INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.156

On Diverse System-Level Design Using Manifold Learning and Partial Simulated Annealing

A. Cobb, A. Roy, D. Elenius, K. Koneripalli and S. Jha ✉

SRI International, United States of America
✉ susmit.jha@sri.com

Abstract

The goal in system-level design is to generate a diverse set of high-performing design configurations that allow trade-offs across different objectives and avoid early concretization. We use deep generative models to learn a manifold of the valid design space, followed by Monte Carlo sampling to explore and optimize design over the learned manifold, producing a diverse set of optimal designs. We demonstrate the efficacy of our proposed approach on the design of an SAE race vehicle and propeller.

Keywords: artificial intelligence (AI), engineering design, cyber-physical systems

1. Introduction

The automated design of systems is a long-standing goal of artificial intelligence (AI), and computer-aided design has been successfully used across a wide spectrum of applications, ranging from microprocessors to programs (Fujita, 2019). But this success is limited to domains where the design intent can be captured using complete and unambiguous specifications. We focus on the design of physical systems, which presents unique challenges beyond the scope of traditional design automation techniques. First, the design process lacks a complete formal characterization and relies on human intuition and domain expertise. The space of designs is large, but the designers have access to several examples of valid designs created for different performance objectives or functional goals. Designers use this knowledge and experience to identify a promising space of candidate designs and conduct manual design space exploration. Second, the design space exploration uses complex multiphysics models (Rider, 2013; Stolarski et al., 2018) spanning across several dimensions such as mechanical, electrical, and fluid-dynamics, and are non-differentiable, blackbox and proprietary. This renders direct use of combinatorial search or gradient-based methods for design inapplicable and necessitates minimizing the number of evaluated candidate-designs during exploration. Finally, the design process is often incremental and requires optimizing over multiple objectives (Schaltz & Soylu, 2011). Hence, it is not sufficient to view design as just an optimization process to find one optimal design; instead, designers create multiple diverse high-performing designs that trade-off different design objectives. This avoids early concretization and enables freedom to select designs for downstream integration and optimization when new objectives are added.
This paper aims to address these challenges and develop a machine learning approach to aid the physical design process, reducing the dependency on human intuition and experience, accelerating the discovery of new designs, and improving the performance and diversity of generated designs. Our primary contribution is the formulation of a two-staged approach, DeLPhy for design using learning focussed on physical system. DeLPhy uses examples of designs to learn a design manifold and simultaneously explores and optimizes designs meeting the specified objectives. We jointly train a

variational autoencoder to generate design examples with a specification network to predict design objectives. The designer specified the specification objective which is represented as the specification network. The following novel contributions in DeLPhy makes it an effective approach to physical design. We use temperature-guided sampling in the latent space of a variational autoencoder to implement a partial simulated annealing approach, whereby we optimize for the specified designs objectives. We use uncertainty-aware Monte Carlo sampling to avoid exploration in unreliable parts of the design manifold. Our sampling approach leads to the generation of a diverse set of designs (possibly, produced using different technologies) that trade-off multiple design objectives. This is critical to designing physical systems, which necessitates diversity to increase downstream adaptation

In Section 2, we describe the problem of physical system design and use an example to illustrate the key aspects of the problem. We also identify the requirements of a machine learning approach to physical design and present DeLPhy in Section 3 that meets these desiderata. We present three case studies to demonstrate the effectiveness of our approach in Section 4 and discuss related work in Section 5. We conclude in Section 6 by summarizing our key findings.

2. Physical System Design

The design freedom for a specific application can be parameterized to define a design space that needs to be explored. We expect multiple competing design objectives that need to be achieved simultaneously. The evaluation of these objectives requires the evaluation of domain models, which are typically slow and computationally expensive multiphysics models. Each point in the design space is called a design configuration. We are given a set of exemplar designs that are valid design configurations but do not address the design objectives under consideration. The goal is to use these exemplar designs to learn a manifold in the design space over which we can explore and identify a diverse set of optimal design configurations that trade off different design objectives.

