PHYSICAL SYSTEM DESIGN USING HAMILTONIAN MONTE CARLO OVER LEARNED MANIFOLDS

Anonymous authors
Paper under double-blind review

ABSTRACT

The design of complex physical systems entails satisfying several competing performance objectives. In practice, some design requirements are often implicit in the intuition and knowledge of designers who have many years of experience working with similar designs. Designers use this experience to sample a few promising candidates in the design space and evaluate or simulate them using detailed, typically slow multiphysics models. The goal in design is usually to generate a diverse set of high-performing design configurations that allow trade-offs across different objectives and avoid early concretization. In this paper, we develop a machine learning approach to automate physical system design. We use deep generative models to learn a manifold of the valid design space, followed by Hamiltonian Monte Carlo (HMC) with simulated annealing to explore and optimize design over the learned manifold, producing a diverse set of optimal designs. Our approach is akin to partial simulated annealing restricted to the learned design manifold, where the annealing schedule is varied to trade-off different objectives. To prevent our approach from traversing off the design manifold and proposing unreliable designs, we leverage Monte Carlo dropout as a way to detect and avoid design configurations where the learned model cannot be trusted. We demonstrate the efficacy of our proposed approach using several case studies that include the design of an SAE race vehicle, propeller, and air vehicle. Across these case studies, we successfully show how our method generates high-performing and diverse designs.

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; Solar-Lezama, 2008). 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 a 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 following novel contributions in DeLPhy makes it an effective approach to physical design:

- We use Hamiltonian Monte Carlo (HMC) with a novel temperature scaling 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 Monte Carlo Dropout to make the prediction of performance objectives uncertainty-aware, and detect new unreliable design configurations. This avoids exploration in unreliable parts of the design manifold.
- Our sampling approach leads to the generation of a diverse set of designs that trade-off multiple design objectives. This is critical to designing physical systems, which necessitates diversity to increase adaptation to downstream design.

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 desiderata 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.

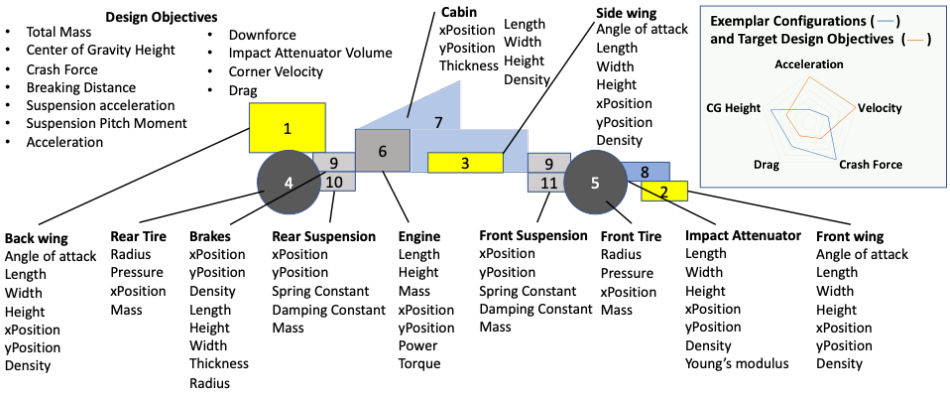


Figure 1: Physical Design Problem for the Formula SAE racing vehicle (Soria Zurita et al., 2018)

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 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. We identify the following desiderata that need to be satisfied by an effective machine learning approach for the design of physical systems.

- The exemplar designs are likely to be far from optimal design objectives, but help identify the design space constraints. This implicit design space must be learned from the example designs.
- Since the target performance objectives can be very different from those of the exemplar designs, the learned models need to be uncertainty-aware and detect when its predictions on new configurations cannot be trusted to avoid exploration of configurations that are likely to be unrealizable.
- A diverse set of optimal designs need to be identified to trade off different objectives, and enable future adaptation to additional objectives. We need an exploration approach that can sample diverse designs that meet the design objectives in addition to optimization.

3 Physical System Design using Machine Learning

In this paper, we develop a two-stage approach `DeLPhy` (illustrated in Figure 2) for design using learning focussed on physical system that satisfies the desiderata identified in the previous section.

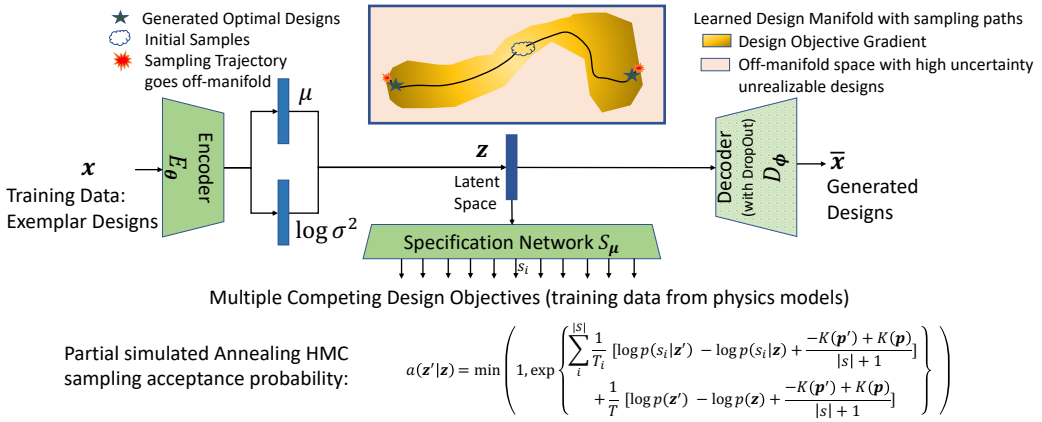


Figure 2: `DeLPhy` uses exemplar designs to learn a variational encoder (VAE) where the decoder is trained with dropout. The latent space represents the learned design manifold. The specification network predicts the design objectives from the latent space. The VAE and the specification network are jointly 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 HMC to sample the latent space, moving towards optimal designs exploiting the gradient information. Further, Monte Carlo dropout 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.

We denote the design space by $\mathcal{X}$ with candidate designs $\mathbf{x} \in \mathcal{X}$ and the specification of the design objectives by $\mathbf{s}$ 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 $\mathbf{z} \in \mathcal{Z}$. The encoder network $\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2 = E_{\boldsymbol{\theta}}(\mathbf{x})$ and $\mathbf{z} = \boldsymbol{\mu} + \epsilon\boldsymbol{\sigma}$ with parameters $\boldsymbol{\theta}$ maps a design to latent space and $\epsilon \sim \mathcal{N}(0, 1)$ is the VAE reparameterized noise. The decoder of the VAE is represented by $\overline{\mathbf{x}} = D_{\boldsymbol{\phi}}(\mathbf{z})$, where $\boldsymbol{\phi}$ are the parameters of the decoder. The specification network $\mathbf{s} = S_{\boldsymbol{\mu}}(\mathbf{z})$ with model parameters $\boldsymbol{\mu}$ predicts the design objectives as a function of the latent design $\mathbf{z}$.

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).

$$\mathcal{L}_\lambda(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\mu}) = \mathbb{E}_{\rho(\mathbf{x})}[D_{\mathrm{KL}}(E_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})||\pi(\mathbf{z}))] + \lambda^T(\mathbb{E}_{\rho(\mathbf{x})E_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}[\texttt{MSE}(\mathbf{x}, D_{\boldsymbol{\phi}}(\mathbf{z})) + \texttt{MSE}(\mathbf{s}, S_{\boldsymbol{\mu}}(\mathbf{z}))])$$

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 $\pi(\mathbf{z})$ and the `MSE` loss, but `MSE` can be replaced with any other error characterization. The loss $\mathcal{L}_\lambda(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\mu})$, with λ as the Lagrange multiplier, is computed using a sampling distribution $\rho(\mathbf{x})$ and it is minimized to obtain the network parameters $\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\mu}$ using the standard method of Lagrange multipliers (Bertsekas, 2014).

Temperature annealed HMC exploration. 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 Hamiltonian Monte Carlo (HMC) (Duane et al., 1987; Neal et al., 2011) with a novel temperature scaling in the latent space of the VAE to implement a partial simulated annealing approach, whereby we optimize for certain objectives while conditioning on a baseline performance with respect to other objectives. HMC is a gradient-based MCMC approach that is well-suited to the exploration of high dimensional distributions. It is especially suited to our problem, as the specification network can be differentiated with respect to the latent space. This enables exploration of the Pareto frontier of the multiobjective design optimization problem and yields a diverse set of design configurations. Formally, for a target performance $\mathbf{s}$, we aim to explore and find optimal designs by sampling from $p(\mathbf{z}|\mathbf{s})$ following by decoding to a design $\overline{\mathbf{x}} = D_{\boldsymbol{\phi}}(\mathbf{z})$.

