Bayesian optimization and attribute adjustment

Stephan Eismann
CS Department
Stanford University
Stanford, CA 94305

Daniel Levy
CS Department
Stanford University
Stanford, CA 94305

Rui Shu
CS Department
Stanford University
Stanford, CA 94305

Stefan Bartzsch
iRT Department
Helmholtz-Zentrum Munich
Munich, Germany

Stefano Ermon
CS Department
Stanford University
Stanford, CA 94305

Abstract

Automatic design via Bayesian optimization holds great promise given the constant increase of available data across domains. However, it faces difficulties from high-dimensional, potentially discrete, search spaces. We propose to probabilistically embed inputs into a lower dimensional, continuous latent space, where we perform gradient-based optimization guided by a Gaussian process. Building on variational autoncoders, we use both labeled and unlabeled data to guide the encoding and increase its accuracy. In addition, we propose an adversarial extension to render the latent representation invariant with respect to specific design attributes, which allows us to transfer these attributes across structures. We apply the framework both to a functional-protein dataset and to perform optimization of drag coefficients directly over high-dimensional shapes without incorporating domain knowledge or handcrafted features.

1 INTRODUCTION

Developing enhanced designs is an overarching goal across engineering disciplines ranging from the optimization of planes in aeronautics (Simpson et al., 2001) and batteries for electric vehicles (Grover et al., 2018) to the development of proteins in bioengineering (Damborsky and Brezovsky, 2014). The different optimization efforts often face the same challenges in form of search-space complexity and costly design evaluations which render naive exhaustive search infeasible and make human-expert knowledge a key success factor. The ever increasing amounts of experimental data have to be considered, however, pose new challenges to manual analysis.

Increasing amounts of data open new opportunities for statistical design approaches. In this context Bayesian optimization has emerged as a data-driven tool for automated design optimization (Shahriari et al., 2016). Bayesian optimization is a model-based approach with a prescribed prior belief on the functional score of designs. Given data, we sequentially update this belief and optimize a surrogate function to our true objective. The choice of this surrogate function thereby trades exploration vs. exploitation in the design space.

By leveraging the problem structure, Bayesian optimization can be much more sample efficient than random search (Snoek et al., 2012b), but it is not immune to the curse of dimensionality. Signal is often sparse in high-dimensional input spaces of many real-world design problems. In addition, desirable target applications like drug or material design involve optimization over *discrete* structures, where even optimizing the model-based surrogate is difficult.

We target the input-space challenges of high-dimensionality and discrete designs by combining Bayesian optimization with deep generative modeling. Specifically, we built on the architecture by Gómez-Bombarelli et al. (2016a; 2016b) combining a variational autoencoder (VAE) and Gaussian process (GP) regression. VAEs (Kingma and Welling, 2013) are probabilistic models which map high-dimensional, possibly discrete inputs to a lower dimensional continuous space. The encoder consists of a neural network whose feature-construction ability is leveraged for dimensionality reduction. We learn a GP on top of the latent space as its predictions enjoy uncertainty estimates such that we can explore the design space based on a confidence measure. At the same time, the continuous space now allows us to use gradient-based methods for optimization.

In this work we make the following contributions: (i) We propose to use parametric label-guidance for the autoencoder and demonstrate how this results in increased

label-prediction accuracy for the datasets we consider. (ii) We present a corresponding graphical model and derive a variational lower bound on the marginal log-likelihood. The variational bound provides us with a principled way of incorporating unlabeled data in the joint training procedure. We show that incorporating unlabeled data results in enhanced reconstruction accuracy for our datasets. (iii) We perform Bayesian design optimization on two different domains: proteins and shapes in laminar flow. Having access to a physics simulator, we show the validity of the approach in the hydrodynamics setting. (iv) We propose an adversarial model extension to render the latent representation invariant with respect to specific, real-valued design attributes. To compensate for the loss of information, we provide these attributes as additional arguments to the decoder. This allows us to transfer attributes across designs when we generate a design from its latent representation.

2 PROBLEM SETUP

We consider the setting of a high-dimensional, possibly discrete input space of designs $\mathcal{X}$. Each design x is associated with a real-valued score $y \in \mathbb{R}$ drawn from a conditional distribution $p^*(y|x)$.

Our goal is to find a design $x \in \mathcal{X}$ that maximizes the expectation $\mathbb{E}[y|x]$. We are given access to an oracle providing a sample of $y \sim p^*(\cdot|x)$, however, we assume that obtaining a sample (evaluating y for a given design x) is expensive. For example, it might require an expensive simulation or conducting a lab experiment.

Furthermore, we assume to have access to samples $D_u = \{x_1, \cdots, x_{N_u}\}$ from $\mathcal{X}$, and a (small) number of labeled examples $D_\ell = \{(x_1, y_1), \cdots, (x_{N_\ell}, y_{N_\ell})\}$. where each pair (x, y) corresponds to a design and a measurement of its score. We assume that labeled and unlabeled examples are sampled from the same marginal distribution.