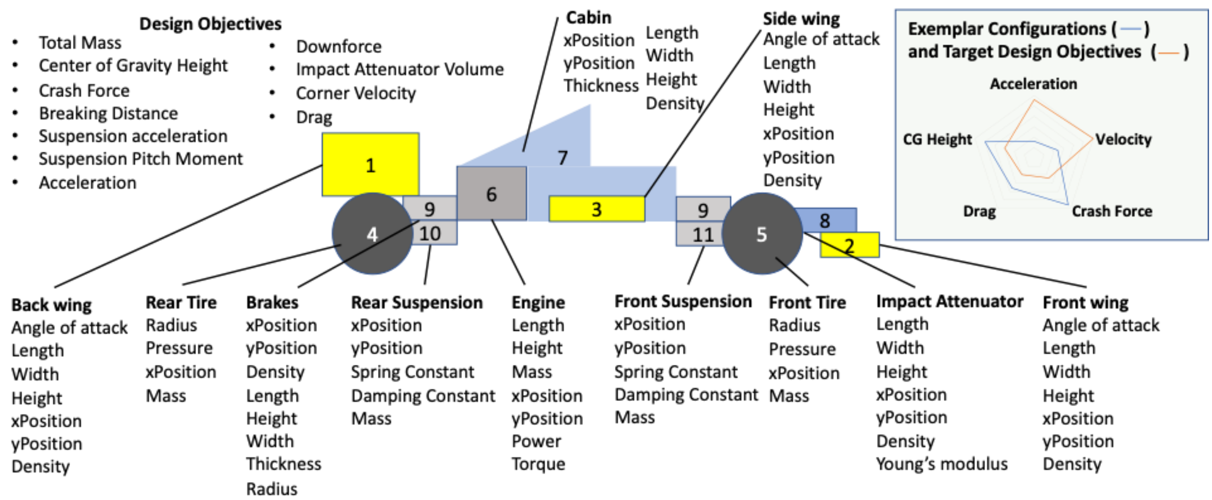

Figure 1. Physical design problem for the formula SAE racing vehicle (Soria Zurita et al., 2018)

Figure 1 illustrates the physical design problem using the example of the Formula SAE racing vehicle from systems engineering literature (Soria Zurita et al., 2018). The design of a Formula SAE racing vehicle comprises 11 subsystems such as the tires, suspensions, engine, cabin, impact attenuator and wings. Each subsystem is described using multiple parameters listed above. For example, the tire components have radius, pressure, x position and mass as parameters. There are 39 parameters that describe the vehicle's design space. A more detailed design could consider higher dimensional description, such as the 3D shape of the wings, to better estimate wind-drag. The design objectives capture the designer's underlying preferences for the system. The 11 objectives listed above can be a mixture of target performance such as the preferred height of the center of gravity, and optimization metrics such as maximizing acceleration and minimizing drag. While some of these objectives could be analytical domain models, the accurate computation of quantities such as drag requires slow

https://doi.org/10.1017/pds.2022.156 Published online by Cambridge University Press

blackbox proprietary software (Rider, 2013). The slow physics domain models need to be approximated by faster surrogate specification models that can allow more efficient exploration and optimization. Further, a possible spread of performance over the objectives of the exemplar vehicle designs is illustrated in the radar plot in the top right corner in blue. The performance of a target vehicle design configuration is shown in red. The target requires us to have much higher velocity and acceleration while reducing drag, height of center of gravity and crash force. Designs must also be adaptable, that is, new metrics might be added later and hence, it is critical to generate not just one optimal design but a number of diverse designs that trade off different design objectives and enable future adaptation to new metrics.