In order to formulate a temperature annealed HMC, we first define a potential energy function $U(\mathbf{z})$ to characterize the negative log-likelihood of the latent design given a target objective.

$$\log p(\mathbf{z}|\mathbf{s}) \propto \sum_i \log p(s_i|\mathbf{z}) + \log p(\mathbf{z}) = -U(\mathbf{z})$$

We augment our system to include a vector, $\mathbf{p}$, of the momentum and introduce a kinetic energy term in the usual form $K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T\mathbf{M}^{-1}\mathbf{p}$. This form of $K(\mathbf{p})$ corresponds to the negative log probability of the zero mean Gaussian distribution with covariance matrix (symmetric, positive definite) $\mathbf{M}$. We can choose other forms of kinetic energy which would lead to different exploration pattern. Through the introduction of the kinetic energy term, we have built a Hamiltonian system from which we can sample using Hamiltonian dynamics according to HMC (Duane et al., 1987; Neal et al., 2011). Our canonical distribution can be written as:

$$p(\mathbf{z}, \mathbf{p}) = \frac{1}{Z}\exp\left(\frac{-U(\mathbf{z})}{T}\right)\exp\left(\frac{-K(\mathbf{p})}{T}\right)$$

where Z is the normalizing constant and T is the temperature. The total energy $H(\mathbf{z}, \mathbf{p}) = U(\mathbf{z}) + K(\mathbf{p})$. In HMC, we sample new values for momentum $\mathbf{p}$ using Gaussian distribution dependent on $\mathbf{M}$. This is followed by a Metropolis update using Hamiltonian dynamics to propose a new state. We

simulate the Hamiltonian dynamics using the Leapfrog method (Leimkuhler & Reich, 2005) and the computed proposed state is accepted as the next state $(\mathbf{z}', \mathbf{p}')$ of the Markov chain with probability:

$$a(\mathbf{z}'|\mathbf{z}) = \min\left(1, \exp\left\{\frac{-H(\mathbf{z}', \mathbf{p}') + H(\mathbf{z}, \mathbf{p})}{T}\right\}\right)$$

We use a new approach of partial simulated annealing where each objective has its own annealing schedule. For each annealing schedule, the temperature at step k is $T(k) = T_0 \mathrm{e}^{-\lambda k}$ 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. The corresponding acceptance probability is:

$$a(\mathbf{z}'|\mathbf{z}) = \min\left(1, \exp\left\{\sum_i^{|\mathbf{s}|} \frac{1}{T_i}\left[\log p(s_i|\mathbf{z}') - \log p(s_i|\mathbf{z}) + \frac{-K(\mathbf{p}') + K(\mathbf{p})}{|\mathbf{s}|+1}\right] + \frac{1}{T}\left[\log p(\mathbf{z}') - \log p(\mathbf{z}) + \frac{-K(\mathbf{p}') + K(\mathbf{p})}{|\mathbf{s}|+1}\right]\right\}\right)$$

For the objectives which are not optimized, we have a fixed T_i and for the objectives being optimized, we select different annealing schedules $T_i(k) = T_0 \mathrm{e}^{-\lambda_i k}$ to drive the optimization towards diverse designs with different trade-offs over the design objectives (more details in Appendix 1).

Uncertainty quantification in `DeLPhy`. 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:

$$\texttt{Uncertainty}(\mathbf{z}) = \texttt{Variance}(\{\mathbf{s} = S_{\boldsymbol{\mu}}(E_{\boldsymbol{\theta}}(\overline{\mathbf{x}})) \mid \overline{\mathbf{x}} \in \texttt{MCDroput}(D_{\boldsymbol{\phi}}(\mathbf{z}))\})$$

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 focussing 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.

Illustration of `DeLPhy`. Before demonstrating the effectiveness of `DeLPhy` in physical design, we use the simple MNIST dataset (LeCun et al., 1998) to provide readers with the intuition on the key aspects of `DeLPhy`. Figure 3 demonstrates diversity-preserving sampling in `DeLPhy` conditioned on the digit being '2' while using partial annealing HMC to reduce the thikness of the digit. The variance in the thickness specification of the MNIST digit sharply rises at sample 4000 and the generated images (IDs 4500 and 6000) are unrecognizable as digits. The digits with IDs greater than 4000 could be misinterpreted as valid digits with low thickness; however, the uncertainty quantification in `DeLPhy` prevents such errors.

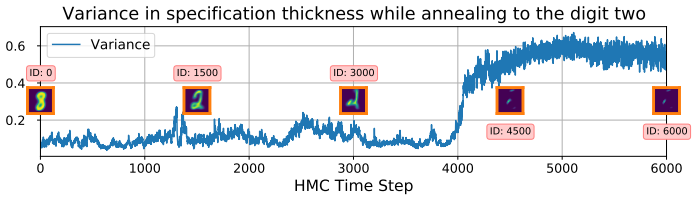


Figure 3: Variance in the predicted thickness while annealing. After sample ID 4000, the variance rapidly rises, indicating unreliability - IDs 4500 and 6000 are unrecognizable as digits.

4 EXPERIMENTS

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 uncertainty quantification in 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. Further experimental details are described in Appendix B and C. Figure 4 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 designs from DeLPhy have high efficiency even at low velocity.

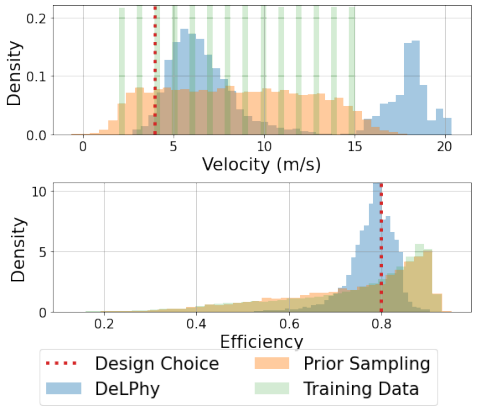
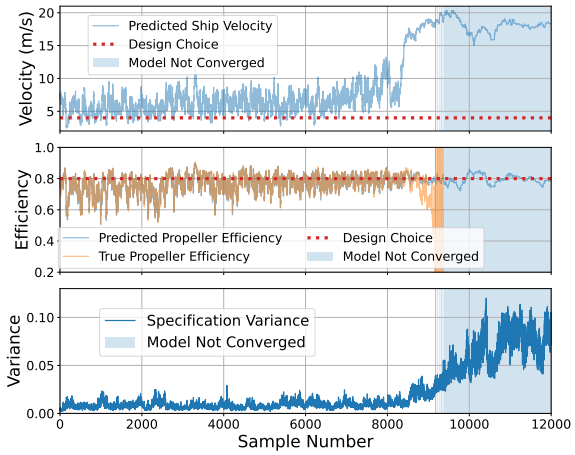
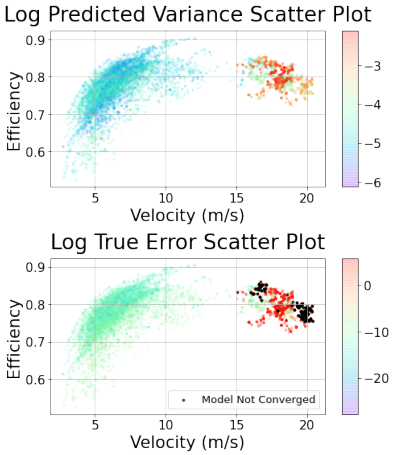


Figure 4: DeLPhy can find propellers with high efficiency at relatively low speed.



(a) Increased deviation of objective functions from OpenProp physics model is detected by DeLPhy.



(b) High true error or physics model failure corresponds to high uncertainty predicted by DeLPhy.

Figure 5: Uncertainty quantification detects unreliable designs during exploration.

Figure 5a, 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 5b 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. Figure 6 shows a few diverse propeller designs produced by DeLPhy 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.

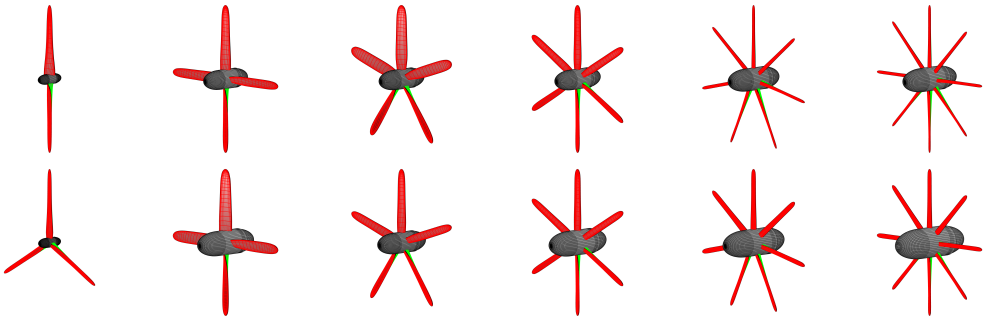
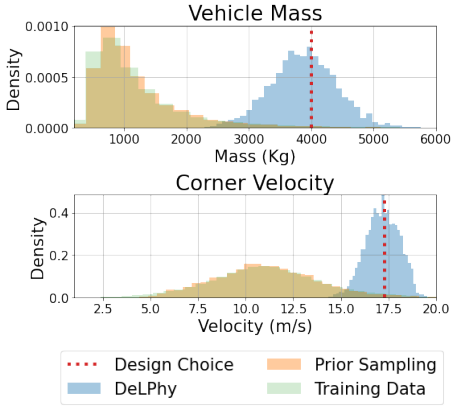
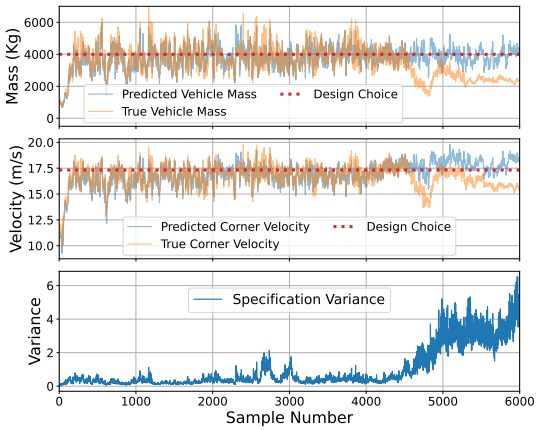


Figure 6: DeLPhy generates diverse designs with different numbers of blades, shape of blades, and hub diameter for similar velocity and efficiency objective. More details in Appendix C and E.

4.2 SAE Race Vehicle Design



(a) DeLPhy generates designs that meet the mass and corner velocity objectives.



(b) DeLPhy detects when the sampling is off-manifold and generates designs that are not reliable.

Figure 7: DeLPhy can reliably generate SAE race vehicle designs satisfying the design objective.

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 4,000 Kg, with a cornering velocity of 17.3 m/s which is challenging as maximum cornering velocity decreases with the mass of the vehicle. Further experimental and case study details are described in Appendix B. Figure 7a shows that DeLPhy samples designs that converge around the design objectives even though the training data and the prior distribution are far from it. Figure 7b shows that 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 sample $4,000$ in the illustrated sampling strategy but DeLPhy can detect this as the variance of the specification output also rapidly increases after $4,000$. Figure 8 shows that the race vehicle designs created by DeLPhy exhibits significant diversity in the choice of key components - engine, tires and brakes.