2.1 BAYESIAN OPTIMIZATION

Bayesian optimization is a data-driven tool to optimize expensive black-box functions. In this model-based approach we start with a prior belief on the functional relationship between inputs and outputs, and update it sequentially as new data is acquired. As actual function evaluations are expensive, we aim to optimize a surrogate or acquisition function instead. A popular choice of acquisition function is expected improvement (EI) (Jones et al., 1998) which strikes to balance exploration vs. exploitation in the search space. To calculate EI we require a prediction with uncertainty for the black-box function values. Gaussian processes (GPs) provide uncertainty quantification and as such they are a standard model in Bayesian optimization. In the framework of GPs we assume that given a finite number of n inputs $x_{1:n}$, the function values $f(x_{1:n})$ are jointly Gaussian and the observations $y_{1:n}$ are normally distributed given f (Rasmussen and Williams, 2006). Because of the GP guidance, it can be more sample efficient than random search, however, Bayesian optimization still faces the challenges of data sparsity when operating in high dimensions. In addition, gradient-based optimization of the (EI) surrogate is not a priori applicable for discrete inputs. Finally, the benefit of uncertainty quantification comes at a price, as learning in the nonparametric GP model is cubic in the number of inputs. To circumvent this bottleneck, different sparse approximations have been developed. One approach to reduce the computational complexity is to calculate the covariance matrix with respect to m inducing points instead of n data points and typically $\mathrm{m} \ll \mathrm{n}$ with complexity $\mathcal{O}\left(\mathrm{m}^2\mathrm{n}\right)$ (Titsias, 2009; McIntire et al., 2016).

2.2 VARIATIONAL AUTOENCODERS

A variational autoencoder is a generative model defining a joint probability distribution between a latent variable z and inputs x. We commonly assume a simple Gaussian prior distribution $p(z)$ and model the input data distribution as a more complex conditional distribution $p_\Psi(x|z)$ where Ψ are the parameters of a neural network. Directly optimizing the marginal likelihood is intractable as it requires integration over the latent space. Kingma and Welling (2013) circumvent this obstacle by proposing an auxiliary inference distribution $q_\Phi(z|x)$ and derive a variational lower bound on the log likelihood

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} &= \mathop{\mathbb{E}}_{q_\Phi(z|x)}\left[\log p_\Psi(x|z)\right] - \mathrm{D}_{\mathrm{KL}}\left(q_\Phi(z|x)\,||p(z)\right) \\ &\leq \log p(x)\end{aligned} \tag{1}$$

Maximizing this objective can be naturally interpreted as minimizing the reconstruction loss of a probabilistic autoencoder and regularizing the posterior distribution towards the prior. Kingma et al. (2014) extend this work to the semi-supervised setting considering both labeled and unlabeled data.

3 BAYESIAN OPTIMIZATION AND A SHAPED LATENT SPACE

We address the optimization challenges of high dimensions and discrete spaces in Bayesian Optimization by combining Gaussian process regression with variational autoencoding. In addition, we further shape the latent space through adversarial training. By learning an invariant latent representation regarding input-specific attributes we are able to transfer these attributes across inputs.

We consider the directed graphical model of inputs x, corresponding labels y and latent variables z shown in Figure 1. Given z, we assume x and y to be independent:

$$p\left(x, y|z\right) = p_{\Psi}\left(x|z\right) p_{\Theta}\left(y|z\right) \ . \tag{2}$$

The data distribution $p_{\Psi}\left(x|z\right)$ is modeled as either multinomial (protein dataset) or multivariate normal with fixed covariance (shape dataset). The discriminative $p_{\Theta}\left(y|z\right)$ is modeled as standard normal $\mathcal{N}\left(\mu_{\theta}(z), 1\right)$. Both distributions are parametrized through neural networks with parameters Ψ and Θ.

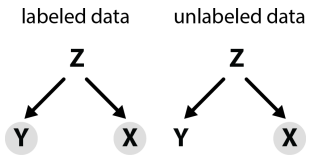

Figure 1: Graphical model connecting input space x with latent variable z and label y. The model assumes conditional independence of x and y given z. The gray shading marks observed quantities.

3.1 SEMI-SUPERVISED LEARNING

Given labeled and unlabeled data D_{ℓ} and D_u, respectively, we aim to optimize the likelihoods $p\left(x, y\right)$ and $p\left(x\right)$. As in the case of the VAE this is intractable to compute due to integration over z and we instead resort to variational lower bounds.

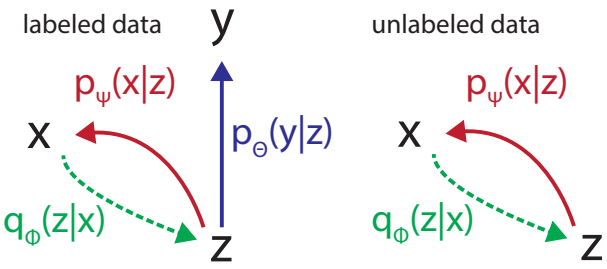

Figure 2: Illustration of the parametrized distributions involved in the derivation of the variational lower bounds.

Introducing the auxiliary model $q_{\Phi}\left(z|x\right)$ and using Jensen's inequality, we derive a variational lower bound on the log-likelihood of labeled data (x, y). The independence of the proposal distribution regarding y reflects the view that x contains all information on y. An illustration of the parametrized distributions is shown in Figure 2.

$$\begin{aligned}
\log p\left(x, y\right) &= \log \int p\left(x, y|z\right) p(z) \mathrm{dz} \\
&= \log \int \frac{p\left(x, y|z\right)}{q_{\Phi}\left(z|x\right)} q_{\Phi}\left(z|x\right) p\left(z\right) \mathrm{dz} \\
&= \log \mathop{\mathbb{E}}_{z \sim q_{\Phi}(z|x)} \left[p\left(x, y|z\right) \frac{p(z)}{q_{\Phi}\left(z|x\right)} \right] \\
&\geq \mathop{\mathbb{E}}_{z \sim q_{\Phi}(z|x)} \left[\log p\left(x, y|z\right) \right] \\
&\quad - \mathrm{D}_{\mathrm{KL}}\left(q_{\Phi}\left(z|x\right) || p(z)\right) \\
&= \mathop{\mathbb{E}}_{z \sim q_{\Phi}(z|x)} \left[\log p_{\Psi}\left(x|z\right) + \log p_{\Theta}\left(y|z\right) \right] \\
&\quad - \mathrm{D}_{\mathrm{KL}}\left(q_{\Phi}\left(z|x\right) || p(z)\right) \\
&\equiv \mathcal{L}_{\ell}
\end{aligned} \tag{3}$$

where D_{KL} indicates Kullback-Leibler divergence.

