ShipGen: A Diffusion Model for Parametric Ship Hull Generation with Multiple Objectives and Constraints

Preprint

Noah J. Bagazinski[*]
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
noahbagz@mit.edu

Faez Ahmed
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
faez@mit.edu

November 9, 2023

Abstract

Ship design is a years-long process that requires balancing complex design trade-offs to create a ship that is efficient and effective. Finding new ways to improve the ship design process can lead to significant cost savings in the time and effort required to design a ship and cost savings in the procurement and operation of a ship. One promising technology is generative artificial intelligence, which has been shown to reduce design cycle time and create novel, high-performing designs. In literature review, generative artificial intelligence has been shown to generate ship hulls; however, ship design is particularly difficult as the hull of a ship requires the consideration of many objectives. This paper presents a study on the generation of parametric ship hull designs using a parametric diffusion model that considers multiple objectives and constraints for the hulls. This denoising diffusion probabilistic model (DDPM) generates the tabular parametric design vectors of a ship hull, which is then constructed into a point cloud and mesh for performance evaluation. In addition to a tabular DDPM, this paper details adding guidance to improve the quality of generated parametric ship hull designs. By leveraging a classifier to guide sample generation, the DDPM produced feasible parametric ship hulls that maintain the coverage of the initial training dataset of ship hulls with a 99.5% rate, a 149x improvement over random sampling of the design vector parameters across the design space. Parametric ship hulls produced with performance guidance saw an average of 91.4% reduction in wave drag coefficients and an average of a 47.9x relative increase in the total displaced volume of the hulls compared to the mean performance of the hulls in the training dataset. The use of a DDPM to generate parametric ship hulls can reduce design time by generating high-performing hull designs for future analysis. These generated hulls have low drag and high volume, which can reduce the cost of operating a ship and increase its potential to generate revenue.

Keywords Naval Architecture · Generative Artificial Intelligence · Deep Generative Models · Denoising Diffusion Probabilistic Model · DDPM · Multi-objective Design · Design Constraint Satisfaction · Drag Reduction · Parametric Design · Ship Design

1 Introduction

Generative artificial intelligence (AI) models produce new instances of information that resemble the data used to train the model. While generative AI is famously known for generating text and image information, it can also be used to generate information to engineer products. Recent advances in generative AI provide promising new avenues to quickly generate designs. Including additional information in the training, such as a design's performance, can be leveraged to create designs with high performance. These advances are especially useful in the design of ships. Ship design currently requires a large team of naval architects to balance design trade-offs in a single ship's design. A generative AI model specifically trained to generate ship hulls can improve this workflow. Training such a model successfully is enabled by the availability of large datasets that include both design and performance information for ship hulls [1]. Hull design was chosen as a starting point for the generative model as the shape of the hull has a direct impact on over 70% of the cost of a ship [2]. It is also one of the first steps in the traditional workflow for ship design [3]. The

[*] Corresponding Author.

hull shape affects several key aspects of a ship's performance, including the buoyancy, upright stability, hydrodynamics, and general arrangements of the ship. With these considerations, the design of ship hulls provides an impactful avenue for the application of machine learning for engineering design.

A well-designed machine learning tool for ship design could learn design trade-offs for ships through the continual design and evaluation of many ship designs. This work demonstrates the use of a guided denoising diffusion probabilistic model (DDPM), a type of deep generative model, to rapidly generate high-performing and feasible parametric ship hull designs by generating parameters in a tabular format. This model, called ShipGen, generates early-stage hull designs considering seven performance metrics, creating shapes with low drag and high cargo-carrying capacity. Figure 1 shows an overview of the work presented, highlighting that the implementation of classifier and performance guidance during the sampling process generates hulls with high performance. This work features model training with a publicly available dataset of parametric ship hulls, called ShipD [1]. The following sections detail the literature review of previous work, the methodology for creating and evaluating a tabular DDPM, the evaluation of ship hulls generated by the DDPM, and a discussion on the impact of the work. The hulls generated with the use of the guided DDPM are intended to be candidate designs for future analysis. As such, these generated hulls may not necessarily look exactly like realistic hull forms, but instead have design features that, in combination, lead to high performance. Through the development of the performance-guided DDPM for ship hull generation, the novel contributions of this paper are:

1. The first known use of denoising diffusion probabilistic models for generating parametric tabular data for an engineering performance-focused design application.
2. Showcase that classifier guidance in the DDPM navigates complex design feasibility constraints to generate feasible samples with over 99% success while maintaining dataset design coverage.
3. Use of guidance to improve ship hull performance, with samples having an average 91.4% reduction in wave drag coefficient and 47.9x more displaced volume compared to the mean performance of the hulls in the dataset.

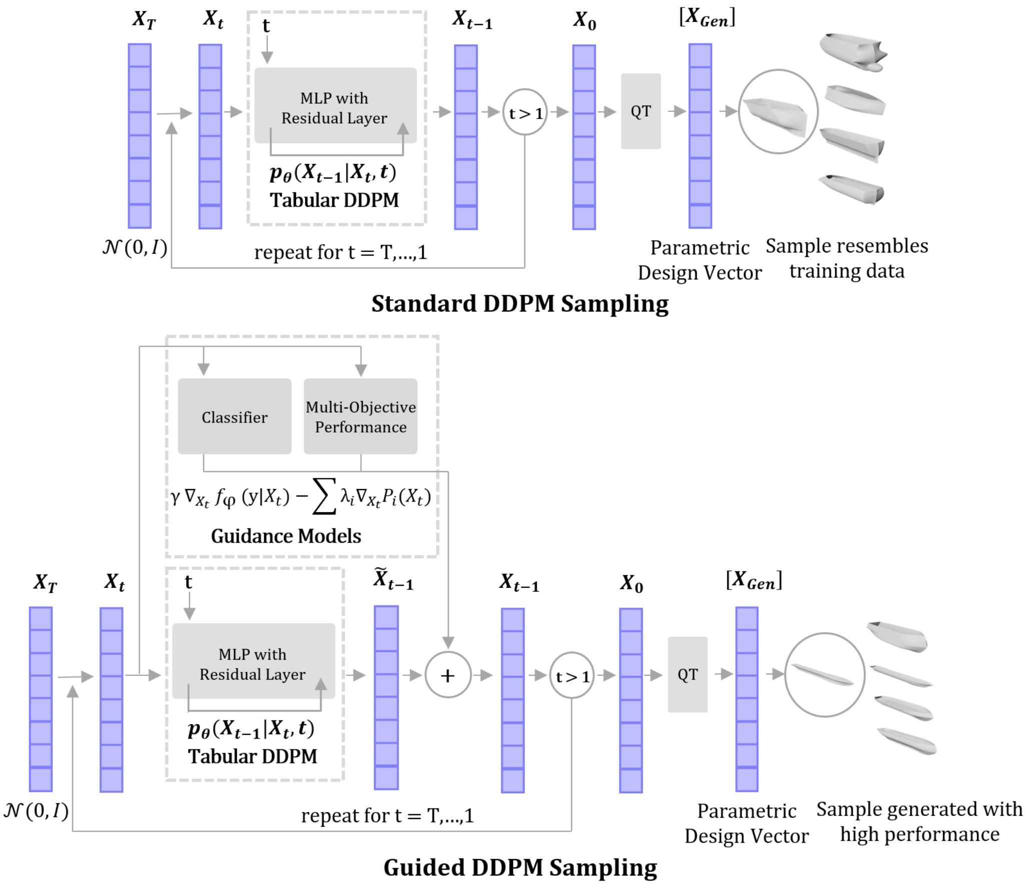


Figure 1: Overview of utilizing a DDPM to generate parametric ship hull designs. When leveraging classifier and performance guidance from pre-trained neural networks, the DDPM is able to generate ship hull designs with high performance.

2 Prior Work

Generative AI for ship hull design was influenced by research in computational ship design and machine learning literature. The first subsection details prior work in computational ship design, including ship hull design representation, hull form design optimization, and the use of machine learning in ship design. The second subsection details the development of diffusion models and their applications in engineering design.

2.1 Computational Ship Design

Computational Ship Design refers to the application of computer-based modeling, simulation, and optimization techniques in the design and analysis of marine vessels, facilitating more efficient, innovative, and integrated design solutions. Historically, computational ship design can be divided into three categories: design representation, forward modeling which includes surrogate models, and inverse design or synthesis, which includes optimization methods. Recently, generative AI methods have emerged as a powerful technique, which could be used for the representation and synthesis of ship hull designs.

In order to design a product with computational methods, the product needs to be represented in a way that a computer can understand. For ship design, the two most popular modes are parameterized vectors [1, 4–12], and free form deformation techniques [13–18]. The benefit of using parameterized design representations for a hull is that the design is defined by a set of tunable parameters that both human designers and computers can interpret. The ease of use of parametric design representations has often limited the diversity of possible hull shapes. Conversely, FFD techniques present a different landscape. They allow for the creation of a broad array of shapes. Yet, these representations can be challenging for humans to interpret without a visual representation of the hull form. The works of Khan et al. [19–21], Shaeffer et al. [22, 23], and Bagazinski et al. [1] have looked at various methods to create diverse design spaces and design datasets for ship hull design. These efforts aim to harness machine learning in ship hull design.

In addition to design representation, computational design often has metrics for evaluating a design's performance. Finding computationally efficient methods for evaluating each generated design could lead to enhanced design generation. Hydrodynamic drag stands out as the predominant performance metric for ships in literature. Several rapid drag prediction techniques exist. Some, like Hollenbach's and Savitsky's methods, rely on statistical regressions from test data [24–26]. Other fast methods to predict wave drag are linear wave solvers, which provide accurate drag measurements with reduced computational effort relative to traditional computational fluid dynamics techniques. These solvers use potential flow to simulate the waves produced by a ship in a steady forward motion to estimate drag as a result of surface wave propagation. Different linear wave solvers include Michell's Integral [27, 28], Rankine Panel Methods [29], Neumann-Kelvin Theory (also called Dawson's Method) [30], and Neumann-Michell Theory [31–33]. These potential flow solvers input the 3D geometry of a hull and provide estimates of drag at typical operating speeds of a hull. The third method of creating a fast prediction of drag is to build a small dataset of drag measures to train a neural network to predict drag from a hull's design representation [5, 6, 9, 13–16, 19, 20, 34, 35].

Combining hull design representation with efficient drag prediction equips designers with the tools needed for optimization algorithms. This enables hull design creation tailored for specific scenarios. A common objective in optimization literature is minimizing hull drag while adhering to geometric constraints. More recently, computational hull design has also been attempted using a tabular generative adversarial network (GAN) to quickly generate ship hull instances, that could be used for seeding populations for design optimization [21]. The next improvement for generative AI in ship design is to implement a denoising diffusion probabilistic model (DDPM) for generating hull designs. Diffusion models provide improvements over GANs for generative design as DDPMs are more stable to train and provide superior sampling quality. Additionally, diffusion models can implement guidance without retraining the whole generative model. This way, new constraints or performance objectives can be integrated into design generation simply, whereas a GAN would need to be retrained for every new design consideration. [36]. For ship hull design, this means that a single model can be trained to generate high quality hulls that are tailored to specific user needs by integrating guidance models for different design considerations. This is particularly useful for ship design so that information from the design of many classes of ships can be considered in designing a ship hull.

2.2 Generative Design with Diffusion Models

The transition from traditional design methods leads to a cutting-edge generative AI model: the denoising diffusion probabilistic model (DDPM). Gaining momentum in the machine learning domain, DDPMs iteratively modify a noisy data vector over many specified steps, transforming random data to mirror the statistics of training data [37]. The development of DDPMs in the last few years has shown that they are capable of generating complex data and already have applications for engineering design. For example, DDPMs were shown to create higher quality images as compared to generative adversarial networks [37], a particularly difficult task as images are comprised of large patterns of pixels to visually represent something a human could see with their eyes.

DDPMs work by training a neural network to predict small iterative denoising steps. The algorithm for training a diffusion model as defined by Ho et al. is in Table 1.

```
1: repeat
2:     $X_0 \sim q(X_0)$
3:     $t \sim Uniform(\{1, ..., T\})$
4:     $\epsilon \sim N(0, I)$
5:     Take gradient descent step on:
       $\nabla_\theta||\epsilon - \epsilon_\theta(\sqrt{\bar{a}_t}X_0 + \sqrt{1-\bar{a}_t}\epsilon, t)||^2$
6: until converged
```

Table 1: This is the training algorithm for a standard DDPM. The DDPM is represented by the function $\epsilon_\theta(X_0, \epsilon, t)$ in step 5.

In the algorithm, the generated sample (design parameters) are represented by X, and noted with subscripts to indicate the denoising timestep. The DDPM itself is represented by ϵ_θ, indicating that the DDPM is trained to predict a small change in random noise across the vector. Once trained, a DDPM generates samples by denoising a Gaussian noise vector over the predetermined timesteps. This results in samples that are within the training data's statistical distribution. In the case of images, this could be a "deep fake" that looks like the training data. In the case of ship hull design, it could be a parameterized ship hull design. The sampling algorithm defined by Ho et al. is defined in Table 2.

```
1: $X_T \sim N(0, I)$
2: for $t = T, ..., 1$ do
3:     $Z \sim N(0, I)$ if $t > 1$, else $z = 0$
4:     $X_{t-1} = \frac{1}{\sqrt{\alpha_t}}(X_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(X_t, t)) + \sigma_t Z$
5: end for
6: return $X_0$
```

Table 2: This is the sampling algorithm for a standard DDPM. The DDPM is represented by the function $\epsilon_\theta(X_t, t)$ in step 4.

Subsequent advancements in DDPMs introduced guidance, where gradients from a classifier neural network guide image synthesis to match a specific image classification label [36]. This evolution birthed text-to-image DDPMs that employ text-based guidance to craft custom, lifelike images [38, 39]. Guided DDPMs have found applications in generating 3D shapes from image data [40].

Guided diffusion can be applied to engineering design generation. For example, guided diffusion has been used to create two-dimensional structures [41–43] and vehicles [44] using image data. In these instances, the guidance of the design generation by image-based DDPMs is applied to constraint satisfaction and improved performance. DDPMs can generate high-quality designs, navigate complex constraints, and implement precise generation with guidance, which makes them an excellent deep generative model for designing ship hulls. The subsequent sections demonstrate a tabular DDPM to generate parametric ship hull designs that give improved performance through the implementation of guidance.

3 Methods

This section outlines the methodology behind developing a guided DDPM for ship hull design. This section explores the ship hull dataset, delves into tabular DDPMs, and introduces both classifier and performance guidance for sampling ship hulls with a DDPM. A secondary methods section on conditional DDPMs is included in the Appendix.