3. Physical System Design Using Machine Learning

In this paper, we develop a two-stage approach DeLPhy (illustrated in Figure 2) for **de**sign using **l**earning focused on **phy**sical system. We denote the design space by X (structural parameters) with candidate designs $x \in X$ and the specification of the design objectives (functional and behavioural parameters) by s which is a vector of competing multidimensional objectives s_i expressing the performance of the target design. In the first stage, we use exemplar designs to learn a generative model in the form of a variational autoencoder (VAE) (Kingma & Welling, 2013; Rezende et al., 2014) over the design space, along with a specification network that predicts the values of the different design objectives from the latent representation. The latent design space is denoted by $z \in Z$. The encoder network μ, $\log \sigma^2 = E_\theta(x)$ and $z = \mu + \epsilon\sigma$ with parameters θ maps a design to latent space and $\epsilon \sim N(0, 1)$is the VAE reparameterized noise. The decoder of the VAE is represented by $\bar{x} = D_\phi(z)$, where ϕ are the parameters of the decoder. The output of the decoder is a design (structural parameters). The specification network $s = S_\mu(z)$ with model parameters μ predicts the design objectives as a function of the latent design z.

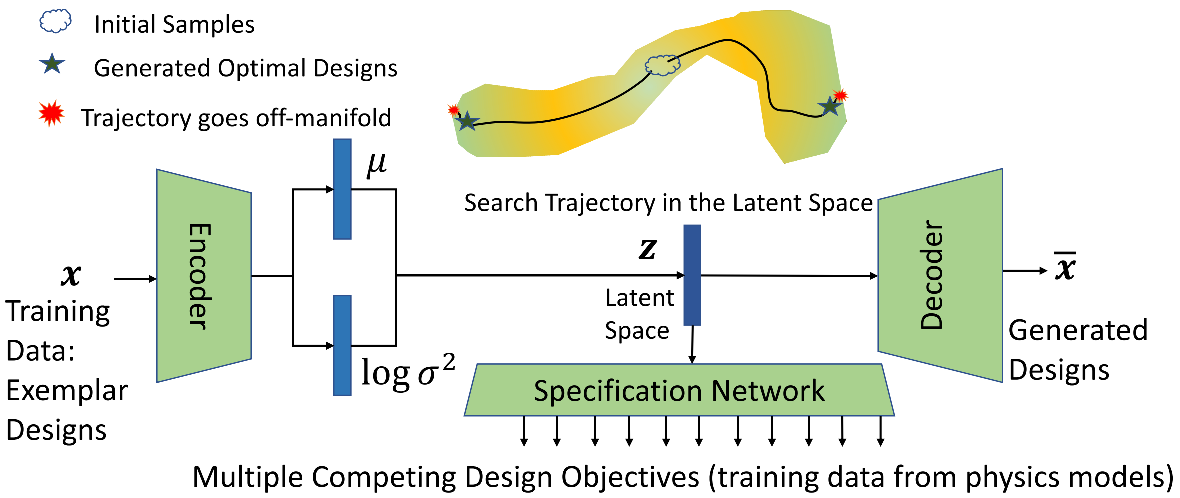

Figure 2. DeLPhy uses exemplar designs to learn a variational encoder (VAE) where the decoder is trained with uncertainty quantification. The latent space represents the learned design manifold. The specification network predicts the design objectives from the latent space. The VAE and the specification network are jointly trained on the exemplar designs and their evaluation on physics models. In the design exploration stage, we condition on the new target design objectives and use temperature annealed sampling the latent space, moving towards optimal designs exploiting the gradient information. Further, Monte Carlo sampling in the decoder leads to multiple design samples for a sampled latent design, which are then passed through the encoder and the specification network to determine a distribution over the design objectives. High variance/uncertainty implies off-manifold designs that may not be unrealizable. DeLPhy finds multiple diverse optimal design configurations.

https://doi.org/10.1017/pds.2022.156 Published online by Cambridge University Press

Training Model. Training the VAE and the specification network models can be done offline without a full knowledge of the target design objectives, and can be reused for different design problems. Since the exemplar designs only need to be valid but not address design objectives, we can generate them by sampling configurations from a simple distribution and evaluating the valid configurations using the physics models. The learned generative model along with the specification network interpolates the design objectives over the configurations and thus, also serves as a differentiable surrogate model minimizing the evaluation of the slow physics models. We train the generative model with Monte Carlo dropout (Gal & Ghahramani, 2016) over the decoder network to make the model uncertainty-aware and enable us to compute the confidence on our predicted design performance. The encoder, decoder, and the specification network are jointly trained using the following loss function where we use a variant of the standard VAE evidence lower bound (ELBO) loss called the generalized ELBO with constrained optimization (Rezende & Viola, 2018).