In contrast, we find in the case of unlabeled data

$$\begin{aligned}
\log p\left(x\right) &= \log \int \int p\left(x, y|z\right) p(z) \mathrm{dz} \, \mathrm{dy} \\
&= \log \int \int q_{\Theta}\left(y|z\right) \frac{p\left(x, y|z\right)}{q_{\Theta}\left(y|z\right)} \\
&\quad q_{\Phi}\left(z|x\right) \frac{p(z)}{q_{\Phi}\left(z|x\right)} \mathrm{dz} \, \mathrm{dy} \\
&\geq \mathop{\mathbb{E}}_{z \sim q_{\Phi}(z|x)} \left[\log p_{\Psi}\left(x|z\right) \right] \\
&\quad - \mathrm{D}_{\mathrm{KL}}\left(q_{\Phi}\left(z|x\right) || p(z)\right) \\
&\equiv \mathcal{L}_{u}
\end{aligned} \tag{4}$$

where we recover the ELBO objective of the VAE. Modeling the proposal distribution $q_{\Phi}\left(z|x\right)$ as multivariate Gaussian and assuming a Gaussian prior $p(z) = \mathcal{N}\left(0, \mathrm{I}\right)$ results in an analytic expression for the divergence term. Together $\mathcal{L}_u$ and $\mathcal{L}_{\ell}$ form a joint lower bound in the semi-supervised setting. During training we optimize a weighted sum of the two bounds. Specifically, labeled and unlabeled data share the same encoder $q_{\Phi}\left(z|x\right)$ and decoder $p_{\Psi}\left(x|z\right)$.

3.2 DESIGN OPTIMIZATION ON THE LATENT SPACE

After the designs are embedded in a lower dimensional continuous latent space we can now use a GP to perform Bayesian optimization. The algorithm for the joined procedure of latent embedding and following optimization is outlined in pseudocode in Algorithm 1.

For each (x_i, y_i) pair from the labeled set, we can compute the mean value of $q_{\Phi}(z|x_i)$ that we denote z_i. This effectively embeds the labeled inputs from D_{ℓ} in the lower-dimensional latent space. We can then fit a GP on

the set $D'_\ell := \{(z_i, y_i)\}_{i \leq N_\ell}$. Depending on the dataset size this is either a full GP or a sparse approximation with inducing points (Titsias, 2009). Subsequently we perform iterative optimization by sampling points from the latent prior $p(z)$ and maximizing expected improvement via gradient ascent. Next, we generate new designs corresponding to the latent points with largest EI value using the decoder, leveraging the generative capability of a VAE. Finally we evaluate the black-box function for these designs. The new data is appended to our dataset and we continue ad libitum. For simplicity, we do not retrain the VAE with each dataset expansion.

Algorithm 1 VAE-guided Bayesian Optimization

Input: Unlabeled data D_u, labeled data $D_\ell = \{(x_i, y_i)\}_{i \leq N_\ell}$, fitness function f, parameters α, β, κ.
$y_{\max} \leftarrow \max_{i \leq N_\ell} y_i$
TrainVAE:
  minimize$_{\Theta,\Phi,\Psi}$ $\alpha \mathcal{L}_u(D_u) + \beta \mathcal{L}_\ell(D_\ell)$
  $D'_\ell \leftarrow \{(z_i, y_i)\}_{i \leq N_\ell}$ with z_i mean of $q_\Phi(\cdot|x_i)$
loop:
  GP $\leftarrow$ **FitGP** $(D'_\ell; \kappa)$
  parallel loop:
    sample $z_i^0 \sim \mathcal{N}(0, \mathrm{I})$
    $(z_i^*, f_i^*) \leftarrow \max_z \mathrm{EI}\left(z, z_0 = z_i^0, y_{\max}, GP\right)$
  $\hat{z} \leftarrow z_b$ with $b \leftarrow \arg\max_i f_i^*$
  $\hat{x} \leftarrow \mathrm{decoder}_\Phi(\hat{z})$ ▷ Create design
  $\hat{y} \leftarrow f(\hat{x})$ ▷ Evaluate design $\hat{x}$
  Add $(\hat{x}, \hat{y})$ to D_ℓ and $(\hat{z}, \hat{y})$ to D'_ℓ
  $y_{\max} \leftarrow \max\{y_{\max}, \hat{y}\}$
return D_ℓ

3.3 ADJUSTING ATTRIBUTES AT TIME OF DECODING

We consider a setting in which we have successfully used our Bayesian optimization framework to find an enhanced design x which we generated by decoding its corresponding latent representation z. In addition, let a be a real-valued attribute intrinsic to a given input design x which is uncorrelated to the functional score y of the design. Taking the case of car designs as an example, y could be a measure of the car's aerodynamic properties and a reflect the car's color.

The joint probability distribution $p(a, x, y, z)$ factorizes as

$$p(a, x, y, z) = p_\Psi(x|a, z) p_\gamma(a|z) p_\Theta(y|z) p(z) \quad (5)$$

with the additional inference network $p_\gamma(a|z)$ (Figure 3). We assume that the input design x and the attribute a are observed variables, and that the label y is sometimes observed (i.e. the semi-supervised setting).

The question we consider is whether we can transfer an attribute across designs, i.e., can we decode our optimized latent representation to designs which share the optimal score but differ with respect to the value of attribute a. Referring again to our example of car designs, the attribute adjustment would consist in changing a car's color after finding an aerodynamically optimal design.