3.1 Ship-D Dataset and Hull Parameterization

The Ship-D dataset consists of 30,000 parameterized ship hulls. The hulls are parameterized with 45 terms. These terms are applied to a set of algebraic equations to define and characterize the surface of the hull. These terms were construed through analyzing and characterizing the shape and curvature of many different publicly available hull geometries. The parameters cover various aspects:

- Principal dimensions (e.g., overall length, beam at main deck)
- Cross-section of the parallel midbody (e.g., deadrise angle, chine radius)
- Geometry of bow and stern taper
- Geometry of bulbs at bow and stern

These parameters, designed to capture a range of curvature and shapes, encompass the features seen in a diverse variety of vessels from large ships to smaller boats. Their dual design facilitates human understanding and computer-generated input. Full documentation of the hull design parameters is provided at `https://decode.mit.edu/projects/ShipGen/`. Additionally, Figure 20 in the Appendix lists the parameters and provides details so human designers can create parametric hulls with this representation. A glimpse into the Ship-D dataset is provided in Figure 2, showcasing the diverse shapes achievable with the parametric design scheme. As these designs are randomly sampled across the entire feasible design space, they do not necessarily look like realistic hull designs. The performance of these hulls was not considered in their random sampling. Many of these hulls are relatively low performing: having high drag, low displacement volumes, and high surface area. The feasibility criteria used to generate these hulls are described in the next section.

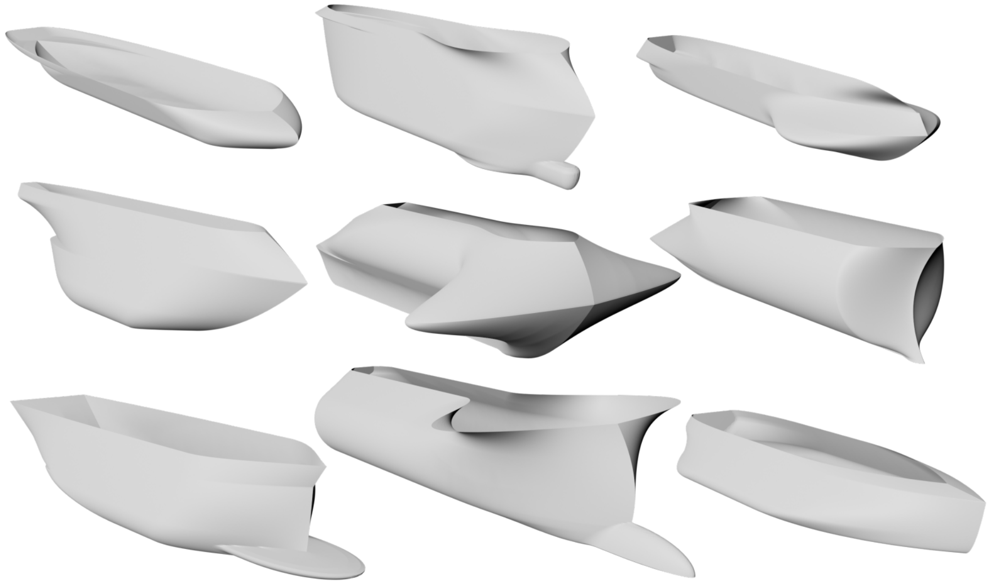


Figure 2: A selection of hulls from the Ship-D dataset, showing the variability possible with the hull parameterization. A random sampling from the dataset may lead to unrealistic hulls, containing combinations of features that do not resemble real-world ships and features that lead to poor performance.

3.1.1 Feasibility Constraints for Hull Geometry

While the parameterization can define a large design space of hull geometries, constraints on the parameterization are needed to ensure that a feasible hull will be produced by a specific set of parameters. To satisfy a "feasible" hull shape, the hull's surface only needs to satisfy two criteria:

1. The hull is watertight, meaning that there are no holes on its surface.
2. The hull surface is not self-intersecting.

As the hull surface is defined by a set of equations with constants dictated by the parameter values, conditions to determine whether a hull's surface satisfies the two main feasibility criteria can be solved algebraically. The advantage to algebraically solving these conditions is significantly reduced computational effort to check hull feasibility with the algebraic constraints compared to feasibility checks with mesh generation. After searching through the design space of the hull parameterization and examining the equations that define the hull surface, a set of forty nine constraints were defined to determine if a hull surface produced from a specific parameterization satisfies the two feasibility criteria. Figure 21 in the Appendix lists the 49 algebraic constraints and provides information on each of their satisfaction conditions.

Conversely, the two feasibility criteria can be checked by constructing the mesh of a hull and analyzing its surface. Mesh generation and feasibility checks are computed in $O(Nlog(N))$, where N is the number of vertices on the mesh. On an Intel Core i9-10980XE processor, the construction and check of a hull mesh with approximately 80,000 vertices is 1.77 seconds. Comparatively, the algebraic constraints check the design feasibility of a parametric hull in 0.000199 seconds. This is a ten-thousand-fold increase in speed for checking design feasibility with the algebraic constraints. A uniform random sampling of the design parameters leads to generation of a feasible hull in approximately 1 per 150 tries. In addition to the 30,000 feasible hulls in the Ship-D dataset, an additional 20,000 design vectors (called invalid samples) that violate at least one feasibility constraint were generated. These invalid samples are used to train models in classifying and distinguishing between feasible and infeasible design vectors [41].

3.1.2 Hull Performance Measures

The Ship-D dataset already contains ten geometric measures and thirty-two wave drag calculations for each hull. The ten geometric measures allow naval architects to characterize a hull when designing a ship. The ten geometric measures are calculated using trapezoidal integration at ten draft marks spaced along the depth of the hull. These ten geometric measures are:

1. Height of draftmark
2. Length of the waterline
3. Area of the waterplane
4. Surface area of the hull below the specified draftmark (wetted surface)
5. Longitudinal centers of flotation (waterplane centroid)
6. Second moment of area about the longitudinal axis of the waterplane
7. Second moment of area about the transverse axis of the waterplane
8. Displaced volume below the draftmark
9. Longitudinal center of buoyancy
10. Vertical center of buoyancy

As these metrics have the units of length to some power L^n, they are normalized by the first term in the parameterization, LOA, to its respective power. For example, lengths are normalized by LOA, areas by LOA^2, volumes by LOA^3, and area moments of inertia by LOA^4. This allows computational analysis on the geometry of the hull to be performed independently of the hull's scale.

In addition to the geometric measures, the Ship-D dataset has thirty-two wave drag coefficients for each hull across four different drafts and eight velocity conditions. The four drafts are 25%, 33%, 50%, and 67% of the hull's total depth. The eight velocity conditions are normalized using Froude scaling. The eight velocities are between $F_n = 0.10$ and $F_n = 0.45$ in increments of 0.05, corresponding to typical operating conditions of traditional displacement hulls [45, 46]. The Froude number is the relative scaling between inertial and gravitational forces described in the equation below:

$$F_n = \frac{U}{\sqrt{gL}} \tag{1}$$

Where U is the hull speed, g is gravity and L is a length scale. The length used in simulating the 32 speed-draft conditions of the hulls was the length of the waterline at the tested draft mark. This way, thirty two unique conditions were measured. Wave drag is both a function of the hulls geometry, and the hydrodynamics of waves propagating off of the hull from it's forward motion. Including a full spectrum of speed-draft conditions in the dataset allows a machine learning model to learn the effects of drag due to changing submerged geometry with draft and speed. This provides significantly more information relating to the geometry and performance of a hull than available by measuring a single operating condition. This allows a generative model using the Ship-D dataset to generalize wave drag in the design process. As the generative model is intended to produce conceptual hull designs, it is imperative that the exact speed-draft condition be unknown so that the model generates hulls that generally have low drag. The future work section will detail goals for generating hull designs tailored to specific use cases, which could include specific speed-draft conditions.

The Michell Integral was chosen to simulate wave drag over other linear wave methods for its relative computational efficiency and the accuracy it provides. The Michell integral is a linear estimate of the wave drag of a slender ship in forward motion. It is defined by the following equation [27, 28]:

$$R_w = \frac{A\rho g^2}{\pi U^2} \int_1^\infty (I^2 + J^2) \frac{\lambda^2}{\sqrt{\lambda^2 - 1}} \, d\lambda \tag{2}$$

where ρ is the density of water, g is gravitational acceleration, U is the ship's speed, and $A, I, andJ$ are integrated terms relating to the surface normal across the hull and the direction of wave propagation. Further insight into these terms is in Michell's paper form 1898 [27].

In addition to scaling the relative speed and draft conditions for the hulls, the wave drag is also scaled to ensure consistency across the dataset:

$$C_w = \frac{R_w}{\frac{1}{2}\rho U^2 \cdot \mathrm{LOA}^2} \tag{3}$$

Typical drag coefficients of hulls are scaled by the wetted surface area of the hull. Within the dataset, however, the wetted surface area of the hulls can vary greatly. Instead, the Length-Overall (LOA) is used instead as this is the first term in the parameterization. For the purposes of applying machine learning using the dataset, the wave drag coefficient can be characterized by the remaining 44 terms in the parameterization and the hull's relative speed and draft.

An additional two measures of the hulls are included in this paper and will be added to the Ship-D dataset. The first measure is the Gaussian curvature of the hull's surface. The second metric is a measure of the largest rectangular prism that can be vertically lowered into the hull, referred to as MaxBox for the remainder of this paper. Gaussian curvature quantifies the double curvature of a surface. The average Gaussian curvature is calculated for these hulls to assess the manufacturing complexity of the hull's surface. As most large ships are constructed from welded sheet steel or aluminum, bending a sheet along two principal axes of curvature is a difficult task for both the sheet forming process and for welding the edge of a complex surface to another. By measuring the average double curvature of each hull, an understanding of the difficulty of manufacturing the hull surface is gained for the dataset. The Gaussian curvature is calculated for the hulls using a finite difference method to measure the principal curvature of the hull in the YZ plane and in the XY plane for a uniform grid of points on the hull [47]. Equation 4 calculates the average Gaussian curvature over the surface of the hull. The terms R_{XY} and R_{YZ} are the radii calculated using the finite difference method along the two principal axes of hull's surface. Gaussian curvature has units $1/L^2$ and is hence normalized by LOA^2

$$GC = \frac{\oiint_S \frac{\mathrm{d}A}{R_{XY}(x,y,z) \cdot R_{YZ}(x,y,z)}}{\text{Total Surface Area}} \tag{4}$$

The MaxBox measures the box with maximum volume that is completely inscribed by the hull that can be vertically lowered into the hull through the waterplane at the hull's top deck. This provides a measure for evaluating a candidate region within the ship hull for allocating cargo holds. Additionally, as the MaxBox is open at the deck of the ship, a crane can service this entire volume within each hull. The MaxBox for each hull was optimized with a Nelder-Mead simplex optimization algorithm to maximize the volume of the box while constrained by the surface of the hull and the waterplane of the top deck [48]. Included in the dataset is the forward (X) position of the box, its length, width, depth, and volume. These results are normalized by their length dimensionality, $1/LOA$ and $1/LOA^3$.

Among the available performance measures in the Ship-D dataset, seven were selected to be implemented in the performance-driven design generation of ship hulls. The goal of selecting the seven performance metrics that generally describe the quality of a hull. These metrics provide an avenue to compare hulls directly to each other with useful characteristics that consider the hulls' hydrodynamics, hydrostatics, and manufacturability. These seven performance metrics are:

1. Aggregated sum of wave drag coefficients.
2. Surface area of the hull up to 50% of its total depth
3. Total surface area of the hull
4. Displaced volume of the hull up to 50% of its total depth
5. Total displaced volume of the hull
6. Volume of the MaxBox
7. Gaussian curvature

The aggregated sum of wave drag coefficients was selected as a way to quickly characterize the general wave drag of a given hull. In large ships, wave drag is the primary component in the ship's total drag. By learning how a ship's hull shape affects drag, a generative AI model could generate hulls with low wave drag, saving ship operation costs through reduced fuel consumption. The aggregated sum of the wave drag coefficients is defined in Equation 5. It is important to note that this performance metric and five of the other metrics are all represented on a logarithmic scale. Due to the geometry of the hull designs these performance metrics span several orders of magnitude across the Ship-D dataset. The distribution of the logarithmic scaled performance metrics is normal, a desired quality for machine learning.

$$C_{w*} = \sum_{i=1}^{32} \log_{10}(C_{w_i}) \tag{5}$$

The surface area of the lower half of the hull was selected as a performance measure as this is the portion of a hull's surface that is most likely to be submerged when the hull is in water. The wetted surface of the ship affects the viscous drag acting on the hull. Reducing the wetted surface area of a hull can reduce the total drag of a ship, saving operation costs through reduced fuel consumption. Additionally, the total surface area of the hull was selected as this can consider the amount of material needed to manufacture the surface of the ship hull. By reducing the total surface area of the ship, manufacturing costs can be reduced by generating hull designs with less total surface area. The two measures of surface area are provided in Equation 6 and Equation 7.

$$SA_{50\%*} = \log_{10}\left(\frac{\int_0^{T/D_d=0.5} \delta SA(z)\,\delta z}{LOA^2}\right) \tag{6}$$

$$SA_{100\%*} = \log_{10}\left(\frac{\int_0^{T/D_d=1.0} \delta SA(z)\,\delta z}{LOA^2}\right) \tag{7}$$

The displaced volume of the bottom half of the hull was selected as it characterizes the portion of the hull that contains much of the displaced volume for buoyant forces. This performance metric therefore characterizes the relative total weight of the ship and its cargo. Learning how the hull design parameters affect the displaced volume of the hull can lead to generating hull designs that can carry more weight. The total displaced volume of the hull was also selected as this measure characterizes the total volume capacity available for cargo, outfitting, and other systems on the ship. Learning how the ship hull design parameters affect its total displaced volume, ship hulls with greater total volume can be generated. These two measures of volume affect the ability of ships to generate revenue through the shipment of cargo. The two volume measures are calculated with Equation 8 and Equation 9. With the intention of maximizing the volume metrics in hull generation, the volume measures are multiplied by -1. This conforms the volume maximization problem to a "minimization" problem akin to the other performance objectives.

$$V_{50\%*} = -\log_{10}\left(\frac{\int_0^{T/D_d=0.5} \delta V(z)\,\delta z}{\mathrm{LOA}^3}\right) \tag{8}$$

$$V_{100\%*} = -\log_{10}\left(\frac{\int_0^{T/D_d=1.0} \delta V(z)\,\delta z}{\mathrm{LOA}^3}\right) \tag{9}$$

An additional measure of volume is the MaxBox volume. As described earlier in this section, the MaxBox metric measures the ratio of the most useful cargo-carrying volume of the ship compared to the hull's total displaced volume. Learning how the design parameters affect MaxBox can lead to the generation of hulls with more useful cargo-carrying capacity. This can also lead to greater revenue through a ship's operation. MaxBox is not on a logarithmic scale like the other measures and it is calculated with Equation 10.

$$\mathrm{MaxBox}_* = -\frac{\mathrm{Volume}_{\mathrm{MaxBox}}}{\mathrm{Volume}_{T/D_d=1.0}} \tag{10}$$

The final performance metric selected is the average Gaussian curvature of the hull. Since Gaussian curvature is a measure of a hull's surface complexity, it affects the manufacturing costs of a ship. Reducing the average Gaussian curvature of a ship hull can lead to reduced manufacturing costs, making it a critical metric for a ship hull. The average Gaussian curvature is normalized for machine learning using Equation 11.

$$GC_* = \log_{10}(GC \cdot LOA^2) \tag{11}$$

While the aforementioned seven metrics were chosen to demonstrate the efficacy of the proposed methodology, it is crucial to highlight that this is not an exhaustive list of performance measures for ship hull evaluation. Indeed, a significant strength of the proposed diffusion model lies in its adaptability. It allows users to integrate additional performance metrics without necessitating retraining. This flexibility underscores the model's robustness and its potential to be tailored to various specific needs, optimizing designs based on a myriad of performance criteria.