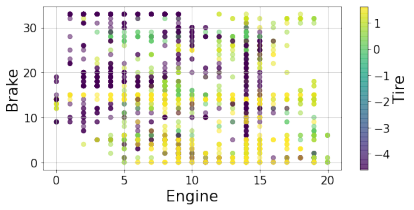
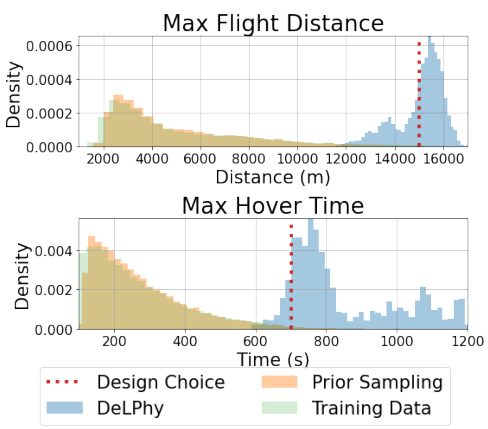


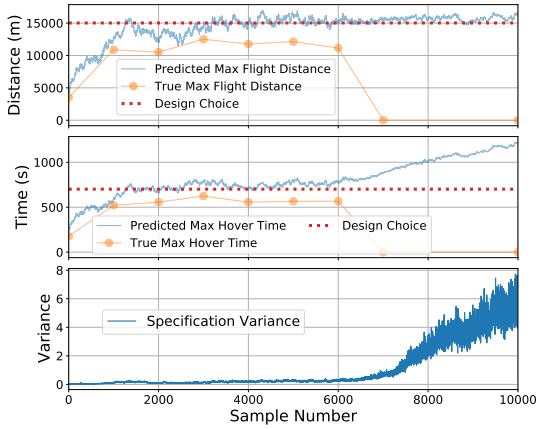
Figure 8: DeLPhy generates vehicles with different engines, tires and brakes (color denotes the third dimension).

4.3 AIR VEHICLE DESIGN

Air vehicles with rotors are capable of vertical takeoff and long-duration hovering, but they exhibit relatively shorter maximum distance of flight compared to fixed wing crafts. One important trade off in the design of such vehicles is to simultaneously meet the objectives of high hovering time and long distance of flight. We focus on battery-operated air vehicles with the main components being propellers, motor, electronic speed controllers and battery. We consider 414 possible APC propellers widely used in the rotorcraft community (APC, 2021). These propellers range across a large variety - thin electric, multi-rotor, slow flyer, reversible, wide chord, narrow chord, pusher and carbon propellers. We consider 83 possible motors and the corresponding speed controllers (TMotors, 2021), and 40 Nickel Cadmium (Ni-cad) batteries and Lithium-ion(Li-ion) batteries (Turnigy, 2021). There are several parameters of the design such as propeller diameter, pitch and mass, battery capacity, max-current and voltage, and motor voltage, max-current, Kt, Km and Kv. For a detailed physics model, we use a proprietary flight simulator model that provides flight diagnostics such as maximum flight distance, maximum hover time, and efficiency. Simpler but less accurate flight models are also available publicly for evaluation of individual designs (RCPlane, 2021; Cameron Dowd, 2021). We set the design targets for flight distance and hover time to be 15,000 m and 700 s respectively. More details on the case study is described in Appendix B and Appendix D.



(a) DeLPhy generates designs that meet the hover-time and flight distance objectives.



(b) DeLPhy detects when the generates designs cannot be trusted using the specification variance.

Figure 9: DeLPhy can reliably generate air vehicle designs satisfying the design objective.

Figure 9a shows that DeLPhy samples design that converge around the specified design objectives for the maximum fight distance and hover-time. Figure 9b shows a sample trajectory of designs generated by DeLPhy. As the maximum flight distance objective is annealed, we see that the proposed designs increasingly focus around a maximum flight distance of 15,000 m, as indicated by the dashed red line. As we keep the temperature of the predicted maximum hover time fixed at 1.0, its relative importance decreases and actually climbs well above the design choice, likely influenced by the focus on flight distance. The bottom plot in Figure 9b shows the variance across the dropout samples, when passed through the specification network. We can clearly see that at around sample 7,000, there is a jump in the variance of the proposed designs. In fact, when running every 1,000$^{\text{th}}$ sample through the high-fidelity physics model as the physics model is too slow to run many configurations, we actually see that the jump in variance has correctly indicated the true drop in actual performance, where the UAV is no

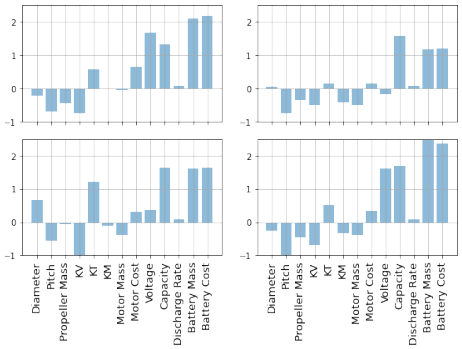


Figure 10: DeLPhy generates air vehicles with different key parameters.

longer able to take off. In fact at samples $8{,}000$ and $9{,}000$, the physics model fails to converge to any solution at all, signifying invalid designs. Figure 10 shows a few examples of the diverse air vehicle designs created by DeLPhy with significant diversity in the key design parameters.

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; 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 (Greenhill et al., 2020; 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 with HMC 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.

Ethics Statement: In this paper, we have developed a machine learning approach to aid the design of physical systems. Design automation can possibly reduce the need for human workforce skilled in solving this problem manually. We expect this concern to be offset by the need for a new workforce that is trained at using AI assistants for design and that can help build better AI co-designers. We expect the creative process of design to eventually be a joint endeavor that combines human ingenuity with learning-based extrapolation.

Reproducibility Statement: The implementation details are described in Section 4 with additional details provided in Appendix B, Appendix C, Appendix D and Appendix E. We have also provided Python notebooks in the supplementary material.

REFERENCES

APC. APC Propellers. `https://www.apcprop.com/technical-information/`, 2021.

Syrine Belakaria, Aryan Deshwal, Nitthilan Kannappan Jayakodi, and Janardhan Rao Doppa. Uncertainty-aware search framework for multi-objective Bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10044–10052, 2020.

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

David Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. In *International Conference on Machine Learning*, pp. 773–782. PMLR, 2019.

David H Brookes and Jennifer Listgarten. Design by adaptive sampling. *arXiv preprint arXiv:1810.03714*, 2018.

Cameron Dowd. PropPy propeller motor model, 2021. URL `https://github.com/cadowd/proppy`.

Adam D Cobb and Brian Jalaian. Scaling Hamiltonian Monte Carlo Inference for Bayesian Neural Networks with Symmetric Splitting. *arXiv preprint arXiv:2010.06772*, 2020.

Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Bayesian Optimization over Hybrid Spaces. *arXiv preprint arXiv:2106.04682*, 2021.

Ship Design Tools At DMS. OpenProp - integrated rotor design and analysis, 2021. URL `https://dmsonline.us/openprop/`.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

Brenden Epps, Julie Chalfant, Richard Kimball, Alexandra Techet, Kevin Flood, and Chrysssostomos Chryssostomidis. 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, 2009.

Masahiro Fujita. Basic and advanced researches in logic synthesis and their industrial contributions. In *Proceedings of the 2019 International Symposium on Physical Design*, pp. 109–116, 2019.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pp. 1050–1059. PMLR, 2016.

Josif Grabocka, Randolf Scholz, and Lars Schmidt-Thieme. Learning surrogate losses. *arXiv preprint arXiv:1905.10108*, 2019.

Stewart Greenhill, Santu Rana, Sunil Gupta, Pratibha Vellanki, and Svetha Venkatesh. Bayesian optimization for adaptive experimental design: A review. *IEEE access*, 8:13937–13948, 2020.

Zhong-Hua Han, Ke-Shi Zhang, et al. Surrogate-based optimization. *Real-world applications of genetic algorithms*, 343, 2012.

Nikolaus Hansen. The CMA evolution strategy: a comparing review. *Towards a new evolutionary computation*, pp. 75–102, 2006.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Slawomir Koziel, David Echeverría Ciaurri, and Leifur Leifsson. Surrogate-based methods. In *Computational optimization, methods and algorithms*, pp. 33–59. Springer, 2011.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Benedict Leimkuhler and Sebastian Reich. *Frontmatter*, pp. i–iv. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2005.

Lanlan Liu, Mingzhe Wang, and Jia Deng. 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. Springer, 2020.

Radford M Neal et al. MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 2(11):2, 2011.

Pascal Notin, José Miguel Hernández-Lobato, and Yarin Gal. Improving black-box optimization in VAE latent space using decoder uncertainty. *arXiv preprint arXiv:2107.00096*, 2021.

RCPlane. RCPlane online calculators, 2021. URL `https://rcplanes.online/`.

Danilo J Rezende and Fabio Viola. Generalized ELBO with constrained optimization, geco. In *Workshop on Bayesian Deep Learning, NeurIPS*, 2018.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286. PMLR, 2014.

Michael Rider. *Designing with Creo Parametric 2.0*. SDC Publications, 2013.

Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

Erik Schaltz and Seref Soylu. Electrical vehicle design and modeling. *Electric vehicles-modelling and simulations*, 1:1–24, 2011.

Ari Seff, Wenda Zhou, Nick Richardson, and Ryan P Adams. Vitruvion: A Generative Model of Parametric CAD Sketches. *arXiv preprint arXiv:2109.14124*, 2021.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

Armando Solar-Lezama. *Program synthesis by sketching*. University of California, Berkeley, 2008.

Nicolás F Soria Zurita, Mitchell K Colby, Irem Y Tumer, Christopher Hoyle, and Kagan Tumer. Design of complex engineered systems using multi-agent coordination. *Journal of Computing and Information Science in Engineering*, 18(1):011003, 2018.

Tadeusz Stolarski, Yuji Nakasone, and Shigeka Yoshimoto. *Engineering analysis with ANSYS software*. Butterworth-Heinemann, 2018.

Xingyuan Sun, Tianju Xue, Szymon M Rusinkiewicz, and Ryan P Adams. Amortized Synthesis of Constrained Configurations Using a Differentiable Surrogate. *arXiv preprint arXiv:2106.09019*, 2021.

TMotors. TMotors multiroter models, 2021. URL https://uav-en.tmotor.com/html/UAV/Multirotor/Motors/.

Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 33, 2020.

Turnigy. Turnigy power systems and batteries, 2021. URL https://turnigy.com/.