Our strategy to enable attribute adjustment is to enforce a latent representation which is invariant to a. If the latent space contains no information on the attribute, the decoder $p_\Psi(x|z, a)$ is forced to learn how to impose a on z in order to achieve proper design reconstruction. We can then adjust attributes by decoding optimized latent points with an attribute value of our choice.

To enforce this invariance, we add adversarial training to the training objective. We formalize this in the following maxmin expression:

$$\max_\Phi \min_\gamma \mathop{\mathbb{E}}_{x \sim q(x)} \mathop{\mathbb{E}}_{z \sim q_\Phi(z|x)} \left[\left(a(x) - \hat{a}_\gamma(z) \right)^2 \right] \quad (6)$$

Here a is the attribute we want to be invariant to, and $\hat{a}_\gamma$ is an estimator of a given the latent set z. The notation $a(x)$ emphasizes the fact that every design x has an intrinsic attribute value a. We model $p_\gamma(a|z)$ as Gaussian with fixed covariance and predict the mean using a neural network with parameters γ. If we assume that p_γ is expressive enough and the network trained such that it can take advantage of all information z has on a, then the objective minimizes the mutual information $\mathrm{I}(a; z)$ and p_γ is forced to settle on mean prediction. Note that this adversarial training objective does not depend on the observation of the label y and can thus leverage both labeled and unlabeled data.

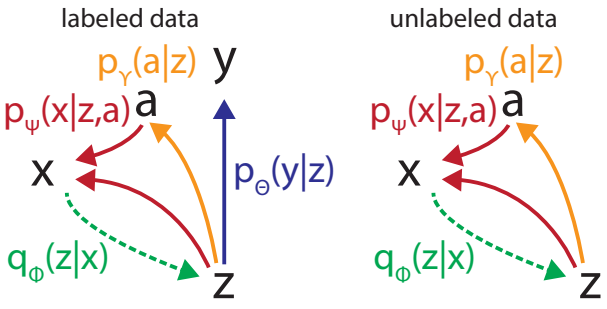

Figure 3: The augmented architecture with adversarial network $p_\gamma(a|z)$ mapping from latent space z to attribute a. Providing a as additional input to the decoder encourages an a-invariant latent representation.

4 DATASETS

We empirically evaluate the performance of our method on two datasets.

4.1 PROTEIN-FITNESS LANDSCAPE

Proteins are of paramount importance for biological systems and in industrial applications such as food processing or biomass conversion. As such the ability to design enhanced proteins is desirable. Proteins form both a high-dimensional as well as discrete design space as a protein is defined by an amino-acid sequence with alphabet size 20.

Protein optimization is especially challenging as (i) the number of target amino-acid sequences grows exponentially with the number of considered amino-acid mutations and (ii) only a very small fraction of all amino-acid sequences results in a functional protein (Keefe and Szostak, 2001).

We base our protein-optimization approach on a large fitness-landscape exploration study of the green fluorescent protein from *Aequorea victoria* (avGFP) (Sarkisyan et al., 2016). GFP is a widely used label-protein in fluorescence microscopy with a sequence of 237 amino acids. Our specific dataset consists of $51,715$ different protein sequences D_ℓ generated by random mutagenesis from the avGFP sequence and associated fluorescence values y as measured by fluorescence-activated cell sorting. On average each protein sequence contains 3.7 mutations compared to avGFP.

Amino acid sequences of the avGFP variants are encoded in a one-hot-style manner through a matrix of size 20×237 – accounting for the 20 essential amino acids and the sequence length of avGFP. All entries of the columns are 0 except for one 1 encoding the amino acid at the specific sequence position.

4.2 DRAG IN LAMINAR FLUID FLOW

The second dataset consists of 5100 two-dimensional shapes x and scalar drag coefficients y associated with the resistance these shapes experience in a constant fluid flow. We consider the case of laminar flow around an object in two dimensions as it allows us to generate training and test data, and perform Bayesian optimization at relatively low computational cost.

We generate the dataset by numerically solving the Navier-Stokes equations (Lifshitz and Landau, 1959) which provide a theoretical description of fluid flow around objects. Figure 4 shows example simulations from the dataset generation. Generated shapes are resized to 42×56 pixels to reduce memory requirements. Further details on the hydrodynamics simulation can be found in the appendix.

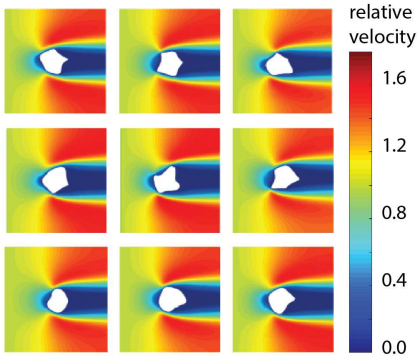

Figure 4: Finite-element simulations of fluid flow around random shapes in two dimensions. The left wall defines the fluid inlet.

5 EXPERIMENTS

In a first set of experiments we investigate the effect label-guided encoding and training with additional unlabeled data have on inference and autoencoder reconstruction error. The second part is concerned with design optimization of proteins and shapes. Finally, we demonstrate how we can use invariant learning to adjust shape attributes, namely area, with little effect on the drag coefficient.

5.1 INFERENCE AND RECONSTRUCTION ERROR

We consider the effect of label-guided encoding and adding unlabeled data to the training on 1) inference of y and 2) autoencoder reconstruction error as defined through the first term in the ELBO objective for the test set.

Tables 1 and 2 summarize the results for the protein and shape dataset, respectively. Details on the model architectures can be found in the appendix.