3.2 Dataset Coverage and Generated Sample Evaluation

In order to characterize the DDPMs' abilities to cover the total parametric dataset space and generate feasible designs, two measures are utilized throughout the remainder of the paper. To visually characterize how a set of generated hull designs covers the dataset space of the Ship-D hulls, a two-dimensional principal component analysis is trained with the Ship-D parametric design vectors. When evaluating the designs generated with DDPMs, the PCA of the generated samples is plotted against the PCA of a random selection of the Ship-D dataset hulls. This shows the relative spread of the generated designs compared to the dataset hulls. In addition to visualization, coverage and realism quantify a model's ability to generate samples similar to the training dataset.

- Coverage is quantified as the mean chamfer distance of each dataset instance from its nearest neighbor among the generated samples.
- Realism, on the other hand, measures the mean chamfer distance of each generated sample instance from its closest match within the dataset [49].

Chamfer distance is the Euclidean distance between two hull design vectors. For two sets of parameterized hull designs, A and B, the Chamfer distance finds the distances from a design vector in A to its nearest neighbor in B. The distance metric used is the squared Euclidean distance between the two vectors. The formula for this evaluation metric is:

$$CD = \|A_n - B_{n*}\|^2 \tag{12}$$

where B_{n*} is the nearest neighbor of the design vector A_n in B. Chamfer distance is normalized for coverage and realism with the following equation:

$$CD_* = \frac{\frac{1}{N_A}\sum_{n=1}^{N_A} CD_n - CD_{\text{worst-case}}}{CD_{\text{best-case}} - CD_{\text{worst-case}}} \tag{13}$$

where N_A is the number of points in set A and CD_* is a normalized Chamfer distance, being coverage or realism. As a baseline, coverage is linearly normalized between the best- and worst-case scenarios derived from an analysis of the dataset. The best-case coverage is the mean nearest neighbor distance of each hull in the dataset (100% coverage), which is equal to 4.315. The worst-case coverage is the mean chamfer distance between the centroid of the dataset hulls and each hull in the dataset, (0% coverage), which is equal to 26.930. realism is also linearly normalized to be between the minimum and maximum chamfer distance between the dataset hulls and their nearest neighbor.

To benchmark the DDPMs' ability to generate feasible samples, two studies on the feasibility of hulls generated by interpolating between hull design vectors were conducted. The first study generated sample hulls by finding the midpoint between two random hull design vectors from the Ship-D dataset. The second study generated sample hulls by finding the midpoint between a random hull and its nearest neighbor in the Ship-D dataset. The results of these studies are provided in the Results Section.

An additional benchmark study was conducted using a tabular generative adversarial network called CTGAN [50]. The CTGAN was trained to generate feasible hull designs, only gathering information from the 30,000 feasible hull designs in the Ship-D dataset. The goal of this benchmark study is to compare the ability of the CTGAN and the DDPM to generate feasible hulls and cover the dataset without explicitly identifying feasible or infeasible hull designs for the model. In training, the CTGAN learns the parametric information encoded in the design vectors and generates samples to match the distribution of the dataset samples. The results of this study are included in Table 3 and Table 4.

When leveraging classifier guidance with a DDPM to generate feasible samples, the dataset coverage is greatly affected by the same hyperparameter that influences the feasibility of generated samples.

3.3 Denoising Diffusion Probabilistic Models

A denoising diffusion probabilistic model (DDPM) is a generative AI model that generates new instances of data by denoising random information over many steps, so that the generated sample falls within the statistical distribution of the training dataset samples. A tabular DDPM was built and trained on the ship hull parametric design information from the Ship-D dataset. The DDPM used to create ShipGen was inspired by the work of Kotelnikov et.al, called TabDDPM [51]. Unlike popular image-focused diffusion models, this DDPM is trained on tabular information to generate tabular information. Prior to training, the parametric design vectors were transformed with a quantile normalizer to re-scale the distribution of the design parameters to have a normal distribution with the same mean and variance as the parameters in the dataset. A second linear transformation re-scaled the bounds so that the range of each parameter exists between -1 and 1. These transformations ensure that the parametric design data is fit for the tabular DDPM. Training this model provided a baseline to verify that the tabular DDPM produces ship hulls with parametric information within the relative distribution of the Ship-D dataset. The Results Section provides the results of both the parameter distribution and feasibility constraint satisfaction of these generated samples.

3.3.1 Standard Diffusion Model

The standard DDPM follows the training and sampling algorithms defined in Table 1 and Table 2. The standard DDPM implicitly learns the statistical relationships between the parameters in each sample. In the sampling process, the trained DDPM generates samples that are statistically similar to the designs in the dataset. There is no extra consideration for design feasibility or design performance. Since the Ship-D dataset is comprised of randomly sampled hulls that meet the feasibility criteria, any increase in the DDPMs ability to produce feasible hulls compared to pure random sampling is due to the DDPM implicitly learning the relationships between the design parameters that lead to feasible hull designs.

During training, a feasible design vector is quantile normalized and partially noised according to the training algorithm in Table 1. Then the DDPM predicts a noised vector given the timestep embedding and the partially noised vector. The mean squared difference between the predicted noise vector and a pure noise vector is the loss of this prediction. The mean squared loss then back-propagates through the DDPM to update its weights and biases. This process is repeated for the 30,000 feasible design vectors across one thousand denoising timesteps in random batches to train the DDPM. Figure 3 illustrates the training process for one design vector at one timestep.

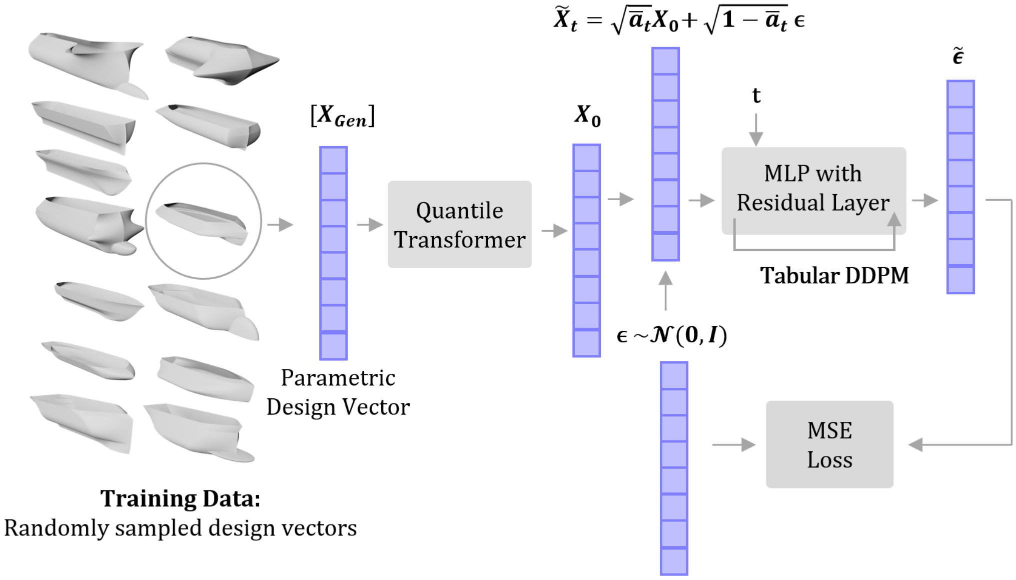

$\tilde{X}_t = \sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon$
$\tilde{\epsilon}$
$[X_{Gen}]$
X_0
t
MLP with Residual Layer
Quantile Transformer
Tabular DDPM
$\epsilon \sim \mathcal{N}(0, I)$
Parametric Design Vector
MSE Loss
Training Data:
Randomly sampled design vectors

Figure 3: During training, the DDPM predicts a denoising step, given a timestep embedding and a partially noised sample design vector.

After training, the standard DDPM can sample new design vectors. The initial seed for sampling is a Gaussian noise vector of size N, where N is the number of design parameters. The DDPM denoises this vector one thousand times, taking into account the timestep embedding at each iteration. After the denoising process, the final denoised vector is reverse-quantile normalized so that it becomes a design vector fitting the Ship-D parametric design scheme. This generated design vector can then be checked for feasibility constraint satisfaction. If the design is feasible, a point cloud and mesh of the hull is generated to evaluate the sample's performance. Figure 4 illustrates the sampling process for a single design vector.

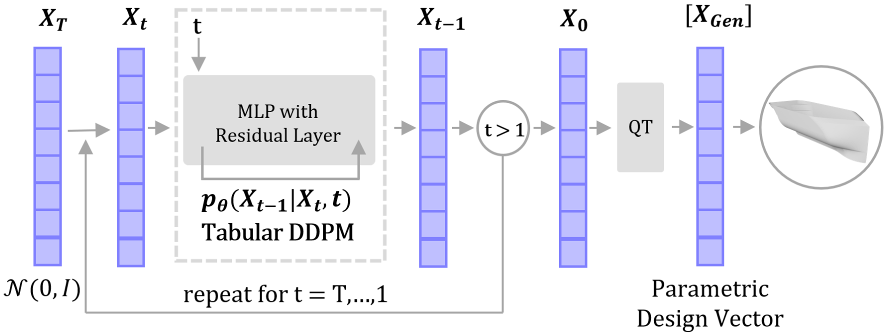


Figure 4: During sampling, the standard DDPM denoises a vector over one thousand timesteps, generating a sample design vector that statistically aligns with the training data.

3.3.2 Classifier Guidance for Diffusion Models

An additional method of influencing sample generation towards feasibility constraint satisfaction is with classifier guidance [36]. Classifier guidance leverages the gradients of a trained design classifier during the standard DDPM's sampling process to influence a design to meet a certain classifier label. In this case, the classifier label characterizes whether a design is feasible or infeasible. Here, the 30,000 Ship-D design vectors and the 20,000 infeasible design vectors trained a classifier to predict design feasibility. At each timestep in the sampling process, the gradient of the trained classifier for a target class, $f_\phi(y|X_t)$, with respect to the parameterized design vector, X_t, is calculated. This gradient is multiplied by a hyperparameter, γ, and is added to the sample during Step 4 of the DDPM sampling algorithm defined in Table 2. A classifier guided DDPM is created by replacing Step 4 with Equation 14.

$$X_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(X_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(X_t, t)\right) + \sigma_t(Z(1-\gamma)) + \gamma\nabla_{X_t} f_\phi(y|X_t) \tag{14}$$

The Results Section provides data from tuning γ, as it will be shown that the hyperparameter has an effect on both the likelihood of producing feasible hull design vectors and on the distribution of these vectors relative to the Ship-D dataset. Without the need to additionally train the standard DDPM itself, adding guidance gradients in the sampling process can be accomplished easily. Figure 5 and Figure 6 illustrate classifier guidance used in conjunction with performance guidance to generate parameterized hull designs with high performance. The next subsection details the addition of more guidance models to generate hulls while considering the hull's performance.

3.3.3 Performance Guidance for Diffusion Models

Similar to classifier guidance, a neural network trained to predict the performance of a hull can also be used to guide sample generation. Seven residual neural networks were trained to predict the normalized performance of a hull given its parametric design vector. The 30,000 feasible hull designs in the ShipD dataset were used for the training data. The performance prediction neural networks all have the same structure: 4 hidden layers with 256 nodes, where the first hidden layer is added as a residual to the final hidden layer. The normalization of the performance metrics distributes them over a Gaussian, improving the prediction accuracy of the neural network.

During sampling, the gradients of the normalized performance prediction from the neural networks is used to guide the DDPM's sampling process. The performance gradient of each of the objectives, $\nabla_{X_t} P_i(X_t)$ is multiplied by a weight, λ_i. While generating samples, the weights of the λ values for each performance objective are normalized so that they are positive and sum to 1.0 for each sample. This way, a broad spectrum of samples is generated with unique combinations of weighted influences from the seven performance objectives. Performance guidance is achieved by replacing Step 4 in the DDPM sampling algorithm with Equation 15.

$$X_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(X_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(X_t, t)\right) + \sigma_t(Z(1-\gamma)) + \gamma\nabla_{X_t} f_\phi(y|X_t) - \sum_{i=1}^{7}\lambda_i \nabla_{X_t} P_i(X_t) \tag{15}$$

During sampling, the gradients of both the classifier and performance prediction models are calculated at each timestep. Weighting these gradients with γ and λ influences the impact each individual model has on the sampling process. The classifier guidance weight, γ, is set equal to 0.5 so that both a high degree of sample diversity and sample feasibility are maintained. Figure 5 shows how guidance from the classifier and performance prediction models are implemented into the denoising process.

Figure 6 highlights the contributions of both the feasibility classifier and the performance prediction neural networks in guidance. For performance guidance, the gradients of the seven performance prediction networks are calculated for each sampling timestep for X_t. Then, the gradients are weighted by their respective λ value. The sum of the weighted gradients is subtracted from the output of the standard DDPM to create the next partially denoised vector, X_{t-1}, in the sampling process. The gradients are subtracted to follow the scheme of "minimizing" the performance objectives in generated samples.