$$L_\lambda(\theta,\phi,\mu) = E_{\rho(x)}\left[D_{KL}\left(E_\theta(z|x)\,||\,\pi(z)\right)\right] + \lambda^T\left(E_{\rho(x)\,E_\theta(Z|x)}\left[MSE\left(x, D_\phi(z)\right) + MSE\left(s, S_\mu(z)\right)\right]\right) \quad (1)$$

This variant allows directly controlling the balance between compression (KL minimization) and the other constraints we wish to enforce in our model (reconstruction error and the accuracy of the specification network). We use a mixture of Gaussian prior $\pi(z)$ and the MSE loss, but MSE can be replaced with any other error characterization. The loss $L_\lambda(\theta,\phi,\mu)$, with λ as the Lagrange multiplier, is computed using a sampling distribution $\rho(x)$ and it is minimized to obtain the network parameters θ,ϕ,μ using the standard method of Lagrange multipliers (Bertsekas, 2014).

This offline first stage in DeLPhy is followed by the second stage of exploration over the design configurations and optimization of the specific target objectives. We use Monte Carlo (MC) (Duane et al., 1987; Neal et al., 2011) sampling with a novel temperature scaling in the latent space of the VAE to implement a partial simulated annealing approach, whereby we optimize for certain performance objectives while conditioning on meeting the required specifications. This enables exploration of the Pareto frontier of the multi-objective design optimization problem and yields a diverse set of design configurations. We use partial simulated annealing where each objective has its own annealing schedule. For each annealing schedule, the temperature at step k is $T(k) = T_0 e^{-\lambda k}$, where T_0 is the initial temperature and λ is the annealing rate. At high temperatures, $T > 1$, the energy gap between the subsequent proposals is reduced, which results in a higher chance of the Metropolis-Hastings step accepting moves to the regions of the space with lower probability. This favors more exploration and enables us to traverse low probability regions.

As T increases, we encourage accepting samples in the regions of high probability. Given a set of multiple objectives, we can treat these asymmetrically during exploration by using a different temperature annealing schedule for each of the objectives, favoring conditioning on some target values while trying to optimize over the others. This makes the approach partial annealing since some objectives are optimized via annealing while the remaining continue to be sampled conditioned on the target objective performances.

Since we are optimizing over a surrogate model, optimization can drive the model out of its training distribution and the predicted values of the design objective on some apparent promising configurations will not match their real values. We can run the slow physics models to detect such errors, but we would like to minimize such a possibility by making our generative model and the specification network uncertainty-aware. We accomplish this without hurting the scalability of our method using Monte Carlo dropout (Gal & Ghahramani, 2016) over the decoder network of the generative model, which allows us to quantify uncertainty in the predicted values of the design objectives. The MC dropout in the decoder is used to sample reconstructions that are passed through the specification network to compute the uncertainty:

$$Uncertainty(z) = Variance\left(\left\{s = S_\mu\left(E_\theta(\bar{x})\right) \,\middle|\, \bar{x} \in MCSample\left(D_\phi(z)\right)\right\}\right) \quad (2)$$

Therefore, rather than deciding on whether a design proposed by the generative model is likely in the design space, we use the specification network to determine the reliability of the design. By focusing

https://doi.org/10.1017/pds.2022.156 Published online by Cambridge University Press

on the regions where the objectives can be predicted with low uncertainty, DeLPhy is able to avoid high uncertainty regions of the manifold and find diverse designs which have high confidence of retaining optimal performance when evaluated against slow but more accurate models. This uncertainty measures the reliability of the design process and does not correspond to the robustness of the design against environmental perturbations.

4. Case Studies

We demonstrate how DeLPhy can be used to generate physical designs using three case studies: propeller design, SAE race vehicle design, and an air vehicle design. In our case studies, we examine the following research questions:

1. Can DeLPhy find valid design configurations for given design objectives?
2. Can DeLPhy detect when the generated designs are unreliable?
3. Do designs generated by DeLPhy exhibit high diversity?