In general, we notice the positive effect label-guided encoding has on test set prediction. Column 'NN' shows the relative prediction error when using $p_\Theta(y|z)$ for inference. In absence of unlabeled data ($N_U = 0$) we consider two settings whose corresponding values are separated by a backslash: i) training $p_\Theta(y|z)$ and the autoencoder jointly (left value) and ii) training autoencoder and discriminative network sequentially (right value). We observe that the discriminative network guides the dimensionality reduction such that label-relevant information

Table 1: Protein dataset: Normalized test set prediction errors for different allocations of labeled (N_L) and unlabeled (N_U) training data. NN, GP-NN and GP-LAT describe the neural network parametrized by Θ and GPs trained on the neural-network features and the latent space, respectively. REC describes relative reconstruction error for test set designs.

N_L/ N_U	NN	GP-NN	GP-LAT	REC
30K/10K	1.04	2.43	1.82	**1.00**
30K/0	**1.00**/1.48	2.45	1.79	1.03
15K/15K	1.22	2.49	1.93	1.14
15K/0	1.31/1.57	2.58	2.18	1.64
5K/5K	1.18	2.62	1.89	1.90
5K/0	1.44/1.70	2.58	1.97	3.11

Table 2: Hydrodynamic dataset: Normalized test set prediction errors for different allocations of labeled (N_L) and unlabeled (N_U) training data. NN, GP-NN and GP-LAT describe the neural network parametrized by Θ and GPs trained on the neural-network features and the latent space, respectively. REC describes relative reconstruction error for test set designs.

N_L/ N_U	NN	GP-NN	GP-LAT	REC
2500/2000	**1.00**	1.41	1.02	**1.00**
2500/0	1.21/1.38	1.44	1.22	1.04
1500/2000	1.10	1.31	1.24	1.07
1500/0	1.19/1.41	1.19	1.07	1.17
500/2000	1.91	1.64	1.79	1.31
500/0	2.05/1.98	1.61	2.02	1.82

in the amino-acid sequence or shape is encoded with enhanced accuracy.

In addition, we also observe a positive effect of adding unlabeled data with respect to the reconstruction error (REC). The effect is more pronounced when less labeled data is available. Incorporating unlabeled data is also beneficial for NN prediction error in almost all cases.

In order to obtain uncertainty measures for the predictions we train and compare two GP models with squared-exponential kernel. Model 1 is trained on the latent-space embedding of the training data (GP-LAT). Model 2 is trained on the features learned by the discriminative neural network $p_\Theta(y|z)$ (GP-NN). Both models consider the situation in which $p_\Theta(y|z)$ and the autoencoder have previously been trained jointly.

We account for dimension-specific length scales in the covariance function such that the Gaussian process can filter irrelevant dimensions. For the protein dataset we use a sparse approximation with 500 inducing points such that the prediction performance of both GP models is impaired compared to the neural network (NN columns in the Tables). Comparing the GPs among each other we note the in general much better performance of the GP trained on the latent-space coordinates.

5.2 OPTIMIZATION OF PROTEIN AND SHAPE DESIGNS

We follow the algorithm outlined in Algorithm 1 to optimize shape and protein designs.

5.2.1 Design of New Protein Variants

A schematic of the optimization framework is shown in Figure 5A. To find the best local EI maxima we independently sample 20,000 start points from the prior $p(z) \sim \mathcal{N}(0, I)$ and perform gradient ascent for each point. Figure 5B visualizes amino-acid mutation sites apparent in the highest-ranked protein variants on the structure of avGFP.

While only experimental verification can provide a precise assessment of the model performance, comparison with the data from literature on development of GFP variants shows that genotypes predicted by our model are free from known deleterious mutations such as mutations in the chromophore-forming amino acids and catalytically active E222 residue (Chudakov et al., 2010). Most of the mutated amino acid side chains in predicted genotypes are oriented towards the solvent in the protein beta barrel structure, in accordance with experimental observations (Sarkisyan et al., 2016) and with the evolutionary conservation of internally oriented residues (Chudakov et al., 2010). Moreover, some predicted genotypes carry combinations of substitutions known to increase brightness of avGFP, such as the F99S/M153T pair of substitutions that in combination with mutation V163A was reported to result in avGFP being 42 times brighter when expressed *in vivo* (Battistutta et al., 2000).

5.2.2 Design of Drag-reduced Shapes

The time and resources required for protein synthesis and functional testing render it expensive to use this application for several rounds of Bayesian optimization given the purpose of this paper and to explore technical model aspects in more detail. For this reason we use a set of two-dimensional shapes and the drag these shapes face under laminar flow conditions. We can calculate the drag forces based on the Navier-Stokes equations in a physics simulation. As such we can perform closed-loop optimization with the goal of finding drag-reduced shapes.

Figure 6 shows a schematic of the optimization procedure which is analogous to the protein case. We generate

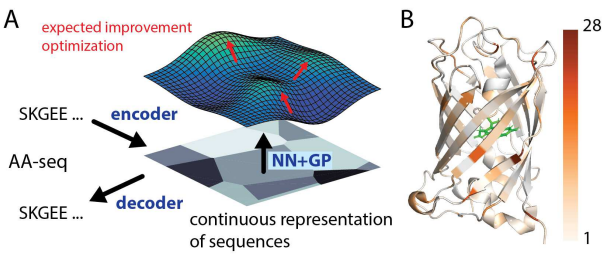

Figure 5: **(A)** Schematic illustrating the optimization of amino-acid sequences to generate variants of the green fluorescent protein (GFP) with enhanced brightness. (**B**) The structure of GFP annotated with the distribution of mutations among 100 sequences suggested by the design algorithm. The main chromophore complex is shown in green.

new object shapes by optimizing EI with respect to the smallest drag coefficient in our training set. Promising latent points are decoded to shape images and their drag coefficients evaluated in our hydrodynamics simulator.