After sampling, the generated design vectors are checked for feasibility. The performance of the feasible generated hulls are then calculated with the same simulations used to create the original dataset. In the Results Section, Table 5 will showcase the mean normalized performance between the dataset hulls and the generated hulls, as well as a scale factor between the true performance of the two sets of hulls. The Results Section also provides data on the fit of these performance regression networks.

4 Results

This section contains the results of the studies described in the Methods Section. The first subsection provides results on the feasibility and design spread of parameterized hulls generated with interpolation between existing hulls in the Ship-D dataset. The second subsection gives the results of generating feasible hulls using a standard tabular DDPM and with a guided DDPM. The third subsection provides the results on generating hulls using performance guidance, including the results of the performance prediction residual neural networks. The Appendix contains the results of training a conditional DDPM with both feasible and invalid hulls.

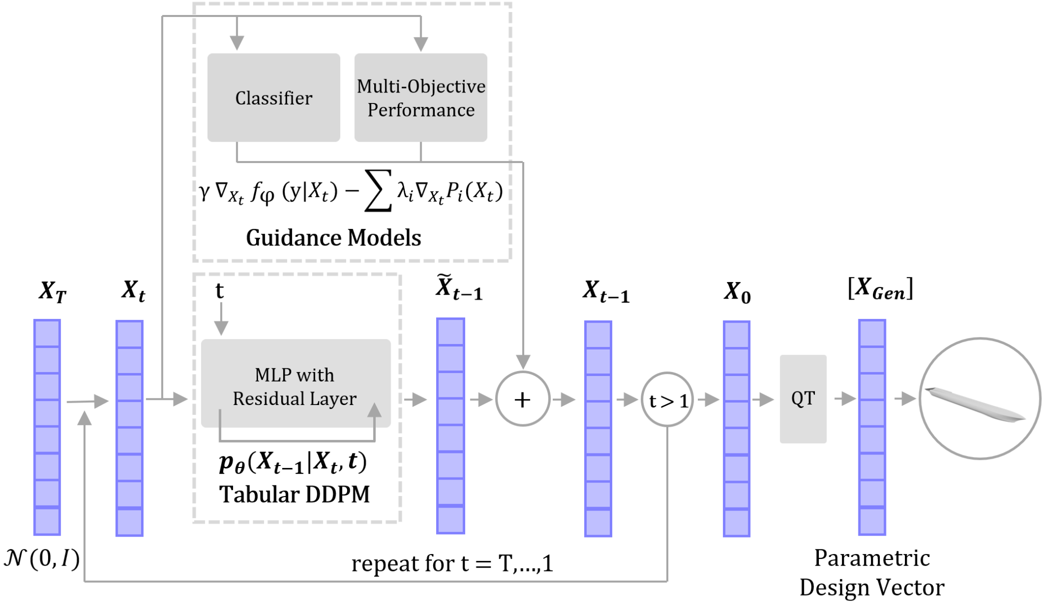


Figure 5: For guided sampling, generated vectors are denoised with the standard DDPM at each timestep and then influenced with guidance gradients in each denoisng iteration.

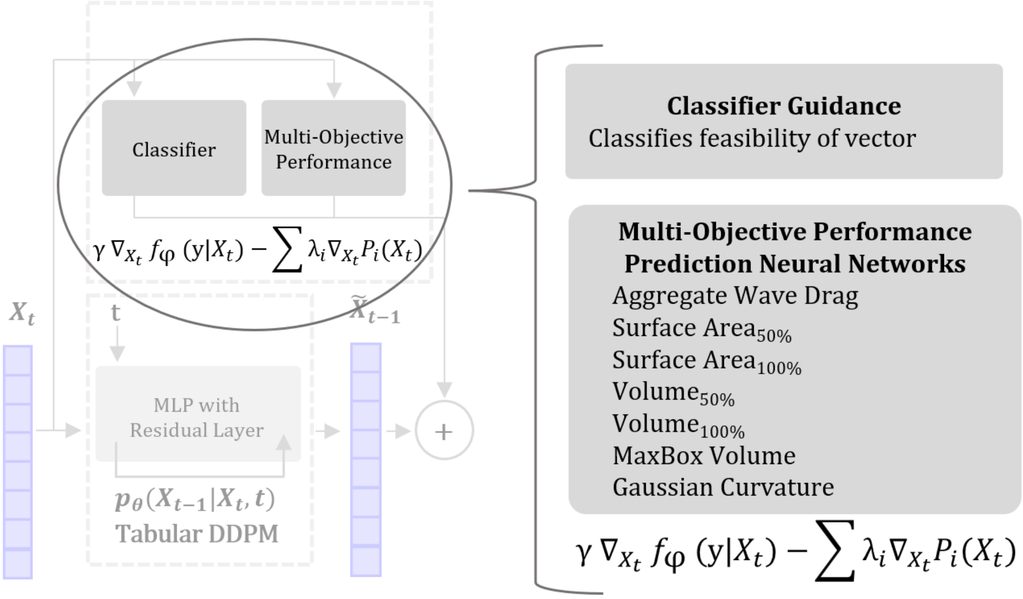

Classifier
Multi-Objective Performance
$\gamma \nabla_{X_t} f_\varphi (y|X_t) - \sum \lambda_i \nabla_{X_t} P_i(X_t)$
X_t
t
$\tilde{X}_{t-1}$
MLP with Residual Layer
$p_\theta(X_{t-1}|X_t, t)$
Tabular DDPM
+
Classifier Guidance
Classifies feasibility of vector
Multi-Objective Performance Prediction Neural Networks
Aggregate Wave Drag
Surface Area$_{50\%}$
Surface Area$_{100\%}$
Volume$_{50\%}$
Volume$_{100\%}$
MaxBox Volume
Gaussian Curvature
$\gamma \nabla_{X_t} f_\varphi (y|X_t) - \sum \lambda_i \nabla_{X_t} P_i(X_t)$

Figure 6: Classifier and performance guidance is the result of leveraging gradients of pre-trained models to influence the denoising process of the DDPM. The figure highlights the models implemented for this experiment.

4.1 Benchmark Feasibility Constraint Satisfaction Studies

An initial study generating hulls using interpolation methods was conducted to measure the success rate of generating feasible hulls using the Ship-D dataset. The first study generated thirty thousand parametric hulls by interpolating the parameters halfway between two random hulls belonging to the Ship-D dataset. This interpolation method generated feasible hulls at a rate of 93.1%, listed in Table 4. The second interpolation method generated hulls by interpolating between a dataset hull and its nearest neighbor hull. The second interpolation method generated feasible hulls, with a success rate of 93.8%. Table 3 lists the dataset coverage of these two interpolation methods. The first interpolation method maintains a normalized coverage ratio of 0.965 compared to the baseline coverage, while the second interpolation method exceeds the baseline coverage, having a ratio of 1.059. These two interpolation methods will serve as benchmarks for the feasibility and dataset coverage analysis on hulls generated with the DDPM.

The CTGAN benchmark study provided a baseline for a trained generative model to generate feasible hulls implicitly. The CTGAN was trained on the 30,000 feasible hull designs Ship-D dataset to implicitly learn the combinations of parameter values that define "feasibility". This study generated thirty thousand samples of hulls and measured the dataset coverage and feasibility of these samples. The sample coverage was decreased with the CTGAN, having a normalized coverage ratio of 0.94 compared to the baseline. This is slightly reduced from the interpolation benchmark coverage ratios of 0.965 and 1.059. The coverage measures for the CTGAN study are included in Table 3. Of the CTGAN generated samples, only 0.7% satisfy all feasibility constraints. This finding is included in Table 4. The CTGAN is only marginally better than randomly sampling the design space to create a feasible parametric hull design. Further analysis of the CTGAN benchmark study is included in the Discussion Section.

4.2 Feasibility Constraint Satisfaction with Tabular Denoising Diffusion Probabilistic Models

This subsection provides the design feasibility and dataset coverage of samples generated with different types of DDPMs. The types of DDPMs considered for sample generation are the standard DDPM and guided DDPM. The following subsections provide the results for each type of DDPM.

4.2.1 Standard DDPM Leads to Good Feasibility and Coverage

A standard DDPM is trained only on the parametric design information from the dataset. Samples generated from a standard DDPM are made up of the implicit statistical relationships learned from the parameters in feasible hulls. The standard DDPM produces feasible hulls 51.1% of the time, as seen in Table 4. Throughout the Results Section, a two-dimensional principal component analysis (PCA) is used to illustrate the spread of generated sample hulls compared to the Ship-D dataset hulls. The PCA is trained on the parametric hull design data from the Ship-D dataset and is used to transform generated samples into the two-dimensional PCA for visualization. Figure 7 shows that the standard DDPM generates samples that maintain most of the dataset coverage, maintaining a normalized coverage ratio of 0.984. The dataset coverage and feasibility of samples created with the standard diffusion model are included in Table 3 and Table 4.

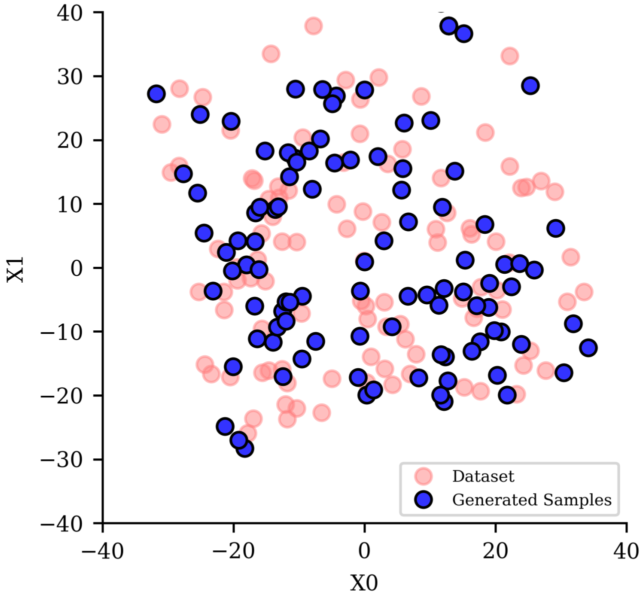


Figure 7: Two-dimensional Principal Component Analysis of the hull parameterization shows that hulls generated with a standard DDPM maintain most of the dataset coverage.

4.2.2 Guided Denoising Diffusion Probabilistic Model for Enhanced Feasibility

While the standard DDPM generates feasible hulls with a relatively high success rate, feasibility can be improved by leveraging guidance from a pre-trained classifier neural network. The classifier identifies hulls as satisfying all the constraints or violating at least one of the forty nine constraints. This classifier network was implemented in the denoising step of generating samples with a standard DDPM to guide the generation of hulls towards satisfying the feasibility criteria. As mentioned in the Methods Section, the degree to which the guidance influences sample denoising is tied to a hyperparameter, γ. Figure 8 shows the percentage of generated feasible samples among generated samples versus γ. Note that when $\gamma = 0$, the denoising process is the same as the standard DDPM. In addition to design feasibility, γ also affects the dataset coverage of the generated samples, as shown in Figure 9. As defined in the Methods Section, generated samples have increased realism with a generated sample by decreasing the Chamfer distance to its nearest neighbor belonging to the dataset of designs. Similarly, the generated samples have increasing dataset coverage with decreasing distance of every dataset point to its nearest neighbor belonging to the generated samples. Realism and coverage are measured as the mean normalized chamfer distance between the generated hulls and the dataset hulls. Table 3 quantifies coverage, showing that increasing γ reduces the dataset coverage substantially. To maintain dataset coverage similar to the interpolation studies, γ should be less than or equal to 0.35. To balance both design feasibility and dataset coverage among generated samples, γ is set to 0.5. This way, feasible samples are generated 99.5% of the time and maintain a dataset coverage ratio greater than 0.9. The remaining plots in this subsection capture a snapshot of samples generated with guided diffusion with γ set to 0.2, 0.35, 0.5, 0.65, 0.80, and 1.0. Table 4 shows the trend of both increasing success in generating feasible hulls. The PCA charts in Figure 10 illustrate the reduction in coverage with increasing γ. Figure 11 shows that classifier guidance has a significant influence on the feasibility of generated samples throughout the denoising process.

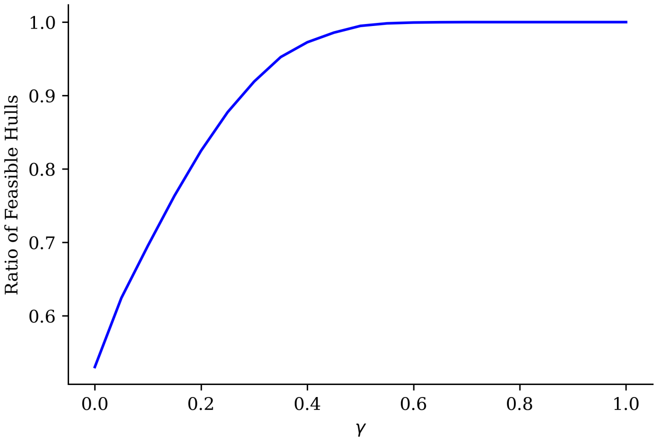


Figure 8: Hull design feasibility is highly dependent on γ in guided diffusion. The percentage of feasible generated hulls is above 90% when γ is greater than 0.3

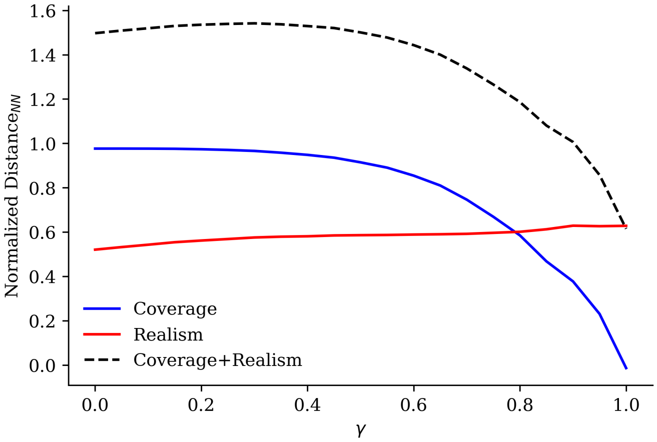


Figure 9: Realism and coverage of the generated samples is strongly affected by γ. When γ is approximately 0.5, the sum of realism and coverage is maximized.