4.1. Propeller Design

Propellers are key components in a range of vehicle classes including aircraft, ships, and underwater vehicles. A propeller design configuration is defined by its geometric properties such as the number of blades, diameter of the propeller, shape and pitch of the blades, and hub diameter. The performance metrics of a propeller include thrust, rotation speed, required torque, and efficiency. In this design problem, we look to trade off velocity and efficiency, whereby the challenge is to design an efficient propeller that operates at low velocities. To build and evaluate the performance of our proposed designs, we use OpenProp (DMS, 2021; Epps et al., 2009) - an open-source tool that is widely used in academia and industry and implements relevant physics models. Figure 3 compares the distribution of the two competing objectives - velocity and the propeller efficiency when sampled using a Gaussian prior in the latent space, and those generated using DeLPhy. The y-axis represents density of the design distribution. The designs from DeLPhy have high efficiency even at low velocity.

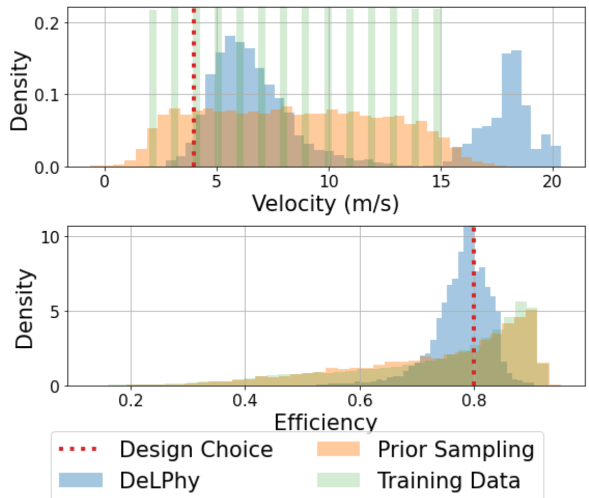

Figure 3. DeLPhy can find propellers with high efficiency at relatively low speed

Figure 4 (left), shows the sample trajectories of the velocity and the propeller efficiency, as well as the corresponding variance on the objectives. Around sample ID 9000, we see high velocities with high efficiency, but the corresponding variance is high, suggesting these are unreliable designs. Figure 4 (right) shows that our predicted high uncertainty area align with high deviation from the detailed model (true error) and we are also able to identify cases (black points) when the OpenProp physics model actually failed to produce a valid output. Physics models also have implicit assumptions on their inputs to converge to a valid output. Thus, we are able to avoid unreliable designs using uncertainty quantification in DeLPhy. Several diverse propeller designs were produced by DeLPhy

https://doi.org/10.1017/pds.2022.156 Published online by Cambridge University Press

with the same objective of efficiency higher than 75% and velocity lower than 4.5 m/s. The propellers have a different number of blades, shape and pitch of the blades, and hub diameter.