Jonathan Viquerat, Jean Rabault, Alexander Kuhnle, Hassan Ghraieb, Aurélien Larcher, and Elie Hachem. Direct shape optimization through deep reinforcement learning. *Journal of Computational Physics*, 428:110080, 2021.

Jie Xu, Andrew Spielberg, Allan Zhao, Daniela Rus, and Wojciech Matusik. Multi-Objective Graph Heuristic Search for Terrestrial Robot Design. *arXiv preprint arXiv:2107.05858*, 2021.

Allan Zhao, Jie Xu, Mina Konaković-Luković, Josephine Hughes, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. Robogrammar: graph grammar for terrain-optimized robot design. *ACM Transactions on Graphics (TOG)*, 39(6):1–16, 2020.

A DeLPhy Partial Simulated Annealing Algorithm

In this section we include the partial simulated annealing algorithmic component of DeLPhy. Part of the input to this stage are the three pre-trained networks, $S_{\boldsymbol{\mu}}(\mathbf{z}), D_{\boldsymbol{\phi}}(\mathbf{z}), E_{\boldsymbol{\theta}}(\mathbf{x})$, and the HMC parameters of step size, ϵ, and trajectory length, L. One must also set an initial $\mathbf{z}$, design targets $\mathbf{s}$, and number of samples to collect, N. The temperature annealing schedule for the individual objectives, $T_i(n)$, and prior $T(n)$ must also be passed as inputs. In Algorithm 1, the Leapfrog function on line 8 runs the leapfrog integration scheme (e.g. see Neal et al. (2011)) from initial momentum and latent space parameters to proposed momentum and latent space parameters. Note that in our work, we have not annealed the prior temperature, where we set $T(n) = 1$.

Algorithm 1 DeLPhy: partial simulated annealing

Inputs: $\mathbf{z}, \epsilon, L, S_{\boldsymbol{\mu}}(\mathbf{z}), D_{\boldsymbol{\phi}}(\mathbf{z}), E_{\boldsymbol{\theta}}(\mathbf{x}), N, \mathbf{s}, T_i(n), T_0(n)$

1: **for** n in $1, \ldots, N$ **do**
2: # *Sample Momentum*
3: $\mathbf{p} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
4: # *Calculate the individual components of the Hamiltonian* ***before*** *the leapfrog step.*
5: $u_i = -\log p(s_i|\mathbf{z})$
6: $u_{\text{prior}} = -\log p(\mathbf{z})$
7: $k = K(\mathbf{p})/(|\mathbf{s}| + 1)$
8: $\mathbf{z}', \mathbf{p}' = \text{Leapfrog}(\mathbf{z}, \mathbf{p}, \epsilon, L)$
9: # *Calculate the individual components of the Hamiltonian* ***after*** *the leapfrog step.*
10: $u_i' = -\log p(s_i|\mathbf{z}')$
11: $u_{\text{prior}}' = -\log p(\mathbf{z}')$
12: $k' = K(\mathbf{p}')/(|\mathbf{s}| + 1)$
13: # *Metropolis–Hastings correction*
14: $u \sim \mathcal{U}(0, 1)$
15: $\alpha = \sum_i^{|\mathbf{s}|} \frac{1}{T_i(n)} \left[u_i - u_i' + \frac{k-k'}{|\mathbf{s}|+1}\right] + \frac{1}{T(n)} \left[u_{\text{prior}} - u_{\text{prior}}' + \frac{k-k'}{|\mathbf{s}|+1}\right]$
16: **if** $\log u > \min(0, \alpha)$ **then**
17: $\mathbf{z}, \mathbf{p} \leftarrow \mathbf{z}', \mathbf{p}'$
18: **end if**
19: **end for**

B Experimental Details

Computational Infrastructure. For the partial simulated annealing component of DeLPhy, we adapt the python package *hamiltorch* (Cobb & Jalaian, 2020) and run with PyTorch version 1.9.1+cu111. We use Python 3.7.10. All experiments are run using NVIDIA's GeForce RTX 2080 Ti Graphics Card.

Propeller. We train a VAE with five layers of $[600, 500, 400, 300, 200]$ for the encoder, with the reverse ordering for the decoder. For the specification network, we us the layer structure of $[128, 64, 32]$. We use a latent space of size 100. For the specification network, we learn to regress the two objectives of **ship velocity** and **efficiency**. We use a data set of size 226,610 and train on $85\ \%$ and validate on the final $15\ \%$ to ensure the model is adequately trained. All input features are normalised using the training data means and standard deviations. We use PyTorch's Adam Kingma & Ba (2014) optimizer with a step size of 10^{-4} and momentum 0.9. We train for 200 epochs with a batch size of 1024. We anneal the temperature of the efficiency component following $10\text{e}^{-6\times10^{-4}k}$, and collect 12,000 samples with a step size of 0.1 and trajectory length of 1. In the decoder we set dropout to 0.05 at training time and collect 100 Monte Carlo dropout samples at test time. The acceptance rate for the partial simulated annealing was 0.6.

SAE Race Vehicle. To learn the latent space vehicle design we train a VAE with two layers of $[512, 256]$ for all of the components including the specification network. We use a latent space of size 32. The design input space is expanded to be 59 dimensional (from 39) to include the properties of each component (e.g. rather than selecting from a discrete choice of motors, we include power,

torque, etc. in the input space). The full feature and objective space is given by either Figure 1 in Section 2 or by Table 1 in Soria Zurita et al. (2018). For training, we use the PyTorch's Adam Kingma & Ba (2014) optimizer with a step size of 10^{-3} and its default parameters. We train the VAE for 1000 epochs with a batch size of 1028, and a learning rate schedule that decreased the learning rate by a factor of 10 at epochs 400 and 800. For the specification network, we learn to regress all 11 objectives. We use a data set of size $300{,}000$ and train on 80 $\%$ and validate on the final 20 $\%$ to ensure the model is adequately trained. For the decoder we train with a dropout of 0.05 and collect 100 Monte Carlo dropout samples at test time. We anneal the temperature of the mass design objective following $10\mathrm{e}^{-6\times10^{-3}k}$, and collect $6{,}000$ samples with a step size of 0.1 and trajectory length of 1. The acceptance rate for the partial simulated annealing was 0.73.

Air Vehicle Design. To learn the latent space of design of vehicles we train a VAE with two layers of $[512, 256]$ for all the components including the specification network. Our VAE model is trained over $52{,}202$ designs and validated over $13{,}051$ with a latent space of 8 dimensions. The training data is collected on a lower fidelity simulator than the test simulator which leads to the expectation that results will be slightly worse on the higher fidelity simulator compared to the validation data performance. For training, we use the PyTorch's Adam Kingma & Ba (2014) optimizer with a step size of 10^{-3} and its default parameters. We train the VAE for 1000 epochs with a batch size of 1028, and a learning rate schedule that decreased the learning rate by a factor of 10 at epochs 300 and 400. For the decoder we train with a dropout of 0.05 and collect 100 Monte Carlo dropout samples at test time. We anneal the temperature of the flight distance objective following $10\mathrm{e}^{\times10^{-3}k}$, and collect $10{,}000$ samples with a step size of 0.002 and an HMC trajectory length of 10. The acceptance rate for the partial simulated annealing was 0.79.

C PROPELLER DESIGN

In this section, we illustrate key geometric properties of a propeller, 3D geometry of propellers, and diversity of propeller designs generated by `DeLPhy`. Figure 11 shows the key geometric properties of a propeller such as number of blades, radius of the propeller, hub of the propeller, chord length and pitch angle of the blade along the length of the blade. Figure 12 shows the 3D shape of the propeller from views. Finally, Figure 13 and 14 show the diversity of propeller designs in terms of number of blades, shape of blades, pitch angles, and hub diameter for the same objective of efficiency higher than 75% and velocity lower than 4.5 m/s.

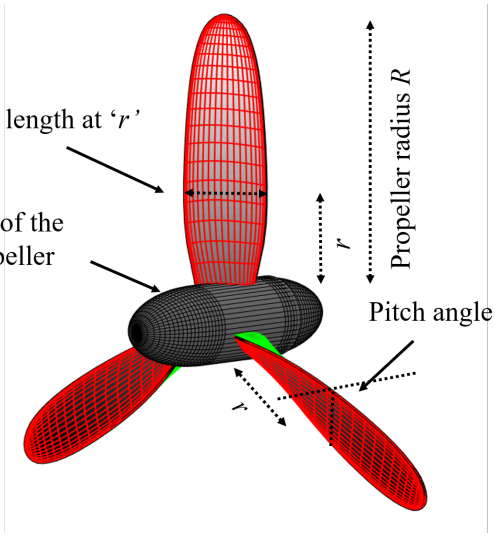


Figure 11: Key geometric properties of a propeller. The front surface is colored in red and the back surface is colored in green. The grid denotes lengthwise and chordwise discretization for the numerical computation.

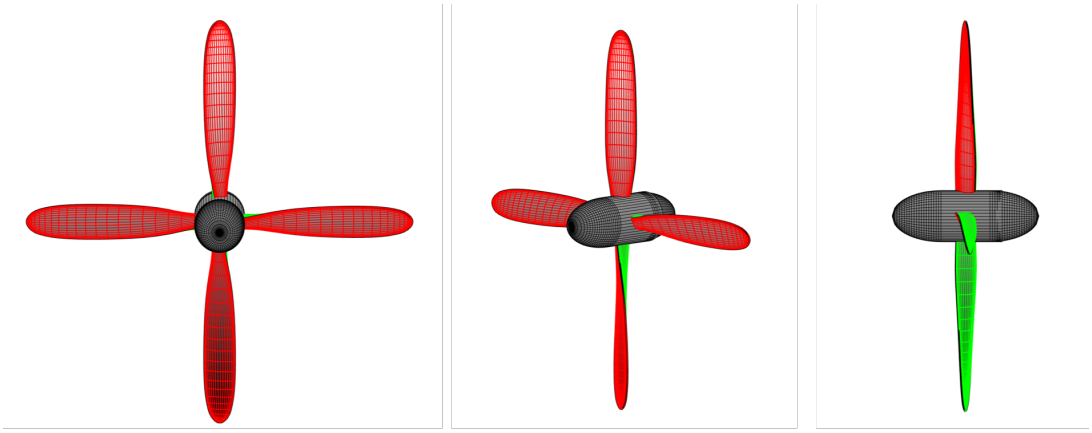


Figure 12: We show three views to illustrate the 3D shape of the propeller - 1) frontal view, 2) view at a 45 degree angle, and 3) side view. The front surface is colored in red and the back surface is colored in green. The grid denotes lengthwise and chordwise discretization for the numerical computation.