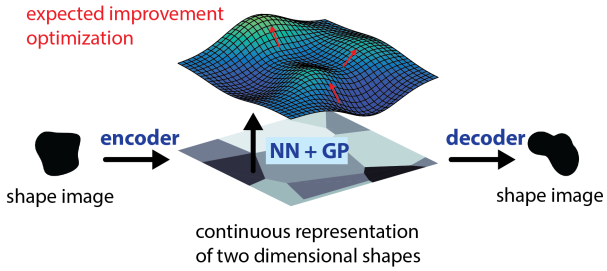

Figure 6: Schematic of the Bayesian optimization procedure to design drag-reduced shapes.

To illustrate the benefits of the joint autoencoder-GP framework we compare three different strategies for Bayesian optimization. In the first setting, we sample random points from the Gaussian prior distribution $p(z)$ and for each point optimize EI via gradient ascent. We contrast this with optimization working directly at the level of shapes. Inputs are random shapes generated in the same way as our training dataset. We consider training the GP directly on the input space ($GP(x)$) as well as including the intermediary mapping of the encoder ($GP(z(x))$). To be able to optimize EI via gradient ascent we relax the assumption of binary pixel values to continuous values in $[0, 1]$.

Figure 7 shows the best drag coefficient generated as a function of shape evaluations for the three strategies. In all cases we sample 600 starting points for gradient ascent and evaluate the resulting 100 shapes corresponding to the largest EI values in our simulator. All models share the same GP kernel function (squared-exponential), optimization parameters and stopping criteria.

Optimizing the acquisition function over the latent space consistently yields the largest reduction in drag coefficient for all rounds of Bayesian optimization. Optimizing the acquisition function over shapes does not improve upon directly simulating the drag coefficient for the random shapes which are chosen as gradient-ascent start points. The GP kernel is unable to extract drag-relevant features from the pixel input. As a consequence of the high-dimensionality of the pixel space, mean predictions for unknown shapes are close to the Gaussian prior and the EI gradient vanishes. We can recover part of the performance by using the 'deep kernel' of the encoder while still optimizing directly on shape images. We reason that the remaining performance gap is due to to the relaxation of continuous pixel values.

Another advantage of the latent-space optimization consists in the fact that we can generate gradient-ascent starting points by simply sampling from $z \sim \mathcal{N}(0, I)$. In contrast, optimization on the input space requires us to have access to new valid structures, i.e. shapes, as naive sampling in the $[0, 1]^{42\times56}$ pixel space breaks the optimization routine as expected.

Figure 8 shows examples of drag-reduced shapes over the course of 500 calls to the hydrodynamics simulator during optimization.

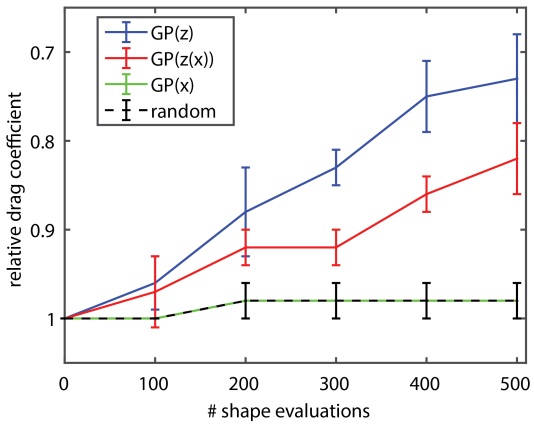

Figure 7: Best relative drag coefficient as a function of number of shape evaluations. The plot compares optimization based on random shapes (x) versus latent points (z). Error bars indicate standard deviation from three independent experiments.

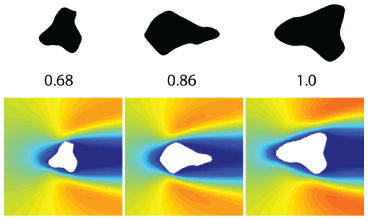

Figure 8: The Bayesian optimization routine produces shapes of reduced drag (< 1) compared to the smallest drag coefficient in the training set ($= 1$). Shape inputs to the hydrodynamic simulator (top) and corresponding flow fields (bottom).

5.3 ATTRIBUTE ADJUSTMENT ACROSS SHAPES

Given a set of shape designs with a real-valued attribute a our goal is for the decoder to learn how to generate a design with a given attribute value based on a latent coordinate z.

We consider two scalar attributes for our shape dataset: color and surface area. To introduce 'color' we create a separate channel for each shape which contains the binary mask multiplied by a random number from $[0, 1]$. During training we provide the true attribute value for each shape to the adversarial network and decoder. To make the learning more stable we slowly fade in the adversarial loss over half the number of maximal training epochs. Early stopping is only considered after this point.

At test time we take a given latent point and feed it along with a desired attribute value from the aforementioned range to the decoder. Figure 9 shows five example shapes decoded each with four different attribute values $[0.2, 0.4, 0.6, 0.8]$ resulting in a specific black-white intensity.

Adjusting color is a relatively easy task as the additionally introduced color channel is entirely orthogonal to the drag coefficient - the quantity our discriminative model $p_\Theta(y|z)$ promotes to be accurately encoded in the latent space. For this reason we consider the area of a shape as another more challenging attribute value to control for. Invariance to shape size requires the architecture to apply a non-trivial geometric transformation or to learn and store the shape information in the latent space in a more abstract way, e.g. through scale-invariant Fourier descriptors.