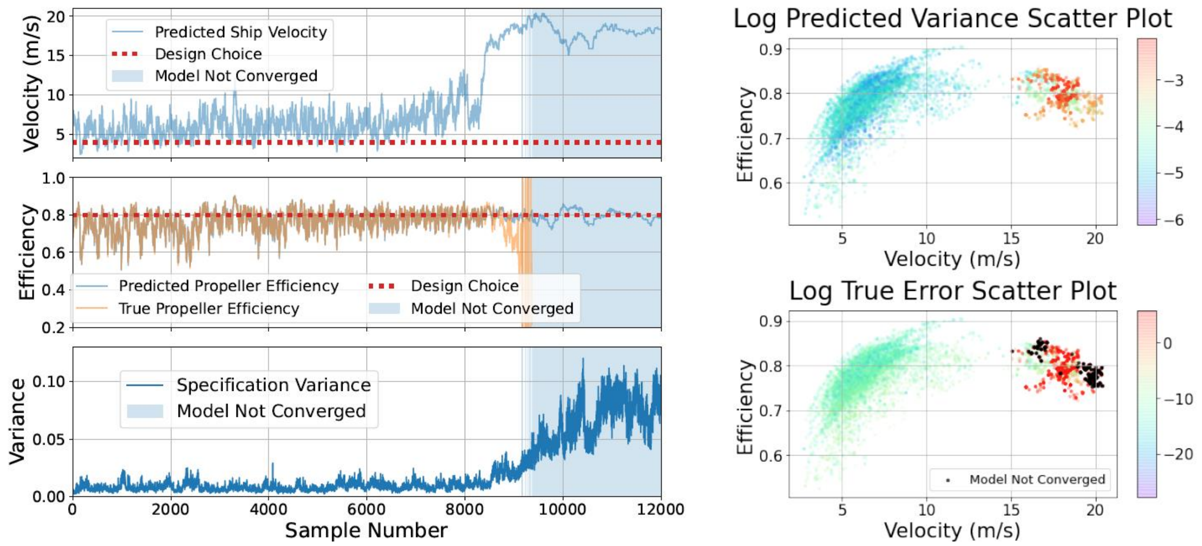

Figure 4. Uncertainty quantification detects unreliable designs during exploration. Left: Increased deviation of objective functions from OpenProp physics model is detected by DeLPhy. Right: High true error or physics model failure corresponds to high uncertainty predicted by DeLPhy

4.2. SAE Race Vehicle Design

The second case study is the SAE race vehicle (Stolarski et al., 2018) described in Section 2 and Figure 1. The design objective is to build a vehicle of a specific weight, with a high cornering velocity which is challenging as maximum cornering velocity decreases with the mass of the vehicle. DeLPhy samples designs that converge around the design objectives even though the training data and the prior distribution are far from it. DeLPhy is able to detect when the sampling trajectory goes off-manifold and generates unreliable designs. The deviation of the predicted objectives diverges from the true value after some number of samples but DeLPhy can detect this as the variance of the specification output also rapidly increases. Figure 5 shows that the race vehicle designs created by DeLPhy exhibits significant diversity in the choice of key components: engine, tires, and brakes.

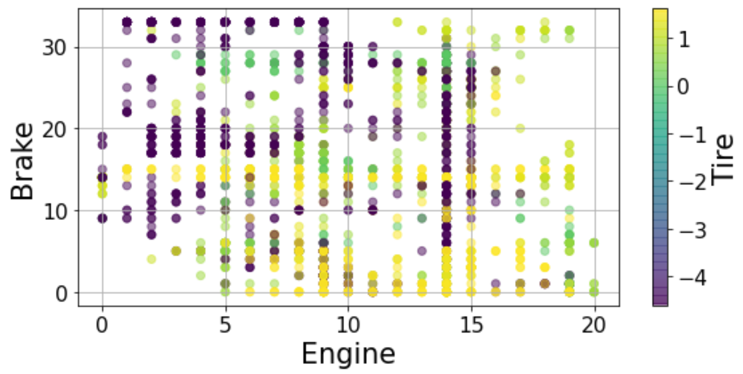

Figure 5. DeLPhy generates vehicles with different engines, tires and brakes (color denotes the third dimension)

5. Related Work

The use of deep generative modelling for computer aided design is a relatively recent research frontier (Seff et al., 2021; Xu et al., 2021; Zhao et al., 2020). These approaches target certain aspects of design such as geometry while we focus on system-level design. Some recent work (Tripp et al., 2020;

https://doi.org/10.1017/pds.2022.156 Published online by Cambridge University Press