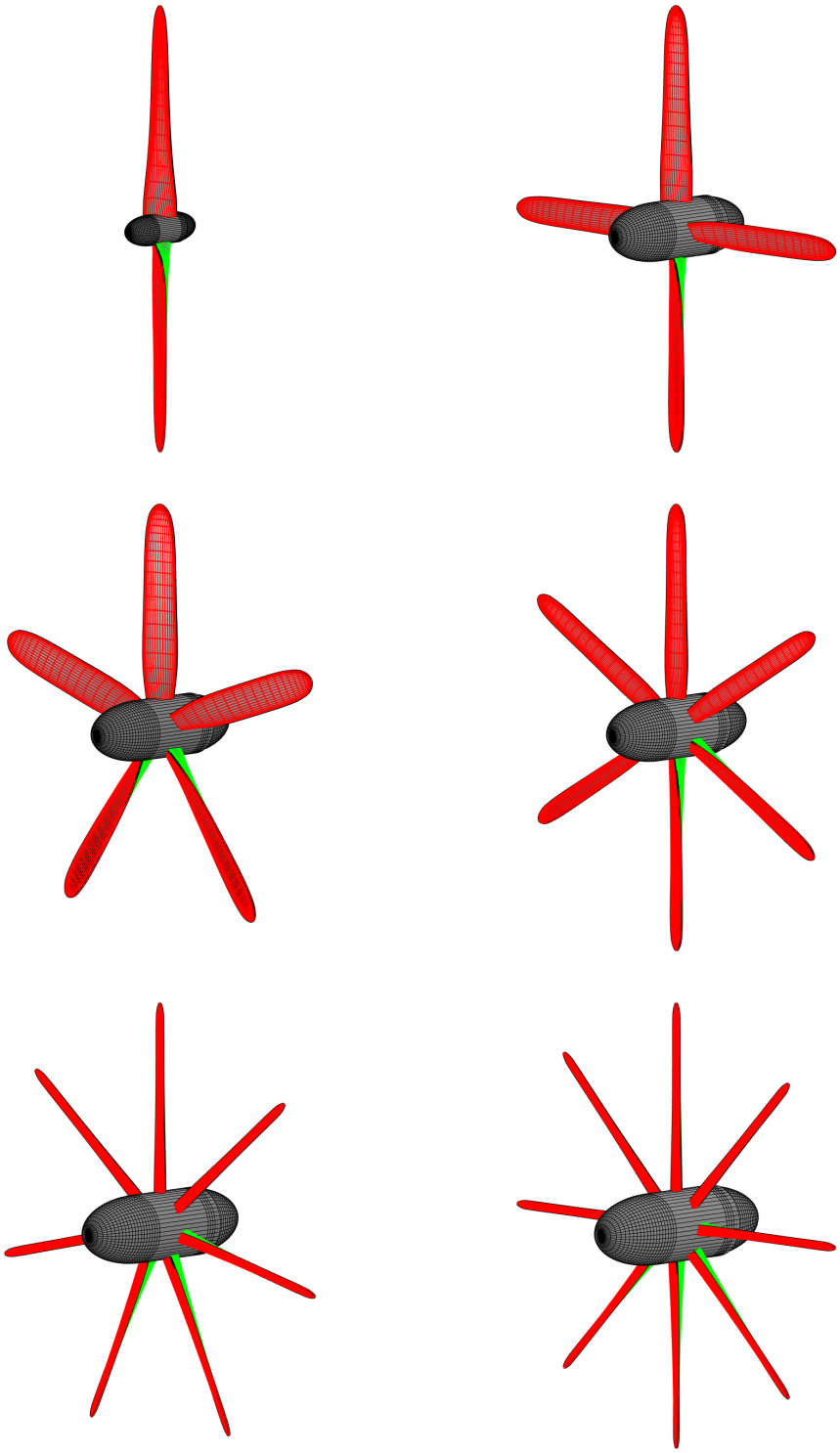


Figure 13: DeLPhy generates diverse designs (with different numbers of blades, shape of blades, pitch angles, and hub diameter) for similar velocity and efficiency objective.

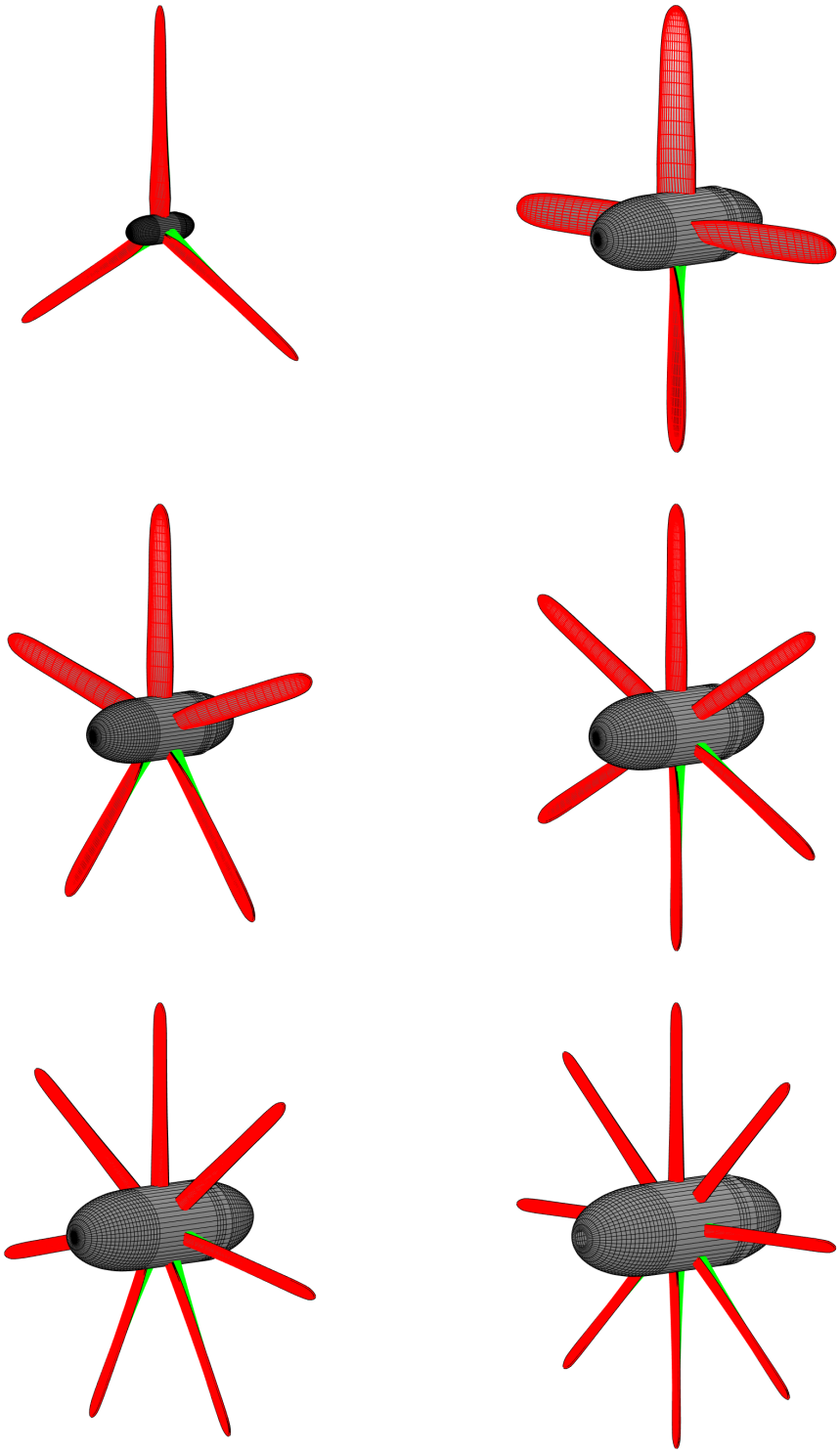


Figure 14: DeLPhy generates diverse designs (with different numbers of blades, shape of blades, pitch angles, and hub diameter) for similar velocity and efficiency objective.

D AIR VEHICLE DESIGN

In this section we include the list of the batteries, propellers, and motors in Tables 1, 2, 3 respectively.

Table 1: Battery properties for Air Vehicle Design

Name	Cost [$]	Mass [g]	Voltage [V]	Capacity [mAh]	Cont. Discharge Rate [C]
Turnigy Graphene 800mAh 2S 20C	8.36	58	7.4	800	20
Turnigy Graphene 1000mAh 2S 75C	9.54	84	7.4	1000	75
Turnigy Graphene 1000mAh 3S 75C	16.88	116	11.1	1000	75
Turnigy Graphene 1300mAh 3S 75C	16.96	137	11.1	1300	75
Turnigy Graphene 1500mAh 3S 75C	22.62	162	11.1	1500	75
Turnigy Graphene 1000mAh 4S 75C	23.65	148	14.8	1000	75
Turnigy Graphene 1300mAh 4S 75C	23.99	173	14.8	1300	75
Turnigy Graphene 1400mAh 3S 75C	24.00	156	11.1	1400	75
Turnigy Graphene 1600mAh 4S 75C Square	24.99	212	14.8	1600	75
Turnigy Graphene 1500mAh 4S 75C	26.49	202	14.8	1500	75
Turnigy Graphene 1000mAh 6S 75C	26.79	212	22.2	1000	75
Turnigy Graphene 1400mAh 4S 75C	27.87	196	14.8	1400	75
Turnigy Graphene 1200mAh 6S 75C	27.99	232	22.2	1200	75
Turnigy Graphene 1600mAh 4S 75C	30.00	212	14.8	1600	75
Turnigy Graphene 2200mAh 3S 75C	32.73	230	11.1	2200	75
Turnigy Graphene 3000mAh 3S 75C	42.54	320	11.1	3000	75
Turnigy Graphene 2200mAh 4S 75C	42.63	295	14.8	2200	75
Turnigy Graphene 4000mAh 3S 75C	55.05	412	11.1	4000	75
Turnigy Graphene 3000mAh 4S 75C	56.77	405	14.8	3000	75
Turnigy Graphene 5000mAh 3S 75C	65.48	490	11.1	5000	75
Turnigy Graphene 4000mAh 4S 75C	70.83	526	14.8	4000	75
Turnigy Graphene 6000mAh 3S 75C	75.16	630	11.1	6000	75
Turnigy Graphene 5000mAh 4S 75C	81.70	630	14.8	5000	75
Turnigy Graphene 3000mAh 6S 75C	82.57	598	22.2	3000	75
Turnigy Graphene 6000mAh 4S 75C	99.80	800	14.8	6000	75
Turnigy Graphene 4000mAh 6S 75C	104.34	760	22.2	4000	75
Turnigy Graphene 5000mAh 6S 75C	116.32	920	22.2	5000	75
Turnigy Graphene 6000mAh 6S 75C	129.99	1140	22.2	6000	75
Turnigy Receiver 1500mAh 4.8V	6.24	95	4.8	1500	10
Turnigy Receiver 1500mAh 6.0V	7.72	120	6.0	1500	10
Turnigy nano-tech 2000mAh 20~40C	14.73	109	6.6	2000	20
Turnigy nano-tech 1700mAh 20~40C	12.33	97	6.6	1700	20
Turnigy nano-tech 3000mAh 20~40C	21.13	167	6.6	3000	20
Turnigy nano-tech 1450mAh 20~40C	11.49	85	6.6	1450	20