Figure 10 shows a selection of five latent points and their associated drag coefficients decoded with four different area values in analogy to the color-adjustment example.

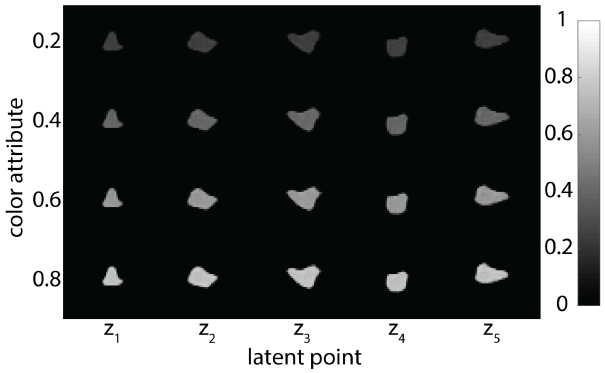

Figure 9: Shapes generated from five exemplary latent points $z_1 ... z_5$ and four different color-attribute values each.

The shapes are not cherry-picked. It is difficult to name an exact drag coefficient for the very small shapes due to the necessary shape rescaling, smoothing and boundary point extraction before any finite-element simulation can be performed. The simulations indicate that the drag coefficient of the scaled shapes stay within 25% of their original values with the largest deviations occurring at the smallest scale. At the same time area values change consistently for all shapes by about 300% (compare Table 3). The consistency of the scaling is remarkable as less than 4.5% of training shapes have an area smaller or larger than 67 and 175 pixels, respectively.

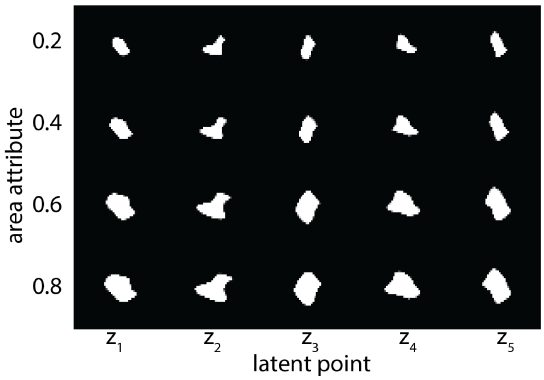

Figure 10: Shapes generated from five exemplary latent points $z_1 ... z_5$ and four different area-attribute values each.

6 RELATED WORK

Our approach draws on a broad basis of prior work and is generally related to conditional VAEs (Kingma et al., 2014) and hybrid models bridging generative and discrim-

Table 3: Pixel areas corresponding to shapes generated from latent points $z_1 ... z_5$ and four different area-attribute values each (compare Figure 10)

z_1	z_2	z_3	z_4	z_5
60	67	64	64	64
91	91	90	89	89
149	143	149	152	152
174	175	176	177	177

inative architectures (Maaløe et al., 2016; Shu et al., 2016; Kuleshov and Ermon, 2017). In contrast to the conditional VAE, we choose a label-independent encoder distribution stressing the perspective on the latent space as continuous embedding of inputs $\mathcal{X}$.

Using neural-network learned features as input to a GP model in the context of autoencoders dates back to Bengio et al. (2007). More recently, Snoek et al. (2012a) proposed non-parametric autoencoder guidance. Wilson et al. (2016a) present deep-kernel learning in which a stacked architecture of neural network and GP is trained jointly. Wilson et al. (2016b) and Huang et al. (2015) target the scalability of these hybrid GP models. Successful optimization within our framework depends on the ability to encode high-dimensional designs in a continuous latent space of lower dimensionality. This assumption is similar to the notion of low effective dimensionality in Wang et al. (2013) and related to Garnett et al. (2014). We considered joint training of GP and autoencoder in the case of the hydrodynamic dataset (5.2.2) but did not find this to outperform the sequential setting in which we train the GP on the latent space after parametric label guidance.

The idea of attribute adjustment draws on learning of fair representations, notably the fair VAE (Louizos et al., 2015). Purushotham et al. (2016) and recently Lample et al. (2017) explore enforcing invariance for adaptation in time-series data and natural images. We propose an adversarial objective based on mean-squared error maximization and demonstrate that attribute adjustment is feasible concurrent with Bayesian optimization.

GPs have been used for protein design and shape optimization - the two domains considered in our datasets. Romero et al. (2013) demonstrated the application of GPs to navigate the fitness landscape of proteins given limited data. The GP model is directly trained on amino acid sequences through a kernel function which relates sequence similarity to the spatial distance of amino acid positions in the folded protein structure. Previous works demonstrating the successful application of GP regression for the optimization of parametric designs in aerodynamic applications include (Simpson et al., 2001; Martin and Simpson, 2005; Jeong et al., 2005; Jouhaud et al., 2007). The aforementioned publications leverage domain knowledge for optimization. In contrast, the approach presented here is solely data-driven and based on deep-generative models.

7 DISCUSSION

Bayesian optimization is a data-efficient approach to optimize complex black-box functions without the need to supply gradients. Nevertheless discrete, high-dimensional input spaces pose a challenge to successful design optimization.

In this work we explore a framework combining variational autoencoding and Gaussian process regression to approach this challenge. We present a variational bound for the underlying graphical model and show how label-guidance enhances the predictive performance and incorporating unlabeled data leads to an increase in reconstruction accuracy.

We apply the optimization framework to the design of enhanced functional proteins. One round of Bayesian optimization proposes meaningful new protein variants which are free of known deleterious mutations.We further introduce a physics-based dataset of two-dimensional shapes and associated drag coefficients in laminar flow. Having access to a simulator allows us to perform closed-loop Bayesian optimization, such that after five rounds we improve by about 30% on the best value in the training set. We demonstrate that optimization based in the latent space outperforms optimization in the design space.