Sanchez-Lengeling & Aspuru-Guzik, 2018) combine the latent space representation of generative models as part of their Bayesian optimization (Shahriari et al., 2015) algorithms. One particular approach relevant to our work is by Notin et al. (2021), where they derive an importance sampling estimator of the mutual information to indicate uncertainty in the latent space for discrete data. In our work, DeLPhy samples over the latent space to simultaneously explore and optimize to ensure diversity. Further, we develop an uncertainty quantification approach that takes into account the variance of the specification network predicting the design objectives. This ensures DeLPhy can avoid exploring design configurations where the predicted performance cannot be trusted. Machine learning methods have also been used for drug discovery and molecule design (Brookes & Listgarten, 2018; Brookes et al., 2019). These approaches have impressive results on solving complex combinatorial optimization problems. In our work, we are focused on the design of diverse physical systems with both continuous and discrete components, and with multiple design objectives which have to be satisfied simultaneously. Surrogate-based optimization is widely explored in design optimization, where the goal is to learn a surrogate function to replace often expensive black-box simulators e.g., computational fluid dynamics simulators (Koziel et al., 2011; Han et al., 2012; Viquerat et al., 2021). The surrogate function aims to capture the physical properties of the design environment and reliably evaluate design samples. These approaches tend to be more scalable compared to the black-box optimization approaches (Belakaria et al., 2020; Deshwal et al., 2021) by avoiding the expensive black-box evaluation during optimization. Further, if the surrogate function is differentiable e.g., a neural network, the gradients are also available to the optimizer to perform an end-to-end optimization Grabocka et al. (2019); Liu et al. (2020); Sun et al. (2021). Our proposed method can leverage advances in better surrogate modeling for more efficient exploration. In contrast to existing methods, the design for physical systems needs to find a diverse set of designs that trade off different objectives and allow further downstream adaptation to new design objectives.

6. Conclusion

Design of a physical system for a given set of design objectives requires domain expertise and creativity. System designers use their experience and knowledge about previous designs to propose new solutions. The challenge of using machine learning for physical design requires a combination of uncertainty-aware extrapolation from existing designs to new design configurations, and efficient exploration and optimization to identify diverse optimal designs. DeLPhy presented in this paper addresses these challenges. DeLPhy comprises two stages. The first is an offline stage of learning the design manifold using a variational autoencoder which is trained to be uncertainty-aware using Monte Carlo dropout in the decoder network. We also jointly learn a specification network to predict the design objectives from the latent space, which helps replace slow domain models with faster differentiable neural network surrogates. The second stage uses partial simulated annealing over the latent space of the autoencoder to explore the design manifold and optimize the design objectives, generating a diverse set of optimal designs. DeLPhy was demonstrated on three case studies involving the design of an SAE race vehicle, a propeller, and an air vehicle. First, DeLPhy is shown to be able to sample designs with objectives which are very different from the original exemplar designs used in learning. Second, DeLPhy uses uncertainty awareness to detect when the predictions of the surrogate model cannot be trusted and thus, enables it to avoid designs that are not realistic. Finally, DeLPhy finds a diverse set of optimal designs in each of the three case studies. This work is a first step towards leveraging deep learning to aid the design of physical systems.

References

Belakaria, S., Deshwal, A., Jayakodi, N., and Doppa, J. 2020. Uncertainty-aware search framework for multi-objective Bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 10044–10052).

Bertsekas, D. 2014. *Constrained optimization and Lagrange multiplier methods*. Academic press.

Brookes, D., and Listgarten, J. 2018. Design by adaptive sampling. *arXiv preprint arXiv:1810.03714*.

Brookes, D., Park, H., and Listgarten, J. 2019. Conditioning by adaptive sampling for robust design. In *International Conference on Machine Learning* (pp. 773–782).