Table 2: Propeller properties for Air Vehicle Design

Propeller Type	Diameter [mm]	Pitch [mm]	Mass [g]
10.5x6	266.700	152.40	34.881071
10x10	254.000	254.00	36.015046
10x10E	254.000	254.00	20.003319
10x3	254.000	76.20	26.081425
10x4	254.000	101.60	30.073017
10x4.5MR	254.000	114.30	15.013829
10x4.5MRP	254.000	114.30	15.013829
10x5	254.000	127.00	28.077221
10x5.5MR	254.000	139.70	15.013829
10x5.5MRP	254.000	139.70	15.013829
10x5E	254.000	127.00	20.003319
10x5EP	254.000	127.00	20.003319
10x6	254.000	152.40	28.939042
10x6E	254.000	152.40	20.003319
10x6EP	254.000	152.40	20.003319
10x7	254.000	177.80	30.889479
10x7E	254.000	177.80	20.003319
10x7EP	254.000	177.80	20.003319
10x8	254.000	203.20	28.939042
10x8E	254.000	203.20	20.003319
10x9	254.000	228.60	32.885275
11.5x4	292.100	101.60	41.095254
11.5x6	292.100	152.40	45.086846
11x10	279.400	254.00	41.957075
11x10E	279.400	254.00	22.997013
11x11	279.400	279.40	41.957075
11x12	279.400	304.80	41.957075
11x12E	279.400	304.80	25.990707
11x12W	279.400	304.80	43.091050
11x13	279.400	330.20	41.095254
11x14	279.400	355.60	39.144817
11x3	279.400	76.20	39.144817
11x4	279.400	101.60	39.144817
11x4.5EP	279.400	114.30	24.947450
11x4.5MR	279.400	114.30	17.009625
11x4.5MRP	279.400	114.30	17.009625
11x5	279.400	127.00	39.144817
11x5.5E	279.400	139.70	22.997013
11x5.5EP	279.400	139.70	22.997013
11x5.5MR	279.400	139.70	17.009625
11x5.5MRP	279.400	139.70	17.009625
11x6	279.400	152.40	39.961279
11x7	279.400	177.80	39.961279
11x7E	279.400	177.80	22.997013
11x8	279.400	203.20	41.095254
11x8.5E	279.400	215.90	23.994911
11x8E	279.400	203.20	22.997013
11x8EP	279.400	203.20	22.997013

11x9	279.400	228.60	41.957075
12.25x3.75	311.150	95.25	43.091050
12.5x10	317.500	254.00	51.028875
12.5x6	317.500	152.40	53.024671
12x10	304.800	254.00	45.948667
12x10E	304.800	254.00	25.990707
12x10W	304.800	254.00	51.890696
12x11	304.800	279.40	47.082642
12x12	304.800	304.80	51.028875
12x12E	304.800	304.80	25.990707
12x12EP	304.800	304.80	25.990707
12x12N	304.800	304.80	45.948667
12x13	304.800	330.20	45.086846
12x13N	304.800	330.20	47.082642
12x4	304.800	101.60	39.961279
12x4.5MR	304.800	114.30	22.135192
12x4.5MRP	304.800	114.30	22.135192
12x5	304.800	127.00	41.095254
12x5.5MR	304.800	139.70	22.135192
12x5.5MRP	304.800	139.70	22.135192
12x6	304.800	152.40	45.948667
12x6E	304.800	152.40	26.988605
12x6EP	304.800	152.40	26.988605
12x7	304.800	177.80	43.091050
12x8	304.800	203.20	47.899104
12x8E	304.800	203.20	25.990707
12x8EP	304.800	203.20	25.990707
12x9	304.800	228.60	43.952871
12x9W	304.800	228.60	53.024671
13.5x10	342.900	254.00	51.028875
13.5x13.5	342.900	342.90	66.904525
13.5x14	342.900	355.60	66.904525
13.5x9	342.900	228.60	49.033079
13x10	330.200	254.00	60.100675
13x10E	330.200	254.00	29.982299
13x10EP	330.200	254.00	29.982299
13x11	330.200	279.40	56.970904
13x13.5N	330.200	342.90	51.890696
13x13N	330.200	330.20	53.024671
13x14	330.200	355.60	65.997345
13x4	330.200	101.60	49.894900
13x4.5EP	330.200	114.30	30.889479
13x4.5MR	330.200	114.30	24.085629
13x4.5MRP	330.200	114.30	24.085629
13x4E	330.200	101.60	29.982299
13x4EP	330.200	101.60	29.982299
13x4W	330.200	101.60	49.894900
13x5.5E	330.200	139.70	31.978095
13x5.5EP	330.200	139.70	34.019250
13x5.5MR	330.200	139.70	23.994911
13x5.5MRP	330.200	139.70	23.994911
13x6	330.200	152.40	47.899104
13x6.5E	330.200	165.10	29.982299
13x6.5EP	330.200	165.10	29.982299
13x7	330.200	177.80	47.899104
13x8	330.200	203.20	49.033079
13x8E	330.200	203.20	30.980197
13x8EP	330.200	203.20	30.980197
13x9	330.200	228.60	60.962496
14x10	355.600	254.00	73.980529
14x10E	355.600	254.00	34.019250
14x11	355.600	279.40	72.846554
14x12	355.600	304.80	77.110300
14x12E	355.600	304.80	35.017148
14x12N	355.600	304.80	66.995243
14x13	355.600	330.20	68.900321
14x13.5	355.600	342.90	70.034296
14x13N	355.600	330.20	68.900321
14x14	355.600	355.60	68.038500
14x14E	355.600	355.60	36.015046
14x14N	355.600	355.60	68.900321
14x4W	355.600	101.60	56.154442
14x5.5MR	355.600	139.70	30.073017
14x5.5MRP	355.600	139.70	30.073017
14x5N	355.600	127.00	49.894900
14x6	355.600	152.40	71.213630
14x6E	355.600	152.40	37.012944
14x7	355.600	177.80	77.110300
14x7E	355.600	177.80	34.019250
14x7EP	355.600	177.80	34.019250
14x8	355.600	203.20	70.034296
14x8.5E	355.600	215.90	37.012944
14x8.5EP	355.600	215.90	37.012944
15.5x12W	393.700	304.80	104.915367
15x10	381.000	254.00	89.856179
15x10E	381.000	254.00	44.996128
15x10EP	381.000	254.00	44.996128
15x11	381.000	279.40	90.990154
15x12	381.000	304.80	90.990154
15x4E	381.000	101.60	44.996128
15x4EP	381.000	101.60	44.996128
15x4W	381.000	101.60	70.034296

15x5.5MR	381.000	139.70	36.015046
15x5.5MRP	381.000	139.70	36.015046
15x6	381.000	152.40	75.976325
15x6E	381.000	152.40	44.996128
15x7	381.000	177.80	77.110300
15x7E	381.000	177.80	44.996128
15x8	381.000	203.20	86.000664
15x8E	381.000	203.20	43.998230
16x10	406.400	254.00	104.915367
16x10E	406.400	254.00	51.981414
16x10EP	406.400	254.00	51.981414
16x12	406.400	304.80	102.919571
16x12E	406.400	304.80	51.981414
16x13	406.400	330.20	106.049342
16x14	406.400	355.60	104.915367
16x16	406.400	406.40	104.053546
16x4E	406.400	101.60	55.020467
16x4EP	406.400	101.60	55.020467
16x4W	406.400	101.60	85.048125
16x5.5MR	406.400	139.70	43.952871
16x5.5MRP	406.400	139.70	43.952871
16x6	406.400	152.40	89.856179
16x6E	406.400	152.40	56.018365
16x7	406.400	177.80	94.981746
16x8	406.400	203.20	100.923775
16x8E	406.400	203.20	51.981414
17x10	431.800	254.00	115.121142
17x10E	431.800	254.00	64.001549
17x10N	431.800	254.00	98.111517
17x10WE	431.800	254.00	84.004868
17x12	431.800	304.80	119.929196
17x12E	431.800	304.80	68.038500
17x12W	431.800	304.80	132.947229
17x13	431.800	330.20	117.933400
17x4W	431.800	101.60	94.981746
17x6	431.800	152.40	111.991371
17x6E	431.800	152.40	64.001549
17x7E	431.800	177.80	64.001549
17x8	431.800	203.20	115.937604
17x8E	431.800	203.20	64.001549
17x8N	431.800	203.20	94.119925
18.1x10	459.740	254.00	132.947229
18.1x12	459.740	304.80	125.009404
18x10	457.200	254.00	132.947229
18x10E	457.200	254.00	72.982631
18x10EP	457.200	254.00	72.982631
18x12	457.200	304.80	144.876646
18x12E	457.200	304.80	73.980529
18x12WE	457.200	304.80	79.015378
18x14	457.200	355.60	151.090829
18x5.5MR	457.200	139.70	61.007855
18x5.5MRP	457.200	139.70	61.007855
18x6W	457.200	152.40	134.943025
18x8	457.200	203.20	132.130767
18x8E	457.200	203.20	72.982631
18x8EP	457.200	203.20	72.982631
18x8W	457.200	203.20	142.880850
19x10E	482.600	254.00	83.006970
19x10EP	482.600	254.00	83.006970
19x11	482.600	279.40	132.130767
19x12E	482.600	304.80	83.006970
19x12WE	482.600	304.80	111.991371
19x8E	482.600	203.20	83.006970
19x8W	482.600	203.20	163.020246
20.5x12WE	520.700	304.80	130.996792
20.5x14E	520.700	355.60	123.058967
20x10	508.000	254.00	185.971900
20x10E	508.000	254.00	95.979644
20x10EP	508.000	254.00	95.979644
20x10W	508.000	254.00	202.119704
20x11E	508.000	279.40	99.018697
20x12	508.000	304.80	191.097467
20x12WE	508.000	304.80	130.996792
20x13E	508.000	330.20	98.020799
20x13EP	508.000	330.20	98.020799
20x14	508.000	355.60	199.987831
20x15C	508.000	381.00	151.090829
20x15E	508.000	381.00	117.933400
20x8	508.000	203.20	182.025667
20x8E	508.000	203.20	95.979644
20x8W	508.000	203.20	193.047904
21x10W	533.400	254.00	225.116717
21x12WE	533.400	304.80	147.144596
21x13E	533.400	330.20	130.134971
21x13WE	533.400	330.20	128.003098
22x10	558.800	254.00	233.871004
22x10E	558.800	254.00	133.990486
22x12E	558.800	304.80	135.986282
22x12WE	558.800	304.80	157.985397
22x8	558.800	203.20	240.130546
24x12E	609.600	304.80	164.018144
25x12.5E	635.000	317.50	223.120921