Generation Method	**Chamfer Distance** *(Lower is Better)*	**Normalized Coverage** *(Higher is Better)*
Random Dataset Sample	4.315	(Baseline) 1.000
Interpolation Study 1	5.099	0.965
Interpolation Study 2	2.976	1.059
CTGAN	**5.660**	**0.940**
Standard DDPM	**4.672**	**0.984**
Guidance: $\gamma = 0.2$	4.731	0.982
Guidance: $\gamma = 0.35$	5.067	0.967
Guidance: $\gamma = 0.5$	6.002	0.925
Guidance: $\gamma = 0.65$	8.453	0.817
Guidance :$\gamma = 0.8$	13.611	0.589
Guidance: $\gamma = 1.0$	27.054	-0.005

Table 3: The table provides the dataset coverage for the different sampling methods. These values are normalized between the best- and worst-case scenarios found in the dataset. The standard DDPM covers the dataset better than the CTGAN. By adding guidance to the DDPM, dataset coverage is maintained when $\gamma \leq 0.5$.

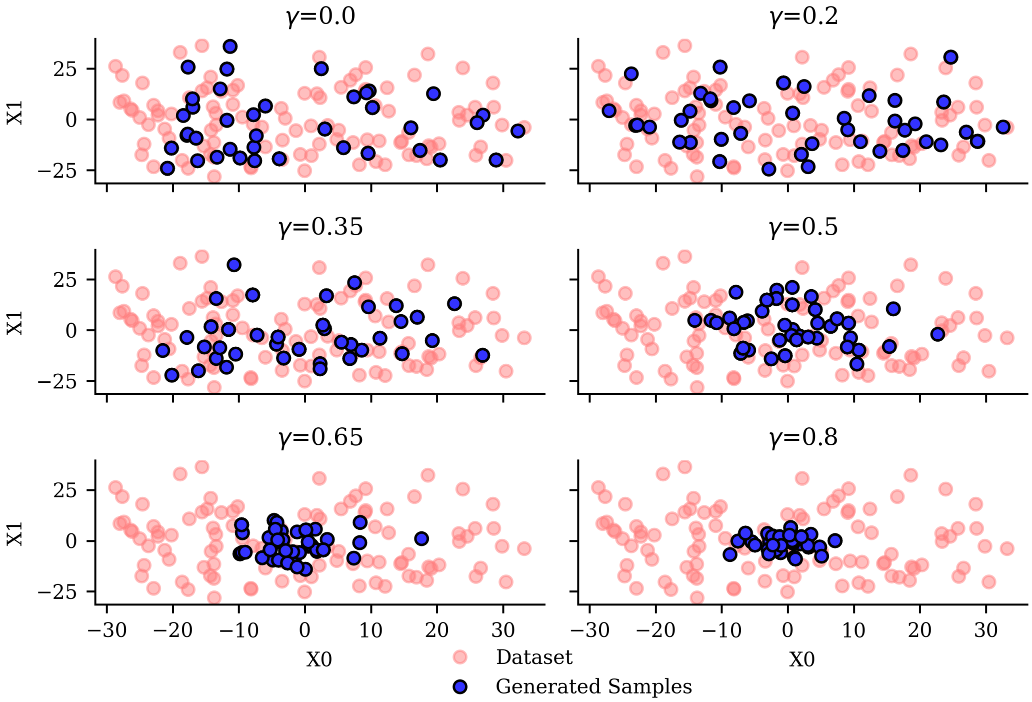


Figure 10: Two-dimensional Principal Component Analysis of the hull parameterization shows that the dataset coverage is reduced by increasing the hyperparameter, γ

4.3 Hull Generation with Performance Guided Denoising Diffusion Probabilistic Model

In addition to generating feasible samples, guidance can also generate high-performing parametric hull designs. The following subsections provide the results from training performance prediction neural networks on seven objectives and the results from measuring and simulating hulls generated using multi-objective performance guidance.

4.3.1 Performance Prediction Training

Using the performance data from the Ship-D dataset, seven residual neural networks were trained to predict the performance of the hulls given the parameterized design vector. Table 5 summarizes the results of the training, using R^2 as a measure of the goodness of fit for these neural networks. Figure 12 shows the plot of the regression prediction versus the simulation calculation for the aggregate wave drag measurement.

Generation Method	Feasibility Rate
Interpolation Study 1	0.931
Interpolation Study 2	0.938
CTGAN	**0.007**
Standard DDPM	**0.511**
Guidance: $\gamma = 0.2$	0.839
Guidance: $\gamma = 0.35$	0.962
Guidance: $\gamma = 0.5$	0.995
Guidance: $\gamma = 0.65$	1.000
Guidance: $\gamma = 0.8$	1.000
Guidance: $\gamma = 1.0$	1.000

Table 4: The table shows the fraction of generated samples that are feasible. The standard DDPM generates feasible samples 73x more often than CTGAN. Increasing γ increases the proportion of feasible samples. All generated samples are feasible when $\gamma \geq 0.65$

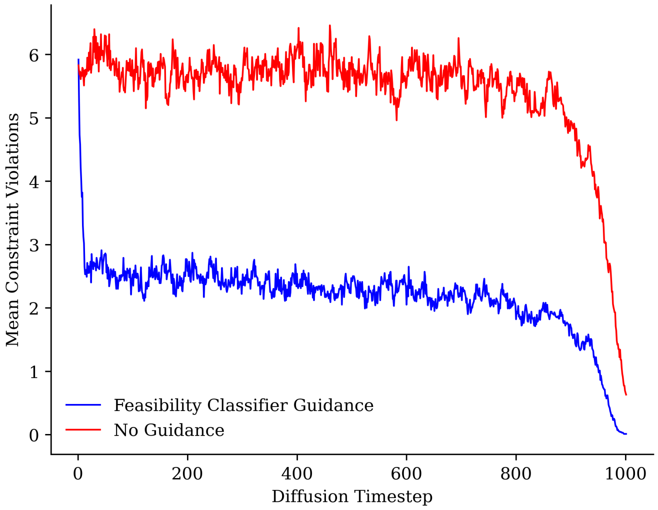


Figure 11: Leveraging classifier guidance improves the feasibility of generated samples throughout the denoising process compared to the standard DDPM (no guidance). The classifier guidance is tuned to $\gamma = 0.5$.

Figure 13 shows the same plots for the remaining six performance metrics. The blue dashed line in these figures represents the perfect regression prediction, exactly aligning with the simulation calculation. The wave drag coefficient, surface area, and volume prediction neural networks have high R^2 fits and hug the blue dashed line closely. The MaxBox and Gaussian Curvature predictions have lower R^2 values, however, they are still sufficient for use with performance-guided DDPM sampling [44].

Performance Objective	Training Fit: $[R^2]$
Wave Drag C_w	0.973
Surface Area$_{50\%}$	0.983
Surface Area$_{100\%}$	0.982
Volume$_{50\%}$	0.988
Volume$_{100\%}$	0.986
Volume$_{\text{MaxBox}}$	0.784
Gaussian Curvature	0.765

Table 5: The performance prediction neural networks have high goodness-of-fits to the training data, which enables performance guidance in DDPM sampling with these objectives.

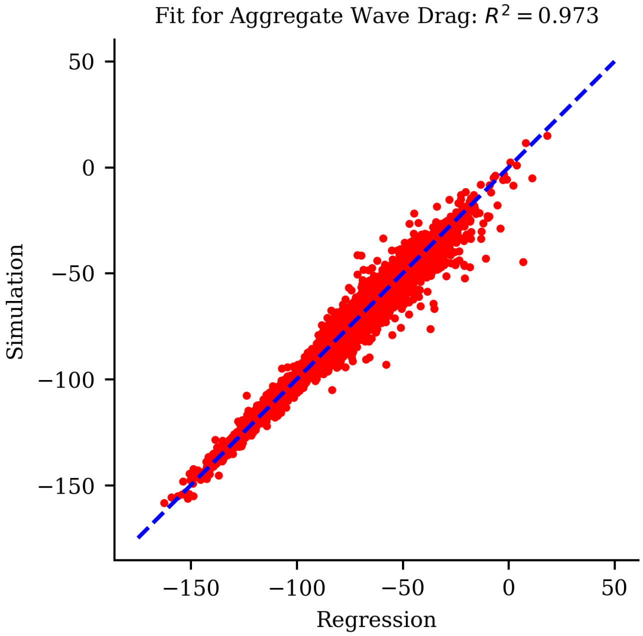


Figure 12: Comparison of the neural network prediction to the simulation value (ground truth) across the dataset for aggregate wave drag. This regression had a R^2 equal to 0.973. A perfect prediction ($R^2 = 1$) is shown by the blue dashed line.

4.3.2 Multi-objective Guided Performance Hull Generation

The seven performance prediction neural networks were implemented with the guided DDPM to generate 1000 hulls. Each objective in these samples was randomly weighted so that the influence of each of the performance metrics varied among the samples. The feasibility classifier guidance was tuned to $\gamma = 0.5$ to maintain some variability and dataset coverage among the samples and to not overpower the performance guidance. The samples generated with performance guidance were feasible 83.9% of the time. The PCA plot of these generated samples is shown in Figure 14. These samples do not cover the sample range of the design space as the Ship-D dataset hulls.

After sampling, the 839 feasible hull designs were simulated and measured with the seven performance objectives. The mean and standard deviation of the performance metrics among the Ship-D dataset and the generated samples are provided in Table 6. These metrics are scaled according to Equations 5- 11, so it is important to note that these values exist on a logarithmic scale. Among these samples, the wave drag coefficients and displaced volumes showed significant improvements in their performance. These improvements were at the expense of a relative increase in the surface area and Gaussian curvature. The generated samples have wave drag coefficients for any single speed/draft condition that is, on average, 91.4% lower than the average wave drag coefficients of the Ship-D dataset hulls. For the displaced volumes, these generated hulls have an average 114x increase in displaced volume in the bottom 50% of the hull depth and an average 47.9x increase in the total displaced volume of the hull. The generated hulls have, on average, 2.1x more total surface area, 4.4x more surface area in the bottom 50% of the hull, and 1.51x more double curvature compared to the Ship-D hulls. This is not desirable. The MaxBox metric saw a small, but negligible decrease in the volume ratio of the hull belonging to the MaxBox, where the generated samples have an average 5.2% reduction in the MaxBox volume ratio compared to the hulls in the dataset. This result, however, is far overshadowed by the substantial increase in total available volume in the hull.

In addition to measuring the performance of these hulls, a .stl mesh and 5 images of each hull were created for visual analysis. Figure 15 shows nine of these hulls. A major difference in these generated hulls is their higher length-to-beam ratios compared to the Ship-D hulls seen in Figure 2.

5 Discussion

The following subsections provide insight into the results of the studies presented. The first subsection reviews the successful generation of feasible designs with the different DDPMs. The second subsection provides an analysis of the dataset coverage of the DDPMs, with special attention made to the γ hyperparameter used in the classifier-guided DDPM. The third subsection analyzes the performance of the hulls generated with performance guidance. The Appendix contains a discussion on the conditional DDPM study.

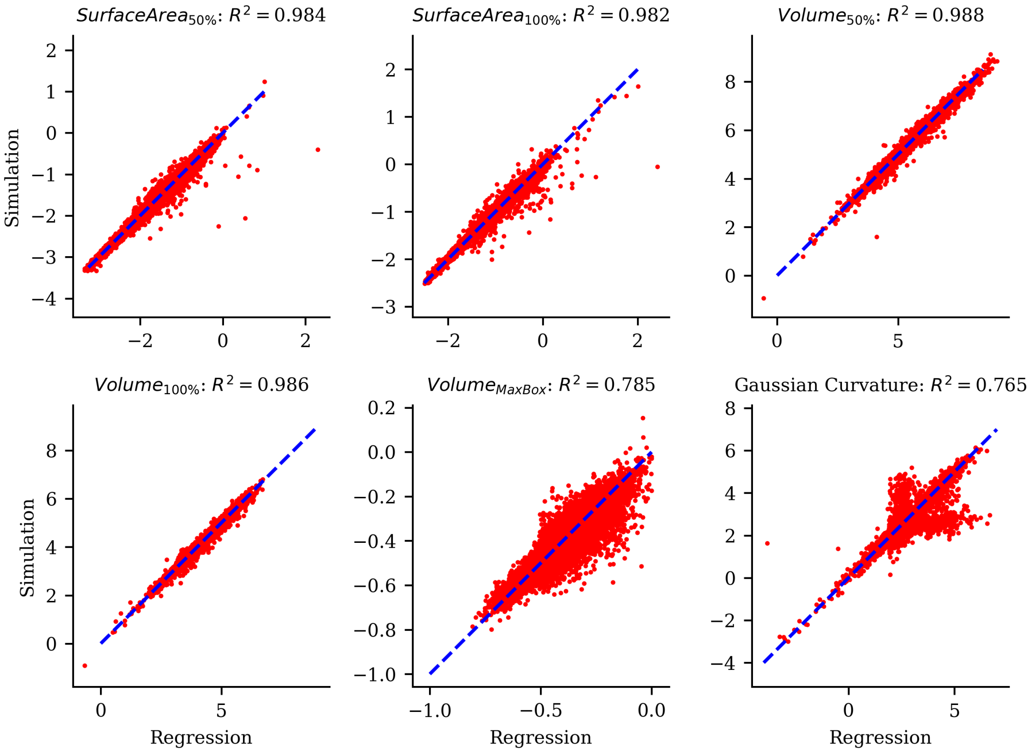


Figure 13: Comparison of the neural network prediction to the simulation value (ground truth) across the dataset for the remaining six performance metrics. All of these performance metric regressions are well resolved. A perfect prediction ($R^2 = 1$) is shown by the blue dashed line.

Performance Objective	**Ship-D Dataset** (Y_{*gen}) Mean	Std.	**Generated Samples** (Y_{*DS}) Mean	Std.	**Scaled Factor** (Y_{gen}/Y_{DS})
Wave Drag C_w	-73.40	17.38	-107.45	23.90	**0.086**
Surface Area$_{50\%}$	-1.71	0.53	-1.07	0.19	4.365
Surface Area$_{100\%}$	-1.09	0.45	-0.76	0.19	2.138
Volume$_{50\%}$	4.78	0.81	2.72	0.59	**114.815**
Volume$_{100\%}$	3.80	0.62	2.12	0.43	**47.863**
Volume$_{\text{MaxBox}}$	-0.407	0.010	-0.384	0.072	0.948
Gaussian Curvature	2.43	0.529	2.61	0.24	1.514

Table 6: The table shows the mean and standard deviation of the performance metrics across the Ship-D dataset hulls and the feasible generated hulls. The generated hulls saw a 47.9x increase in total volume and a 91.4% relative decrease in wave drag coefficient across all speeds.

