{m}^\star$ selected by the XPLT policy. Each dot corresponds to a different prior distribution. The size of each dot is inversely proportional to the variance of the reward distribution $r(\mathbf{m}^\star)$.

CONFLICT OF INTEREST

On behalf of all authors, the corresponding author states that there is no conflict of interest.

SUPPLEMENTARY INFORMATION

The online version contains supplementary material available at https://doi.org/10.1007/s11837-022-05368-z.

REFERENCES

1. A.D. Andricopulo, R.V. Guido, and G. Oliva, *Curr. Med. Chem.* 15, 37 (2008).
2. G. Schneider and H.-J. Böhm, *Drug Discov. Today* 7, 64 (2002).
3. M. Kandeel and M. Al-Nazawi, *Life Sci.* 251, 117627 (2020).
4. J. Bajorath, *Nat. Rev. Drug Discov.* 1, 882 (2002).
5. B.K. Shoichet, *Nature* 432, 862 (2004).
6. R. Gómez-Bombarelli, J. Aguilera-Iparraguirre, T.D. Hirzel, D. Duvenaud, D. Maclaurin, M.A. Blood-Forsythe, H.S. Chae, M. Einzinger, D.-G. Ha, T. Wu, G. Markopoulos, S. Jeon, H. Kang, H. Miyazaki, M. Numata, S. Kim, W. Huang, S.I. Hong, M. Baldo, R.P. Adams, and A. Aspuru-Guzik, *Nat. Mater.* 15, 1120 (2016).
7. E.O. Pyzer-Knapp, C. Suh, R. Gómez-Bombarelli, J. Aguilera-Iparraguirre, and A. Aspuru-Guzik, *Annu. Rev. Mater. Res.* 45, 195 (2015).
8. E.O. Pyzer-Knapp, G.N. Simm, and A.A. Guzik, *Mater. Horiz.* 3, 226 (2016).
9. K. Abdel-Latif, R.W. Epps, F. Bateni, S. Han, K.G. Reyes, and M. Abolhasani, *Adv. Intel. Sys.* 3, 2000245 (2021).
10. A.E. Gongora, K.L. Snapp, E. Whiting, P. Riley, K.G. Reyes, E.F. Morgan, and K.A. Brown, *iScience* 24, 102262 (2021).
11. S. Baek and K.G. Reyes, *Comput. Mater. Sci.* 193, 110385 (2021).
12. A. Tropsha and A. Golbraikh, *Curr. Pharm. Des.* 13, 3494 (2007).
13. C.-A. Azencott, A. Ksikes, S.J. Swamidass, J.H. Chen, L. Ralaivola, and P. Baldi, *J. Chem. Inf. Model.* 47, 965 (2007).
14. Q.U. Ain, A. Aleksandrova, F.D. Roessler, and P.J. Ballester, *WIRES Comput. Mol. Sci.* 5, 405 (2015).
15. P.J. Ballester and J.B. Mitchell, *Bioinformatics* 26, 1169 (2010).
16. B. Sanchez-Lengeling and A. Aspuru-Guzik, *Science* 361, 360 (2018).
17. J. Lim, S. Ryu, J.W. Kim, and W.Y. Kim, *J. Cheminform.* 10, 1 (2018).
18. W. Gao and C.W. Coley, *J. Chem. Inf. Model.* 60, 5714 (2020).
19. K. Deb, Multi-objective optimization. In *Search Methodologies* (Springer, 2014), pp 403–449.
20. N. Beume, B. Naujoks, and M. Emmerich, *Eur. J. Oper. Res.* 181, 1653 (2007).
21. Y. Wang, K.G. Reyes, K.A. Brown, C.A. Mirkin, and W.B. Powell, *SIAM, J. Sci. Comput.* 37, B361 (2015).
22. J. Snoek, H. Larochelle, and R.P. Adams, Practical Bayesian optimization of machine learning algorithms. In: F. Pereira, C. Burges, L. Bottou, K. Weinberger (Eds.) Advances in Neural Information Processing Systems, Vol. 25, (Curran Associates, Inc., 2012).
23. P. Auer, N. Cesa-Bianchi, and P. Fischer, *Mach. Learn.* 47, 235 (2002).
24. E.V. Bonilla, K. Chai, and C. Williams, Multi-task Gaussian process prediction. In: J. Platt, D. Koller, Y. Singer, and S. Roweis (Eds.) *Advances in Neural Information Processing Systems*, Vol. 20, (Curran Associates, Inc., 2007).
25. A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin, *Bayesian Data Analysis* (Chapman and Hall/CRC Press, London, 2013), pp. 72–74.
26. 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:2109.11683.