Finally, we consider an adversarial extension to our model. By forcing the latent space to be invariant w.r.t. an attribute value of choice, we are able to select this attribute value when decoding a latent point and impose it on our design. The combination of label-guidance and attribute-adversarial training shapes the information encoded in the latent space. We envision that the adversarial-model extension might be a fruitful approach to transfer functional groups or domains across molecules. In addition, the performance of discriminative models trained on the latent space could benefit when uninformative factors of variation are removed from the latent code based on domain knowledge.

Acknowledgements

The authors thank Jonathan Kuck, Neal Jean and Aditya Grover for fruitful discussions. This research was supported by TRI, NSF (#1651565, #1522054, #1733686) and a Hellman Faculty Fellowship.

References

R. Battistutta, A. Negro, and G. Zanotti. Crystal structure and refolding properties of the mutant F99s/M153t/V163a of the green fluorescent protein. *Proteins: Structure, Function, and Bioinformatics*, 41 (4):429–437, 2000.

Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.

D. M. Chudakov, M. V. Matz, S. Lukyanov, and K. A. Lukyanov. Fluorescent proteins and their applications in imaging living cells and tissues. *Physiological reviews*, 90(3):1103–1163, 2010.

J. Damborsky and J. Brezovsky. Computational tools for designing and engineering enzymes. *Current opinion in chemical biology*, 19:8–16, 2014.

R. Garnett, M. A. Osborne, and P. Hennig. Active learning of linear embeddings for Gaussian processes. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 230–239. AUAI Press, 2014. ISBN 0-9749039-1-4.

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, and T. Wu. Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Nature materials*, 15(10): 1120, 2016a.

R. Gómez-Bombarelli, D. Duvenaud, J. M. Hernández-Lobato, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *arXiv preprint arXiv:1610.02415*, 2016b.

A. Grover, T. Markov, P. Attia, N. Jin, N. Perkins, B. Cheong, M. Chen, Z. Yang, S. Harris, W. Chueh, and S. Ermon. Best arm identification in multi-armed bandits with delayed feedback. In A. Storkey and F. Perez-Cruz, editors, *Proceedings of Machine Learning Research*, volume 84, pages 833–842, Proceedings of Machine Learning Research, 2018. PMLR.

W. B. Huang, D. Zhao, F. Sun, H. Liu, and E. Y. Chang. Scalable Gaussian process regression using deep neural networks. In *IJCAI*, pages 3576–3582, 2015.

S. Jeong, M. Murayama, and K. Yamamoto. Efficient optimization design method using kriging model. *Journal of aircraft*, 42(2):413–420, 2005.

D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

J.-C. Jouhaud, P. Sagaut, M. Montagnac, and J. Laurenceau. A surrogate-model based multidisciplinary shape optimization method with application to a 2d subsonic airfoil. *Computers & Fluids*, 36(3):520–529, 2007.

A. D. Keefe and J. W. Szostak. Functional proteins from a random-sequence library. *Nature*, 410(6829):715, 2001.

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

D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

V. Kuleshov and S. Ermon. Deep hybrid models: Bridging discriminative and generative approaches. UAI, 2017.

G. Lample, N. Zeghidour, N. Usunier, A. Bordes, and L. Denoyer. Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems*, pages 5967–5976, 2017.

E. M. Lifshitz and L. D. Landau. *Course of theoretical physics, volume 6, fluid mechanics*. Pergamon Press, Oxford UK, 1959.

C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*, 2015.

L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

J. D. Martin and T. W. Simpson. Use of kriging models to approximate deterministic computer models. *AIAA journal*, 43(4):853–863, 2005.

D. G. Matthews, G. Alexander, M. Van Der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman. GPflow: A Gaussian process library using TensorFlow. *The Journal of Machine Learning Research*, 18(1):1299–1304, 2017.

M. McIntire, D. Ratner, and S. Ermon. Sparse Gaussian processes for Bayesian optimization. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 517–526, Jersey City, New Jersey, USA, 2016. AUAI Press. ISBN 978-0-9966431-1-5.

S. Purushotham, W. Carvalho, T. Nilanon, and Y. Liu. Variational recurrent adversarial deep domain adaptation. *openreview.net*, 2016.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2006.

P. A. Romero, A. Krause, and F. H. Arnold. Navigating the protein fitness landscape with Gaussian processes. *Proceedings of the National Academy of Sciences*, 110 (3):E193–E201, 2013.

K. S. Sarkisyan, D. A. Bolotin, M. V. Meer, D. R. Usmanova, A. S. Mishin, G. V. Sharonov, D. N. Ivankov, N. G. Bozhanova, M. S. Baranov, and O. Soylemez. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397, 2016.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

R. Shu, H. H. Bui, and M. Ghavamzadeh. Bottleneck conditional density estimation. *arXiv preprint arXiv:1611.08568*, 2016.

T. W. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree. Kriging models for global approximation in simulation-based multidisciplinary design optimization. *AIAA journal*, 39(12):2233–2241, 2001.

J. Snoek, R. P. Adams, and H. Larochelle. Nonparametric guidance of autoencoder representations using label information. *Journal of Machine Learning Research*, 13(Sep):2567–2588, 2012a.

J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012b.

M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

Z. Wang, M. Zoghi, F. Hutter, D. Matheson, and N. De Freitas. Bayesian optimization in high dimensions via random embeddings. In *IJCAI*, pages 1778–1784, 2013.

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016a.

A. G. Wilson, Z. Hu, R. R. Salakhutdinov, and E. P. Xing. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, pages 2586–2594, 2016b.