5.1 Feasibility Constraint Satisfaction

Of the different DDPMs, only the classifier-guided DDPM showed to successfully sample feasible parameterized hull designs with the same success rate as the interpolation study. The standard DDPM, while only producing a feasible hull approximately half of the time, was able to do so only by implicitly learning the statistical relationships between design parameters in feasible hulls. The standard DDPM's feasibility success rate of 51.1% success rate is significantly higher than the success rate of 0.66% seen by randomly sampling the design space. In the comparison between CTGAN and DDPM, a significant performance gap was identified, particularly in constraint modeling. The CTGAN benchmark study revealed that the standard DDPM is two orders of magnitude more successful at generating feasible hulls than the CTGAN model. While the dataset coverage study demonstrated CTGAN's ability to produce parameterized vectors representing the dataset statistics, leading to high coverage, it struggled to generate these design vectors with combinations of parameter values that result in high feasibility. This suggests that CTGAN may face challenges in implicitly learning the statistical correlations between the parameters to the extent that DDPM does, underlying the

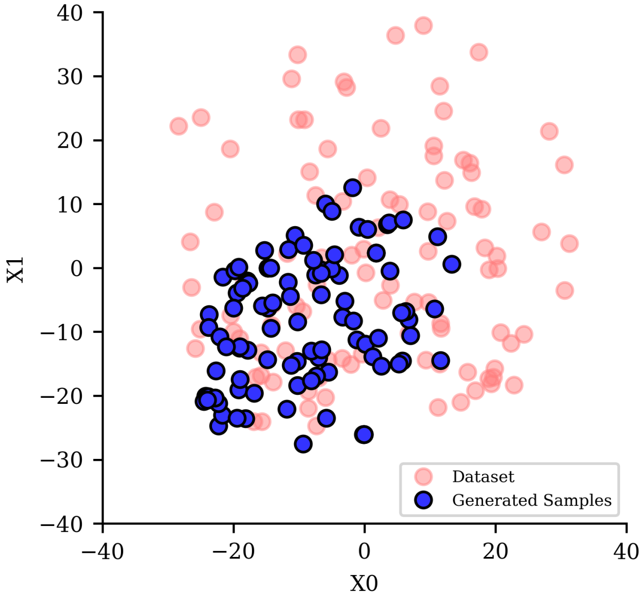


Figure 14: Two-dimensional Principal Component Analysis of the hull parameterization shows that the performance-guided DDPM with $\gamma = 0.5$ leads to sample coverage that is skewed relative to the distribution of the Ship-D dataset as a result of the performance guidance.

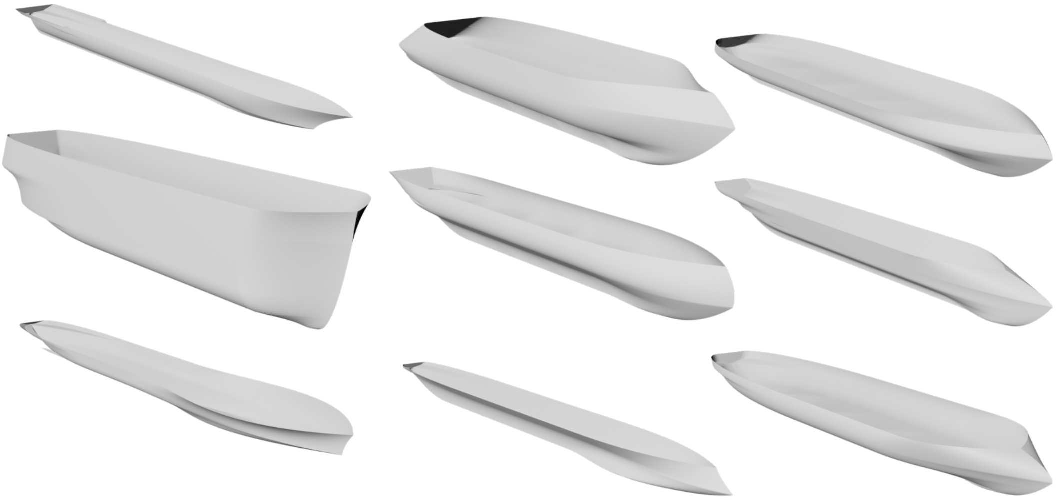


Figure 15: A selection of hulls generated with multi-objective guided performance generation. Notice the relative slenderness of the hulls leading to drastically reduced drag coefficients relative to the dataset hulls.

need for a comprehensive examination. Such an examination, backed by empirical and theoretical analysis, is essential to delve deeper into the observed challenges and understand the inherent model characteristics or learning behaviors causing the performance disparity. Without this thorough analysis, making definitive claims regarding the observed differences remains speculative. A deeper comparative study on constraint satisfaction across different deep generative models is needed to make such claims.

Finally, the guided classifier guidance showed that by tuning the γ hyperparameter, the rate of feasible hull generation varied. In order to meet the feasibility benchmark of 93% feasible hulls, the γ should be set between 0.35 and 1. Table 4 also shows that simply by including a small influence of guidance ($\gamma = 0.2$), the success of generating feasible hulls improves significantly compared to the standard DDPM. For performance guidance, the success rate of generating feasible hulls was 83.9%, which is lower than the benchmark target, but this comes at the benefit of producing high-performing hulls, even with $\gamma = 0.5$. This reduction in the feasibility satisfaction rate is due to the added influence of the performance guidance, which does not consider feasibility when generating samples. Further work in hyperparameter tuning can lead to higher success rates in feasible and high-performing hull generation. Overall, a DDPM with classifier guidance can be used to generate feasible design vectors with a high degree of success in an extremely complicated design space.

5.2 Dataset Coverage

Among the DDPMs, there were varying degrees of dataset coverage. Visually comparing the design space coverage of the Ship-D dataset hulls and the generated samples proved to be reasonably effective at analyzing the dataset coverage. Among the two interpolation methods, interpolation between a random design vector and its nearest neighbor was the best benchmark for dataset coverage as the diversity in the generated samples relied on the diversity of the randomly selected design vectors. The standard DDPM was also effective at maintaining dataset coverage as it was trained to generate sample hulls that are representative of the hulls in the dataset, as seen in Figure 7. The standard DDPM also maintains dataset coverage better than CTGAN. Finally, the DDPM with classifier guidance heavily relied on γ for maintaining dataset coverage. As γ is increased to produce feasible hulls with a higher success rate, the dataset coverage of the feasible samples decreases. Figure 9 was made to quantify the coverage and realism of the samples in addition to visually inspecting the PCA distribution of the generated samples. The best balance of maintaining a high feasibility success rate and dataset coverage was at $\gamma \approx 0.35 - 0.5$, as shown in Table 3. With increasing γ the generated samples lose diversity and cluster around the center of the PCA distribution. While the performance-guided DDPM was not intended to generate designs that cover the dataset, Figure 14 suggests that these generated samples do maintain some diversity and dataset coverage. Overall, the classifier-guided DDPM is shown the maintain a large breadth of dataset coverage with careful tuning of its hyperparameters.

5.3 Performance Guidance

The DDPM with performance guidance produced hulls with mixed results. The performance guidance created hulls with an average 91.4% lower drag and 47.9x higher displaced volumes than the hulls from the original dataset. This is a highly desirable outcome of the performance guided sample generation. This outcome, however, came at the expense of the generated hulls having increased surface area and double curvature, which is not ultimately desirable. Further in-depth analysis, such as life cycle costs assessments, is needed to weigh the impact these results would have on a real, scaled-up ship instead of a non-dimensionalized parametric hull shape. Future work can also consider different tuning of the γ and λ hyperparameters in the performance guidance of the hulls. The random λ weights used for this study were likely not scaled appropriately for the magnitude of the gradients of the different performance metrics. This could explain why the aggregate wave drag coefficient and displaced volume metrics were improved drastically; while MaxBox was relatively unaffected, and surface area and Gaussian curvature were increased. As these performance gradients were calculated using the weights of the regression neural networks, it is possible that the magnitudes of gradients between the different regression models disproportionately affected the net influence of guidance on design generation. This is rather apparent when comparing the significant, yet desired, increase in hull volume and the undesired increase in surface area. These two competing performance objectives should have maintained some balance of improvement across the DDPM generated samples; however, the displaced volume performance objectives saw overwhelming improvement that included a relative detriment to the surface area objectives. Nonetheless, the significant improvements in wave drag coefficient and displaced volume have strong economic prospects on the cost of operating a ship: the cost of fuel (drag) and the ability to generate revenue (carry cargo). Leveraging a DDPM with performance guidance has been shown to generate hull shapes considering multiple objectives that can lead to huge cost savings to ship operators. Future work will explore generating hulls with specific performance requirements to find explicit applicability of guided DDPMs to generate hulls tied to real cost savings in ship design.

In addition to the performance of the generated hulls, these generated samples share more semblance of real ship hulls than do the Ship-D hulls. Figure 15 shows a sample of the generated hulls for visual inspection. These hulls have higher length-to-beam ratios than the Ship-D hulls and have streamlines that are more akin to real ship hulls.

6 Conclusion

The goal of this work was to generate ship hulls using a denoising diffusion probabilistic model that considers the performance of the hull as part of the design generation. First, by training a DDPM on a dataset of randomly generated feasible hull designs, the DDPM was able to implicitly learn statistical relationships between the design parameters to generate feasible parametric hulls. Then, by incorporating guidance from performance prediction models trained on the same dataset of hulls, the DDPM was able to generate high-performing hulls with only information learned from the low-performing hulls in the dataset.

One critical aspect of leveraging generative AI on parametric design information is to generate feasible designs. A standard diffusion model can generate feasible hulls approximately 51.1% of the time. While this is much better than the success rate seen by randomly generating hull parameterizations ($\sim$ 0.66%), standard diffusion models do not yield feasible hulls at rates similar to interpolation methods ($\sim$ 93%). By leveraging the gradients of a classifier during the sampling process, the standard DDPM saw increased feasibility among generated hull designs. The classifier guidance in the denoising process was influenced by a tunable parameter, γ. By varying γ, the guided DDPM was able to generate hulls with different success rates of feasibility at the expense of design coverage across the dataset. It was found that γ = 0.5 led to high hull feasibility (99.5%) with limited detriment to dataset coverage.

As guidance was shown to be the most successful and versatile method of producing feasible hull designs, guidance was also used to generate high-performing designs. Seven neural networks were trained to predict the different performance metrics given a hull's design vector. The gradients of these performance prediction neural networks were implemented for performance guidance in the DDPM's sampling process. The aggregate wave drag coefficients of the generated hulls had a 91.4% mean reduction in drag coefficient compared to the Ship-D hulls. The total displaced volume of the generated samples was on average 47.9x larger than the mean displaced volume of the Ship-D dataset hulls. However, surface area, Gaussian curvature, and MaxBox of the generated samples did not improve compared to the hulls in the dataset. Overall, the significantly reduced drag coefficients and increased displaced volume are extremely beneficial to ship design. These performance metrics dictate how expensive a voyage is (fuel costs due to drag) and how much cargo the ship can carry (how much money can be made on a voyage). With this work, the economic prospect of leveraging generative AI to design ship hulls is shown.

6.1 Future Work

Future work will focus on the continued study of generative AI to generate ship hulls and other systems on a ship. Immediate future work will look at continued tuning of the λ weights during performance guidance to generate hulls that have improved performance in all of the objectives. To accomplish this, a study on hyperparameter tuning and guidance gradients will be performed. Further, work in leveraging guided diffusion to generate high-performing ship hulls with specific performance targets will be explored. The goal of this future work is to generate hulls that consider specific user-defined constraints (such as dimensions, volume, speed, etc.) with high performance. This way, the design of ship hulls using DDPMs could be analogous to similarly structured online text-to-image DDPMs, such as *Dall-E* [38] and *Stable Diffusion* [39]. In addition to generating ship hulls, further work in DDPMs to generate other aspects of ship design will be explored, such as structural design generation, packing arrangements, machinery, and outfitting on a ship.

7 Acknowledgements

This research is funded by the United States' Department of Defense, Office of Naval Research, via the National Defense Science and Engineering Graduate (NDSEG) Fellowship program. The authors would like to thank MIT Supercloud for providing some of the computational resources needed to perform this work [52]. The data and code used for the ShipGen project are available at `https://decode.mit.edu/projects/shipgen/`

A APPENDIX: Conditional Diffusion Model

The appendix contains an additional study performed by training a conditional DDPM. Conditional DDPMs are similar to the standard DDPM, however, their structure includes extra layers that embed extra information in the training and sampling process. The extra information in this study is a sample's classification of being feasible or invalid. The following subsections detail the Methods, Results, and a Discussion on leveraging a conditional DDPM to generate hull designs.

A.1 Methods

A tabular DDPM was built with additional conditioning embedding layers to influence the model to produce designs that satisfy the feasibility constraints. In addition to the 30,000 parametric hulls in the Ship-D dataset, 20,000 design vectors were randomly generated that do not meet at least one of the feasibility constraints. The feasible ship hull design vectors and the infeasible vectors were labeled respectively. While training the tabular DDPM, the feasibility label was provided to the conditional embedding layer with its respective design vector in training. The goal of the conditioning is to use that additional label to influence the sampling process to guide the tabular DDPM to produce designs that satisfy the feasibility constraints. To modify the standard DDPM to become a conditional DDPM modify the gradient step (Step 5) in the DDPM training algorithm with Equation 16, where C is the conditional embedding layer that is concatenated to the first layer of the standard DDPM. In sampling, replacing Step 4 of the sampling algorithm with Equation 17, where C is the same conditional embedding layer concatenated to the first layer of the DDPM. The DDPM training and sampling algorithms are provided in Table 1 and Table 2, respectively. The Results Section provides the sample distribution and constraint satisfaction of ship hull design vectors generated with the conditioned tabular DDPM.

$$\nabla_\theta ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t, C)||^2 \tag{16}$$

$$X_{t-1} = \frac{1}{\sqrt{\alpha_t}}(X_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(X_t, t, C)) + \sigma_t Z \tag{17}$$

A.2 Results

The conditional DDPM was trained on the thirty thousand Ship-D hulls that satisfy all forty nine constraints and twenty thousand invalid samples that violate varying numbers of the forty nine constraints. Two separate sample generations were performed with the conditional DDPM. The first study tried to intentionally generate feasible hulls. Figure 16 shows that the conditional DDPM produces feasible hulls 39.8% of the time. The PCA plot in Figure 17 shows that the spread of the generated samples is within the bounds of the dataset. This conditional DDPM can also intentionally generate invalid samples. Although this is not useful in design work, the generation of invalid samples with the conditional DDPM shows that the model can distinguish between "positive" and "invalid" samples in sample generation. Figures 18 and 19 showcase the results of hulls that were intentionally created to violate the hull parameterization's feasibility criteria.