26x13E	660.400	330.20	209.014272
26x15E	660.400	381.00	214.003762
27x13E	685.800	330.20	232.011285
27x13EP	685.800	330.20	232.011285
4.1x4.1E	104.140	104.14	3.129771
4.1x4.1EP	104.140	104.14	3.129771
4.2x2	106.680	50.80	3.129771
4.2x4	106.680	101.60	3.129771
4.5x3.5	114.300	88.90	3.991592
4.5x4	114.300	101.60	3.991592
4.5x4.1E	114.300	104.14	3.991592
4.75x4	120.650	101.60	5.942029
4.75x4.5E	120.650	114.30	3.991592
4.75x4.5EC	120.650	114.30	3.129771
4.75x4.75E	120.650	120.65	3.991592
4.75x4.75EC	120.650	120.65	3.129771
4.75x4.75EP	120.650	120.65	3.991592
4.75x5.5E	120.650	139.70	3.991592
4.7x4.2E	119.380	106.68	3.129771
5.1x4.5E	129.540	114.30	5.125567
5.25x5.5E	133.350	139.70	3.991592
5.25x6.25E	133.350	158.75	3.991592
5.5x2	139.700	50.80	7.076004
5.5x2.5	139.700	63.50	7.076004
5.5x4.5E	139.700	114.30	3.991592
5.5x4.5EP	139.700	114.30	3.991592
5.5x4.7E	139.700	119.38	3.991592
5.5x6.5E	139.700	165.10	3.991592
5.7x3	144.780	76.20	5.125567
5x3	127.000	76.20	5.942029
5x3E	127.000	76.20	3.129771
5x3EP	127.000	76.20	3.129771
5x4R-RH	127.000	101.60	7.983184
5x5E	127.000	127.00	3.991592
5x5EP	127.000	127.00	3.991592
5x7.5E	127.000	190.50	4.989490
5x7.5EP	127.000	190.50	4.989490
6.3x4	160.020	101.60	11.067596
6.5x2.9	165.100	73.66	11.067596
6.5x3.7	165.100	93.98	11.067596
6.5x5.0	165.100	127.00	11.067596
6.5x5.5	165.100	139.70	11.067596
6.5x6.0	165.100	152.40	11.067596
6.5x6.5	165.100	165.10	11.067596
6x2	152.400	50.80	7.937825
6x3	152.400	76.20	7.937825
6x3R-RH	152.400	76.20	7.983184
6x4	152.400	101.60	5.125567
6x4E	152.400	101.60	5.125567
6x4EP	152.400	101.60	5.125567
6x5	152.400	127.00	6.985286
6x5.5E	152.400	139.70	5.125567
6x6E	152.400	152.40	5.125567
6x6EP	152.400	152.40	5.125567
7.4x7.5C	187.960	190.50	15.875650
7.4x7.6C	187.960	193.04	15.875650
7.4x7.7C	187.960	195.58	15.875650
7.8x4	198.120	101.60	17.871446
7.8x6	198.120	152.40	17.871446
7.8x7	198.120	177.80	17.871446
7x10	177.800	254.00	13.063392
7x3	177.800	76.20	11.929417
7x4	177.800	101.60	11.929417
7x4E	177.800	101.60	7.983184
7x4EP	177.800	101.60	7.983184
7x5	177.800	127.00	11.067596
7x5E	177.800	127.00	7.983184
7x5EP	177.800	127.00	7.983184
7x6	177.800	152.40	13.063392
7x6E	177.800	152.40	7.983184
7x6EP	177.800	152.40	7.983184
7x6W	177.800	152.40	13.063392
7x7	177.800	177.80	13.063392
7x7E	177.800	177.80	7.983184
7x8	177.800	203.20	13.063392
7x9	177.800	228.60	13.063392
8.75x5.0	222.250	127.00	28.077221
8.75x7.0N	222.250	177.80	28.077221
8.75x7.5N	222.250	190.50	28.077221
8.75x7.5NN	222.250	190.50	28.939042
8.75x7.5W	222.250	190.50	24.947450
8.75x7.75NN	222.250	196.85	24.947450
8.75x8.0NN	222.250	203.20	28.939042
8.75x8.5N	222.250	215.90	28.077221
8.75x8.75NN	222.250	222.25	24.947450
8.75x8.75W	222.250	222.25	24.947450
8.75x9.0NN	222.250	228.60	26.081425
8.75x9.0W	222.250	228.60	26.081425
8.75x9.25NN	222.250	234.95	24.947450
8.75x9.25W	222.250	234.95	24.947450
8.8x8.5	223.520	215.90	24.947450
8.8x8.75	223.520	222.25	26.943246

8.8x8.9	223.520	226.06	26.081425
8.8x9.0	223.520	228.60	26.081425
8.8x9.25	223.520	234.95	26.943246
8.8x9.5	223.520	241.30	26.081425
8.8x9.75	223.520	247.65	26.081425
8x10	203.200	254.00	19.005421
8x3.75	203.200	95.25	17.009625
8x4	203.200	101.60	15.875650
8x4.5MR	203.200	114.30	9.071800
8x4.5MRP	203.200	114.30	9.071800
8x4E	203.200	101.60	13.018033
8x5	203.200	127.00	19.005421
8x6	203.200	152.40	19.005421
8x6E	203.200	152.40	14.015931
8x6EP	203.200	152.40	14.015931
8x7	203.200	177.80	19.005421
8x8	203.200	203.20	19.005421
8x8E	203.200	203.20	15.013829
8x8EP	203.200	203.20	15.013829
8x9	203.200	228.60	19.005421
9.25x6.0	234.950	152.40	17.009625
9.3x3	236.220	76.20	24.947450
9.5x4.5	241.300	114.30	28.077221
9.5x6	241.300	152.40	26.943246
9.5x7.5N	241.300	190.50	28.939042
9.5x7N	241.300	177.80	28.939042
9.625x3.75N	244.475	95.25	30.889479
9x10	228.600	254.00	22.135192
9x3	228.600	76.20	24.947450
9x3N	228.600	76.20	15.013829
9x4	228.600	101.60	22.951654
9x4.5E	228.600	114.30	17.962164
9x4.5EP	228.600	114.30	17.962164
9x4.5MR	228.600	114.30	11.067596
9x4.5MRP	228.600	114.30	11.067596
9x4.5R-RH	228.600	114.30	26.036066
9x5	228.600	127.00	22.135192
9x6	228.600	152.40	22.135192
9x6.5	228.600	165.10	30.073017
9x6E	228.600	152.40	18.007523
9x6EP	228.600	152.40	18.007523
9x6N	228.600	152.40	17.871446
9x7	228.600	177.80	22.951654
9x7.5	228.600	190.50	30.073017
9x7.5C	228.600	190.50	20.139396
9x7.5E	228.600	190.50	18.007523
9x8	228.600	203.20	24.085629
9x8.5	228.600	215.90	30.073017
9x9	228.600	228.60	24.085629
9x9E	228.600	228.60	18.007523
7x4SF	177.800	101.60	5.125567
7x4SFP	177.800	101.60	5.125567
7x4.1SF	177.800	104.14	3.492643
7x5SF	177.800	127.00	5.125567
7x6SF	177.800	152.40	5.125567
8x3.8SF	203.200	96.52	7.076004
8x3.8SFP	203.200	96.52	7.076004
8x4.1SF	203.200	104.14	4.490541
8x4.2SFR	203.200	106.68	4.989490
8x4.7SF	203.200	119.38	7.076004
8x4.7SFP	203.200	119.38	7.076004
8x6SF	203.200	152.40	7.076004
9x3.7SF	228.600	93.98	4.989490
9x3.8SF	228.600	96.52	9.071800
9x3.8SFP	228.600	96.52	9.071800
9x4.1SF	228.600	104.14	4.989490
9x4.4SFR	228.600	111.76	6.531696
9x4.6SF	228.600	116.84	4.989490
9x4.7SF	228.600	119.38	9.071800
9x4.7SFP	228.600	119.38	9.071800
9x6SF	228.600	152.40	9.071800
9x7.5SF	228.600	190.50	9.933621
10x3.8SF	254.000	96.52	11.929417
10x3.8SFP	254.000	96.52	11.929417
10x4.6SF	254.000	116.84	7.983184
10x4.7SF	254.000	119.38	11.929417
10x4.7SFP	254.000	119.38	11.929417
10x7SF	254.000	177.80	11.929417
11x3.8SF	279.400	96.52	15.013829
11x3.8SFP	279.400	96.52	15.013829
11x4.6SF	279.400	116.84	9.978980
11x4.7SF	279.400	119.38	15.013829
11x4.7SFP	279.400	119.38	15.013829
12x3.8SF	304.800	96.52	17.871446
12x3.8SFP	304.800	96.52	17.871446
12x4.7SF	304.800	119.38	17.871446
12x4.7SFP	304.800	119.38	17.871446
12x6SF	304.800	152.40	19.005421
12x8SF	304.800	203.20	19.005421
13x4.7SF	330.200	119.38	22.135192
13x4.7SFP	330.200	119.38	22.135192
14x4.7SF	355.600	119.38	24.947450