Deshwal, A., Belakaria, S., and Doppa, J. 2021. Bayesian Optimization over Hybrid Spaces. *arXiv preprint arXiv:2106.04682*.
DMS 2021 Ship Design Tools At DMS. OpenProp - integrated rotor design and analysis, 2021. URL: https://dmsonline.us/openprop/
Duane, S., Kennedy, A., Pendleton, B., and Roweth, D. 1987. Hybrid Monte Carlo. *Physics letters B*, *195*(2), p.216–222.
Epps, B., Chalfant, J., Kimball, R., Techet, A., Flood, K., and Chryssostomidis, C. 2009. OpenProp: An open-source parametric design and analysis tool for propellers. In *Proceedings of the 2009 grand challenges in modeling & simulation conference* (pp. 104–111).
Fujita, M. 2019. Basic and advanced researches in logic synthesis and their industrial contributions. In *Proceedings of the 2019 International Symposium on Physical Design* (pp. 109–116). https://doi.org/10.1145/3299902.3311069
Gal, Y., and Ghahramani, Z. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning* (pp. 1050–1059). https://dl.acm.org/doi/10.5555/3045390.3045502
Grabocka, J., Scholz, R., and Schmidt-Thieme, L. 2019. Learning surrogate losses. *arXiv preprint arXiv:1905.10108*.
Han, Z.H., Zhang, K.S., and others 2012. Surrogate-based optimization. *Real-world applications of genetic algorithms*, *343*.
Kingma, D., and Welling, M. 2013. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*.
Koziel, S., Ciaurri, D., and Leifsson, L. 2011. Surrogate-based methods. In *Computational optimization, methods and algorithms* (pp. 33–59). Springer.
Leimkuhler, B., and Reich, S. 2005. Simulating Hamiltonian Dynamics. In *Cambridge Monographs on Applied and Computational Mathematics* (pp. i-iv). Cambridge University Press.
Liu, L., Wang, M., and Deng, J. 2020. A unified framework of surrogate loss by refactoring and interpolation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16* (pp. 278–293).
Neal, R. 2011. MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, *2*(11), p.2.
Notin, P., Hernández-Lobato, J., and Gal, Y. 2021. Improving black-box optimization in VAE latent space using decoder uncertainty. *arXiv preprint arXiv:2107.00096*.
Rezende, D., and Viola, F. 2018. Generalized ELBO with constrained optimization, geco. In *Workshop on Bayesian Deep Learning, NeurIPS*.
Rezende, D., Mohamed, S., and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning* (pp. 1278–1286). https://doi.org/10.5555/3044805.3045035
Rider, M. 2013. *Designing with Creo Parametric 2.0*. SDC Publications.
Sanchez-Lengeling, B., and Aspuru-Guzik, A. 2018. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, *361*(6400), p.360–365.
Schaltz, E., and Soylu, S. 2011. Electrical vehicle design and modeling. *Electric vehicles-modelling and simulations*, *1*, p.1–24.
Seff, A., Zhou, W., Richardson, N., and Adams, R. 2021. Vitruvion: A Generative Model of Parametric CAD Sketches. *arXiv preprint arXiv:2109.14124*.
Shahriari, B., Swersky, K., Wang, Z., Adams, R., and De Freitas, N. 2015. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, *104*(1), p.148–175. https://dx.doi.org/10.1109/JPROC.2015.2494218
Soria Zurita, N., Colby, M., Tumer, I., Hoyle, C., and Tumer, K. 2018. Design of complex engineered systems using multi-agent coordination. *Journal of Computing and Information Science in Engineering*, *18*(1), p.011003. https://doi.org/10.1115/1.4038158
Stolarski, T., Nakasone, Y., and Yoshimoto, S. 2018. *Engineering analysis with ANSYS software*. Butterworth-Heinemann. https://doi.org/10.1016/C2016-0-01966-6
Sun, X., Xue, T., Rusinkiewicz, S., and Adams, R. 2021. Amortized Synthesis of Constrained Configurations Using a Differentiable Surrogate. *arXiv preprint arXiv:2106.09019*.
Tripp, A., Daxberger, E., and Hernández-Lobato, J. 2020. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, *33*.
Viquerat, J., Rabault, J., Kuhnle, A., Ghraieb, H., Larcher, A., and Hachem, E. 2021. Direct shape optimization through deep reinforcement learning. *Journal of Computational Physics*, *428*, p.110080.
Xu, J., Spielberg, A., Zhao, A., Rus, D., and Matusik, W. 2021. Multi-Objective Graph Heuristic Search for Terrestrial Robot Design. *arXiv preprint arXiv:2107.05858*.
Zhao, A., Xu, J., Konakovi\'c-Lukovi\'c, M., Hughes, J., Spielberg, A., Rus, D., and Matusik, W. 2020. RoboGrammar: graph grammar for terrain-optimized robot design. *ACM TOG*, *39*(6), p.1–16. https://dl.acm.org/doi/abs/10.1145/3414685.3417831