14x4.7SFP	355.600	119.38	24.947450

Table 3: Motor properties for Air Vehicle Design

Motor Type	Cost [$]	KV [RPM/V]	KT [Nm/A]	Mass [g]
MT1306 3100KV	41.90	3100	0.003080	11.2
MN2204 1400KV	25.90	1400	0.006821	23.0
MT2208 1100 KV	43.90	1100	0.008681	45.0
MT2216 V2 800KV	54.90	800	0.011937	75.0
MN2212 KV780	46.90	780	0.012243	65.0
MN2212 KV920	46.90	920	0.010380	65.0
MN3110 KV470	61.90	470	0.020318	98.0
MN3110 KV700	61.90	700	0.013642	99.0
MN3110 KV780	61.90	780	0.012243	100.0
MN3508 KV380	69.90	380	0.025130	103.0
MN3508 KV580	69.90	580	0.016464	103.0
MN3508 KV700	69.90	700	0.013642	104.0
MN3510 KV360	79.90	360	0.026526	117.0
MN3510 KV630	79.90	630	0.015158	119.0
MN3510 KV700	79.90	700	0.013642	118.0
MN3515 KV400	96.90	400	0.023873	183.0
MN3520 KV400	109.90	400	0.023873	222.0
MN4010 KV370	86.90	370	0.025809	137.0
MN4010 KV475	86.90	475	0.020104	137.0
MN4010 KV580	86.90	580	0.016464	137.0
MN4012 KV340	92.90	340	0.028086	155.0
MN4012 KV400	92.90	400	0.023873	155.0
MN4012 KV480	92.90	480	0.019894	155.0
MN4014 KV330	96.90	330	0.028937	171.0
MN4014 KV400	96.90	400	0.023873	171.0
MN5208 KV340	99.90	340	0.028086	196.0
MN5212 KV340	109.90	340	0.028086	249.0
MN5212 KV420	109.90	420	0.022736	249.0
AT2308 KV1450	29.99	1450	0.006586	47.0
AT2308 KV2600	29.99	2600	0.003673	48.0
AT2310 KV2200	27.99	2200	0.004341	52.0
AT2312 KV1150	34.99	1150	0.008304	60.0
AT2312 KV1400	34.99	1400	0.006821	60.0
AT2317 KV880	39.99	880	0.010851	79.0
AT2317 KV1250	39.99	1250	0.007639	79.0
AT2317 KV1400	39.99	1400	0.006821	80.0
AT2321 KV950	44.99	950	0.010052	93.0
AT2321 KV1250	44.99	1250	0.007639	94.0
AT2814 KV900	49.99	900	0.010610	108.0
AT2814 KV1050	49.99	1050	0.009095	107.0
AT2814 KV1200	49.99	1200	0.007958	108.0
AT2820 KV880	59.99	880	0.010851	139.0
AT2820 KV1050	59.99	1050	0.009095	139.0
AT2820 KV1250	59.99	1250	0.007639	141.0
AT2826 KV900	69.99	900	0.010610	175.0
AT2826 KV1100	69.99	1100	0.008681	175.0
AT3520 KV550	79.99	550	0.017362	218.0
AT3520 KV720	79.99	720	0.013263	339.0
AT3520 KV850	79.99	850	0.011234	221.0
AT3530 KV580	99.99	580	0.016464	298.0
AT4120 KV250	109.99	250	0.038197	304.0
AT4120 KV500	109.99	500	0.019099	305.0
AT4120 KV560	109.99	560	0.017052	300.0
AT4125 KV250	115.99	250	0.038197	350.0
AT4125 KV540	115.99	540	0.017684	355.0
AT4130 KV230	119.99	230	0.041519	408.0
AT4130 KV300	119.99	300	0.031831	405.0
AT4130 KV450	119.99	450	0.021221	408.0
AS 2308 KV1450	18.99	1450	0.006586	49.0
AS 2308 KV2600	18.99	2600	0.003673	49.0
AS 2312 KV1150	19.99	1150	0.008304	63.0
AS 2312 KV1400	19.99	1400	0.006821	63.0
AS 2317 KV880	21.95	880	0.010851	81.0
AS 2317 KV1250	21.95	1250	0.007639	81.0
AS 2317 KV1400	21.95	1400	0.006821	81.0
AS 2814 KV900	33.99	900	0.010610	110.0
AS 2814 KV1050	33.99	1050	0.009095	112.0
AS 2814 KV1200	33.99	1200	0.007958	112.0
AS 2814 KV2000	33.99	2000	0.004775	116.0
AS 2820 KV880	34.00	880	0.010851	136.0
AS 2820 KV1050	34.00	1050	0.009095	144.0
AS 2820 KV1250	34.00	1250	0.007639	144.0
KDE2306XF-2550	28.95	2550	0.003700	42.0
KDE2315XF-965	60.95	965	0.009900	75.0
KDE2315XF-885	60.95	885	0.010800	75.0
KDE2814XF-775	71.95	775	0.012300	125.0
KDE2814XF-515	71.95	515	0.018500	125.0
KDE3510XF-715	92.95	715	0.013400	175.0
KDE3510XF-475	92.95	475	0.020100	175.0
KDE3520XF-400	112.95	400	0.023900	245.0
KDE4012XF-400	114.95	400	0.023900	200.0
KDE4014XF-380	118.95	380	0.025100	215.0
KDE4213XF-360	133.95	360	0.026500	230.0

E COMPARISON WITH DIRECT OPTIMIZATION

In this section, we compare DeLPhy with a commonly used direct optimization approach covariance matrix adaptation evolution strategy (CMA-ES) (Hansen, 2006) in terms of design diversity. As shown in Figure 15, our approach can generate a more diverse set of designs that match the design objective compared to the CMA-ES. Here, our design objective is to generate efficient propellers at a low velocity (efficiency higher than 75% and velocity lower than 4.5 m/s). Note that DeLPhy generates diverse designs in terms number of blades, the shape of blades, pitch angles, and hub diameter. CMA-ES generates designs with a slight variation of numbers of blades, whereas the other geometric properties are quite similar.

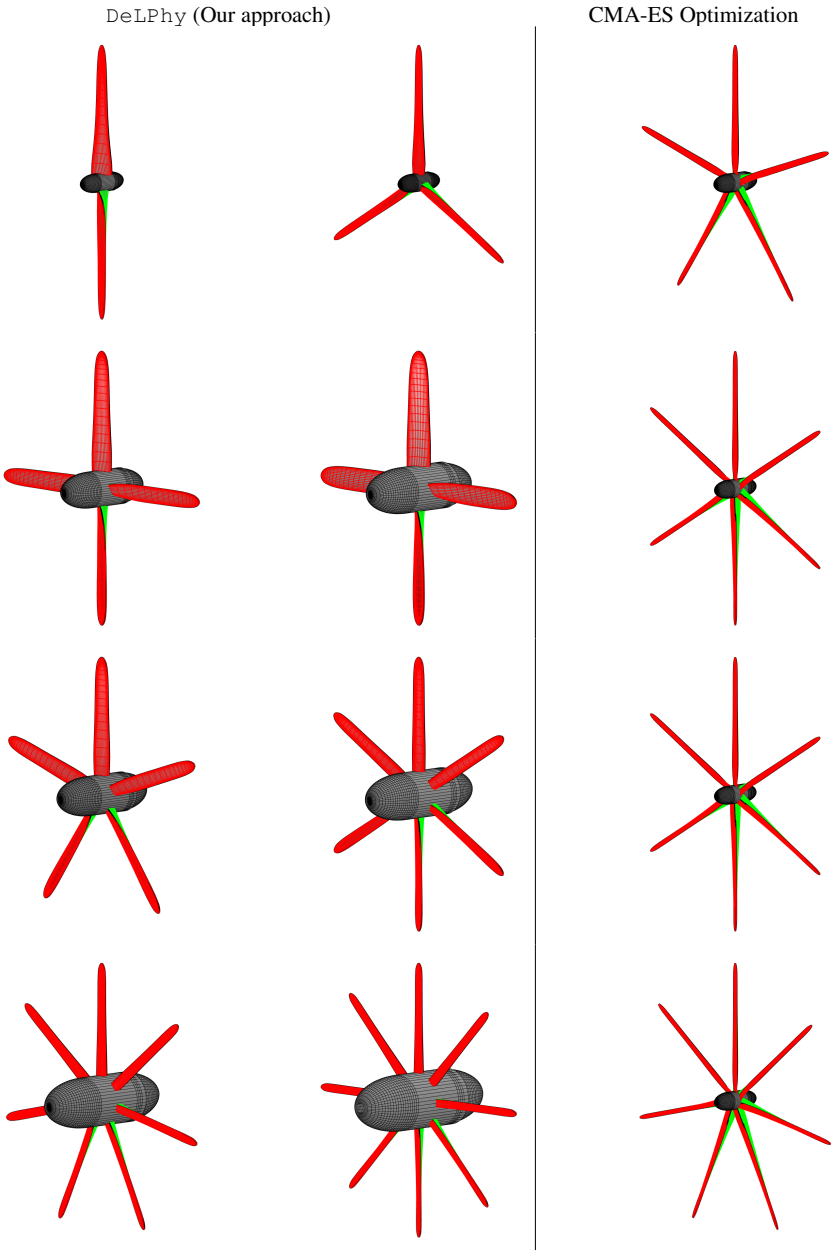


Figure 15: DeLPhy generates a more diverse set of designs, with a different numbers of blades, the shape of blades, pitch angles, and hub diameter, compared to CMA-ES (Hansen, 2006) which is a direct optimization approach.