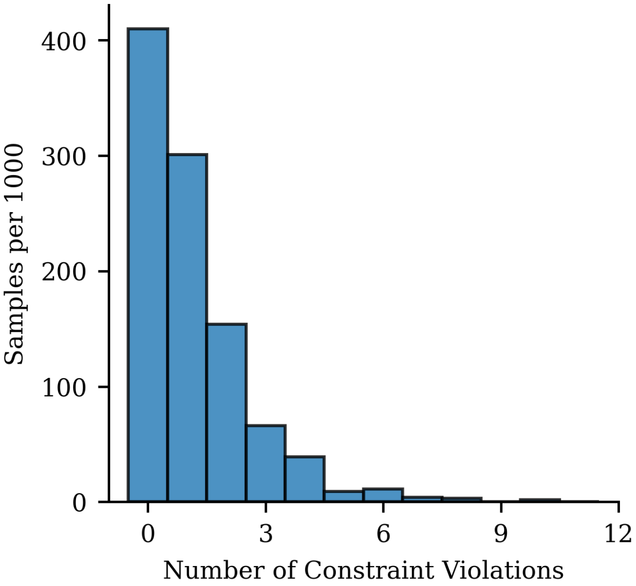


Figure 16: Bar graph showing the number of individuals generated with an increasing number of constraint violations. Leveraging a conditional DDPM to generate hull parameterization leads to hull parameterizations that satisfy all constraints 39.8% of the time

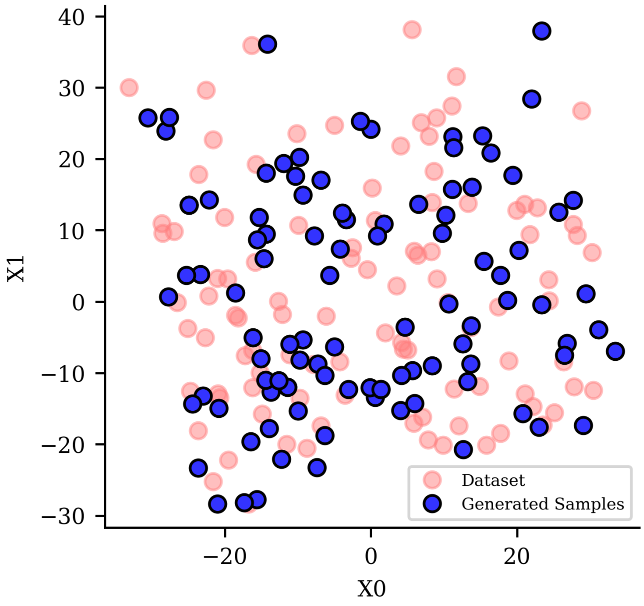


Figure 17: Two-dimensional Principal Component Analysis of the hull parameterization shows that hulls generated with a conditional DDPM maintain most of the dataset coverage; however, there is less design feasibility among the samples than desired.

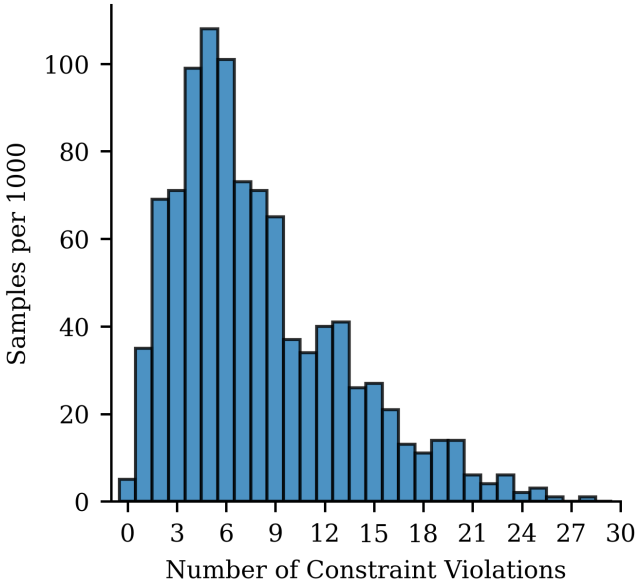


Figure 18: Bar graph showing the number of individuals generated with an increasing number of constraint violations. Leveraging a conditional DDPM to generate invalid hull parameterization leads to the generation of samples that violate a large spread of a number of design feasibility constraints.

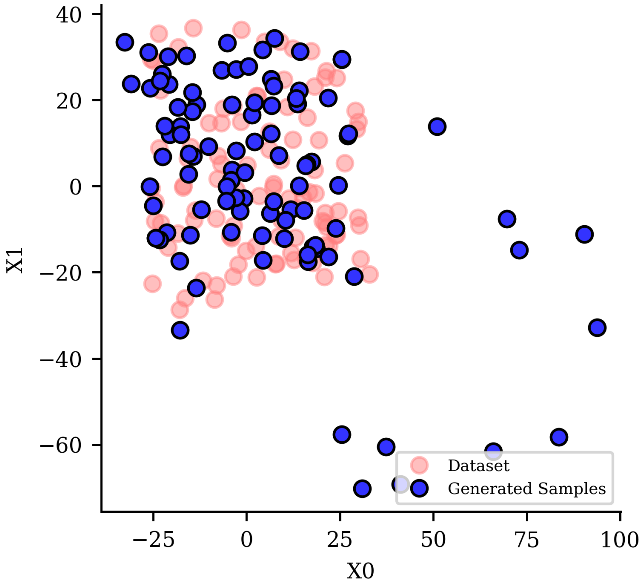


Figure 19: Two-dimensional Principal Component Analysis of the hull parameterization shows that invalid hull samples cover the dataset space and more, as some of the parameters are intentionally sampled outside of the feasible range of some parameters.

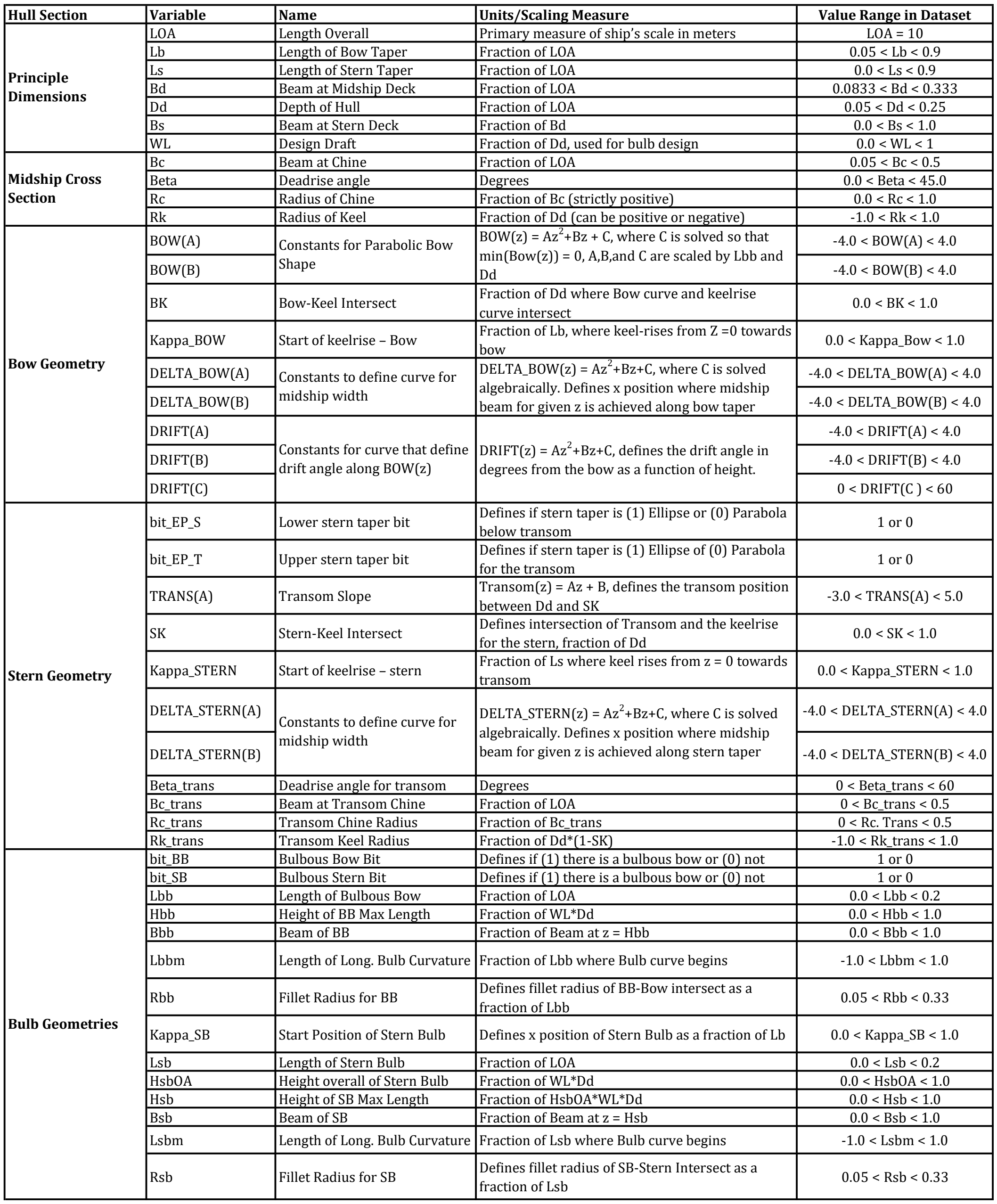
A.3 Discussion

The conditional DDPM was not able to produce feasible hulls at a rate that meets the interpolation study benchmarks. It was surprising to see how poorly the conditional DDPM performed given that the model was trained with distinction between feasible and infeasible design vectors. The conditional model, however, could intentionally create infeasible hulls at will, which further adds to the surprise of the poor feasible hull generation. It seems that the conditional DDPM struggled to distinguish the statistical relationships between the parameters in the feasible and infeasible hulls during training. This result did not yield improvement over the standard DDPM in generating feasible hulls, but it was excellent in generating a large diversity of hulls with at least one constraint violation. This could be useful in future studies concerning design feasibility.

B APPENDIX: Parametric Hull Design: Parameters and Constraints

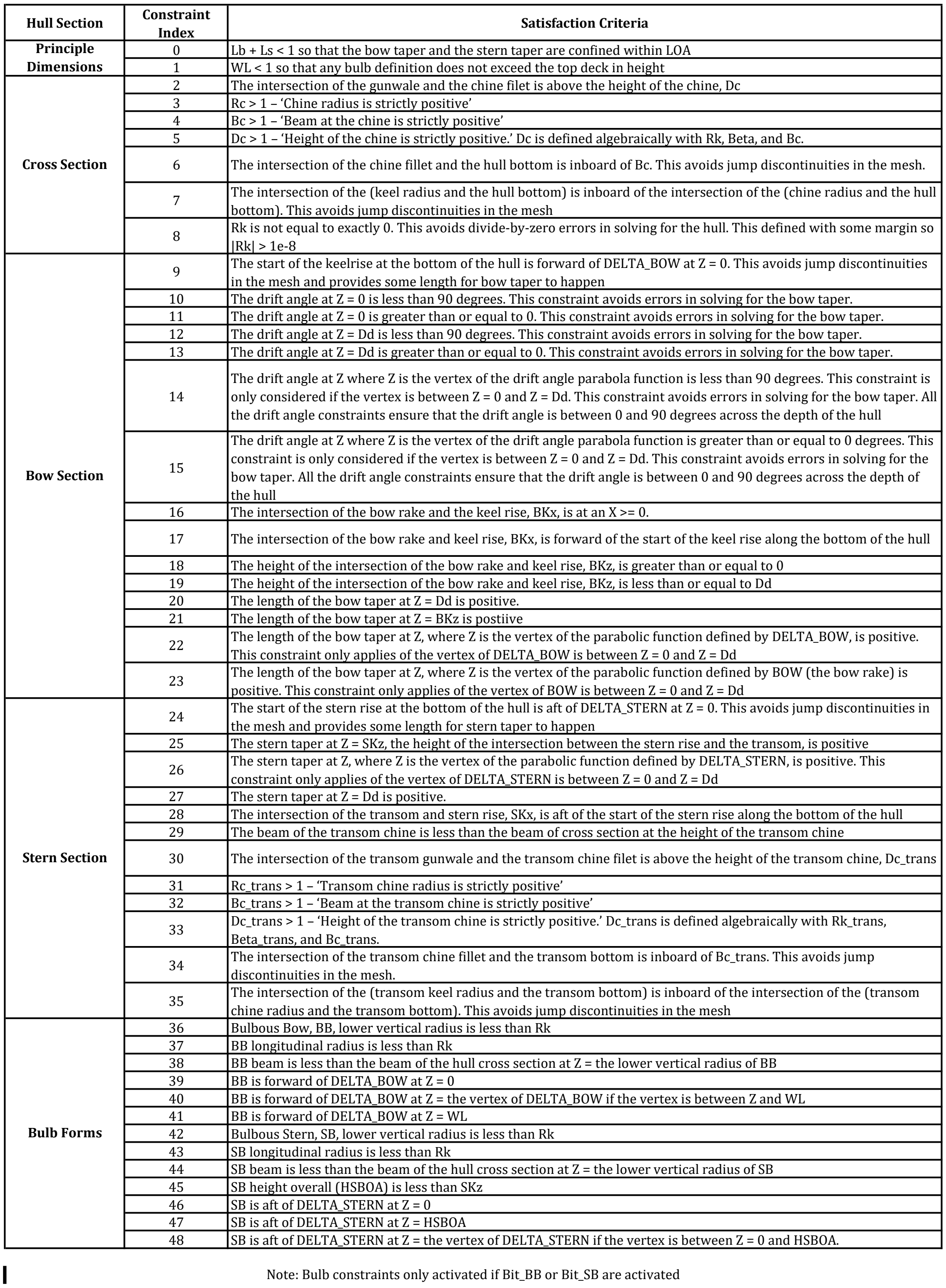
This appendix provides documentation for the 45 hull design parameters and the 49 algebraic feasibility constraints. Figure 20 lists the design parameters, describes the features, provides the ranges for each parameter in the Ship-D Dataset. Figure 21 lists each of the 49 algebraic constraints, and describes the conditions to satisfy each constraint. By satisfying all 49 algebraic constraints, the hull will satisfy the two feasibility criteria:

1. The hull is watertight, meaning that there are no holes on its surface.
2. The hull surface is not self-intersecting.



Hull Section	Variable	Name	Units/Scaling Measure	Value Range in Dataset
Principle Dimensions	LOA	Length Overall	Primary measure of ship's scale in meters	LOA = 10
	Lb	Length of Bow Taper	Fraction of LOA	0.05 < Lb < 0.9
	Ls	Length of Stern Taper	Fraction of LOA	0.0 < Ls < 0.9
	Bd	Beam at Midship Deck	Fraction of LOA	0.0833 < Bd < 0.333
	Dd	Depth of Hull	Fraction of LOA	0.05 < Dd < 0.25
	Bs	Beam at Stern Deck	Fraction of Bd	0.0 < Bs < 1.0
	WL	Design Draft	Fraction of Dd, used for bulb design	0.0 < WL < 1
Midship Cross Section	Bc	Beam at Chine	Fraction of LOA	0.05 < Bc < 0.5
	Beta	Deadrise angle	Degrees	0.0 < Beta < 45.0
	Rc	Radius of Chine	Fraction of Bc (strictly positive)	0.0 < Rc < 1.0
	Rk	Radius of Keel	Fraction of Dd (can be positive or negative)	-1.0 < Rk < 1.0
Bow Geometry	BOW(A)	Constants for Parabolic Bow Shape	BOW(z) = Az^2+Bz + C, where C is solved so that min(Bow(z)) = 0, A,B,and C are scaled by Lbb and Dd	-4.0 < BOW(A) < 4.0
	BOW(B)			-4.0 < BOW(B) < 4.0
	BK	Bow-Keel Intersect	Fraction of Dd where Bow curve and keelrise curve intersect	0.0 < BK < 1.0
	Kappa_BOW	Start of keelrise – Bow	Fraction of Lb, where keel-rises from Z =0 towards bow	0.0 < Kappa_Bow < 1.0
	DELTA_BOW(A)	Constants to define curve for midship width	DELTA_BOW(z) = Az^2+Bz+C, where C is solved algebraically. Defines x position where midship beam for given z is achieved along bow taper	-4.0 < DELTA_BOW(A) < 4.0
	DELTA_BOW(B)			-4.0 < DELTA_BOW(B) < 4.0
	DRIFT(A)	Constants for curve that define drift angle along BOW(z)	DRIFT(z) = Az^2+Bz+C, defines the drift angle in degrees from the bow as a function of height.	-4.0 < DRIFT(A) < 4.0
	DRIFT(B)			-4.0 < DRIFT(B) < 4.0
	DRIFT(C)			0 < DRIFT(C) < 60
Stern Geometry	bit_EP_S	Lower stern taper bit	Defines if stern taper is (1) Ellipse or (0) Parabola below transom	1 or 0
	bit_EP_T	Upper stern taper bit	Defines if stern taper is (1) Ellipse of (0) Parabola for the transom	1 or 0
	TRANS(A)	Transom Slope	Transom(z) = Az + B, defines the transom position between Dd and SK	-3.0 < TRANS(A) < 5.0
	SK	Stern-Keel Intersect	Defines intersection of Transom and the keelrise for the stern, fraction of Dd	0.0 < SK < 1.0
	Kappa_STERN	Start of keelrise – stern	Fraction of Ls where keel rises from z = 0 towards transom	0.0 < Kappa_STERN < 1.0
	DELTA_STERN(A)	Constants to define curve for midship width	DELTA_STERN(z) = Az^2+Bz+C, where C is solved algebraically. Defines x position where midship beam for given z is achieved along stern taper	-4.0 < DELTA_STERN(A) < 4.0
	DELTA_STERN(B)			-4.0 < DELTA_STERN(B) < 4.0
	Beta_trans	Deadrise angle for transom	Degrees	0 < Beta_trans < 60
	Bc_trans	Beam at Transom Chine	Fraction of LOA	0 < Bc_trans < 0.5
	Rc_trans	Transom Chine Radius	Fraction of Bc_trans	0 < Rc. Trans < 0.5
	Rk_trans	Transom Keel Radius	Fraction of Dd*(1-SK)	-1.0 < Rk_trans < 1.0
Bulb Geometries	bit_BB	Bulbous Bow Bit	Defines if (1) there is a bulbous bow or (0) not	1 or 0
	bit_SB	Bulbous Stern Bit	Defines if (1) there is a bulbous bow or (0) not	1 or 0
	Lbb	Length of Bulbous Bow	Fraction of LOA	0.0 < Lbb < 0.2
	Hbb	Height of BB Max Length	Fraction of WL*Dd	0.0 < Hbb < 1.0
	Bbb	Beam of BB	Fraction of Beam at z = Hbb	0.0 < Bbb < 1.0
	Lbbm	Length of Long. Bulb Curvature	Fraction of Lbb where Bulb curve begins	-1.0 < Lbbm < 1.0
	Rbb	Fillet Radius for BB	Defines fillet radius of BB-Bow intersect as a fraction of Lbb	0.05 < Rbb < 0.33
	Kappa_SB	Start Position of Stern Bulb	Defines x position of Stern Bulb as a fraction of Lb	0.0 < Kappa_SB < 1.0
	Lsb	Length of Stern Bulb	Fraction of LOA	0.0 < Lsb < 0.2
	HsbOA	Height overall of Stern Bulb	Fraction of WL*Dd	0.0 < HsbOA < 1.0
	Hsb	Height of SB Max Length	Fraction of HsbOA*WL*Dd	0.0 < Hsb < 1.0
	Bsb	Beam of SB	Fraction of Beam at z = Hsb	0.0 < Bsb < 1.0
	Lsbm	Length of Long. Bulb Curvature	Fraction of Lsb where Bulb curve begins	-1.0 < Lsbm < 1.0
	Rsb	Fillet Radius for SB	Defines fillet radius of SB-Stern Intersect as a fraction of Lsb	0.05 < Rsb < 0.33

Figure 20: List of the hull design parameters, their scaling, and their value ranges within the dataset.



Hull Section	Constraint Index	Satisfaction Criteria
Principle Dimensions	0	Lb + Ls < 1 so that the bow taper and the stern taper are confined within LOA
	1	WL < 1 so that any bulb definition does not exceed the top deck in height
Cross Section	2	The intersection of the gunwale and the chine filet is above the height of the chine, Dc
	3	Rc > 1 – 'Chine radius is strictly positive'
	4	Bc > 1 – 'Beam at the chine is strictly positive'
	5	Dc > 1 – 'Height of the chine is strictly positive.' Dc is defined algebraically with Rk, Beta, and Bc.
	6	The intersection of the chine fillet and the hull bottom is inboard of Bc. This avoids jump discontinuities in the mesh.
	7	The intersection of the (keel radius and the hull bottom) is inboard of the intersection of the (chine radius and the hull bottom). This avoids jump discontinuities in the mesh
	8	Rk is not equal to exactly 0. This avoids divide-by-zero errors in solving for the hull. This defined with some margin so |Rk| > 1e-8
Bow Section	9	The start of the keelrise at the bottom of the hull is forward of DELTA_BOW at Z = 0. This avoids jump discontinuities in the mesh and provides some length for bow taper to happen
	10	The drift angle at Z = 0 is less than 90 degrees. This constraint avoids errors in solving for the bow taper.
	11	The drift angle at Z = 0 is greater than or equal to 0. This constraint avoids errors in solving for the bow taper.
	12	The drift angle at Z = Dd is less than 90 degrees. This constraint avoids errors in solving for the bow taper.
	13	The drift angle at Z = Dd is greater than or equal to 0. This constraint avoids errors in solving for the bow taper.
	14	The drift angle at Z where Z is the vertex of the drift angle parabola function is less than 90 degrees. This constraint is only considered if the vertex is between Z = 0 and Z = Dd. This constraint avoids errors in solving for the bow taper. All the drift angle constraints ensure that the drift angle is between 0 and 90 degrees across the depth of the hull
	15	The drift angle at Z where Z is the vertex of the drift angle parabola function is greater than or equal to 0 degrees. This constraint is only considered if the vertex is between Z = 0 and Z = Dd. This constraint avoids errors in solving for the bow taper. All the drift angle constraints ensure that the drift angle is between 0 and 90 degrees across the depth of the hull
	16	The intersection of the bow rake and the keel rise, BKx, is at an X >= 0.
	17	The intersection of the bow rake and keel rise, BKx, is forward of the start of the keel rise along the bottom of the hull
	18	The height of the intersection of the bow rake and keel rise, BKz, is greater than or equal to 0
	19	The height of the intersection of the bow rake and keel rise, BKz, is less than or equal to Dd
	20	The length of the bow taper at Z = Dd is positive.
	21	The length of the bow taper at Z = BKz is postiive
	22	The length of the bow taper at Z, where Z is the vertex of the parabolic function defined by DELTA_BOW, is positive. This constraint only applies of the vertex of DELTA_BOW is between Z = 0 and Z = Dd
	23	The length of the bow taper at Z, where Z is the vertex of the parabolic function defined by BOW (the bow rake) is positive. This constraint only applies of the vertex of BOW is between Z = 0 and Z = Dd
Stern Section	24	The start of the stern rise at the bottom of the hull is aft of DELTA_STERN at Z = 0. This avoids jump discontinuities in the mesh and provides some length for stern taper to happen
	25	The stern taper at Z = SKz, the height of the intersection between the stern rise and the transom, is positive
	26	The stern taper at Z, where Z is the vertex of the parabolic function defined by DELTA_STERN, is positive. This constraint only applies of the vertex of DELTA_STERN is between Z = 0 and Z = Dd
	27	The stern taper at Z = Dd is positive.
	28	The intersection of the transom and stern rise, SKx, is aft of the start of the stern rise along the bottom of the hull
	29	The beam of the transom chine is less than the beam of cross section at the height of the transom chine
	30	The intersection of the transom gunwale and the transom chine filet is above the height of the transom chine, Dc_trans
	31	Rc_trans > 1 – 'Transom chine radius is strictly positive'
	32	Bc_trans > 1 – 'Beam at the transom chine is strictly positive'
	33	Dc_trans > 1 – 'Height of the transom chine is strictly positive.' Dc_trans is defined algebraically with Rk_trans, Beta_trans, and Bc_trans.
	34	The intersection of the transom chine fillet and the transom bottom is inboard of Bc_trans. This avoids jump discontinuities in the mesh.
	35	The intersection of the (transom keel radius and the transom bottom) is inboard of the intersection of the (transom chine radius and the transom bottom). This avoids jump discontinuities in the mesh
Bulb Forms	36	Bulbous Bow, BB, lower vertical radius is less than Rk
	37	BB longitudinal radius is less than Rk
	38	BB beam is less than the beam of the hull cross section at Z = the lower vertical radius of BB
	39	BB is forward of DELTA_BOW at Z = 0
	40	BB is forward of DELTA_BOW at Z = the vertex of DELTA_BOW if the vertex is between Z and WL
	41	BB is forward of DELTA_BOW at Z = WL
	42	Bulbous Stern, SB, lower vertical radius is less than Rk
	43	SB longitudinal radius is less than Rk
	44	SB beam is less than the beam of the hull cross section at Z = the lower vertical radius of SB
	45	SB height overall (HSBOA) is less than SKz
	46	SB is aft of DELTA_STERN at Z = 0
	47	SB is aft of DELTA_STERN at Z = HSBOA
	48	SB is aft of DELTA_STERN at Z = the vertex of DELTA_STERN if the vertex is between Z = 0 and HSBOA.

Note: Bulb constraints only activated if Bit_BB or Bit_SB are activated

Figure 21: List of the parametric hull design constraints and a description of their satisfaction criteria

References

1. Bagazinski, N. J. & Ahmed, F. *Ship-D: Ship Hull Dataset for Design Optimization using Machine Learning* in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2023).
2. Lin, C.-K. & Shaw, H.-J. Feature-based estimation of preliminary costs in shipbuilding. *Ocean Engineering* **144,** 305–319 (2017).
3. Evans, J. H. Basic design concepts. *Journal of the American Society for Naval Engineers* **71,** 671–678 (1959).
4. Brown, A. & Salcedo, J. Multiple-objective optimization in naval ship design. *Naval Engineers Journal* **115,** 49–62 (2003).
5. Feng, Y., el Moctar, O. & Schellin, T. Parametric Hull Form Optimization of Containerships for Minimum Resistance in Calm Water and in Waves. *Journal of Marine Science and Applications* (Jan. 2022).
6. Read, D. *A drag estimate for concept-stage ship design optimization* (The University of Maine, 2009).
7. Zhang, Y., Kim, D.-J. & Bahatmaka, A. *Parametric Method Using Grasshopper for Bulbous Bow Generation* in *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)* (2018), 307–310.
8. Chrismianto, D. & Kim, D.-J. Parametric bulbous bow design using the cubic Bezier curve and curve-plane intersection method for the minimization of ship resistance in CFD. *Journal of Marine Science and Technology* **19,** 479–492 (2014).
9. Lu, Y., Chang, X. & Hu, A.-k. A hydrodynamic optimization design methodology for a ship bulbous bow under multiple operating conditions. *Engineering Applications of Computational Fluid Mechanics* **10,** 330–345 (2016).
10. Knight, J. T., Zahradka, F. T., Singer, D. J. & Collette, M. D. Multiobjective Particle Swarm Optimization of a Planing Craft with Uncertainty. *Journal of Ship Production and Design* **30,** 194–200. ISSN: 2158-2866. eprint: `https://onepetro.org/JSPD/article-pdf/30/04/194/2204981/sname-jspd-2014-30-4-194.pdf`. `https://doi.org/10.5957/jspd.2014.30.4.194` (Nov. 2014).
11. Knight, J. T., Singer, D. J. & Collette, M. D. Testing of a spreading mechanism to promote diversity in multi-objective particle swarm optimization. *Optimization and Engineering* **16,** 279–302 (June 2015).
12. Hodges, J., Wheeler, M., Belhocine, M. & Henry, J. AI/ML APPLICATIONS FOR SHIP DESIGN. *ICCAS 2022* (Sept. 2022).
13. Wang, Y. *et al.* Three-dimensional ship hull encoding and optimization via deep neural networks. *Journal of Mechanical Design* **144,** 101701 (2022).
14. Ao, Y., Li, Y., Gong, J. & Li, S. An artificial intelligence-aided design (AIAD) of ship hull structures. *Journal of Ocean Engineering and Science* (2021).
15. Ao, Y., Li, Y., Gong, J. & Li, S. Artificial Intelligence Design for Ship Structures: A Variant Multiple-Input Neural Network-Based Ship Resistance Prediction. *Journal of Mechanical Design* **144,** 091707 (2022).
16. Peri, D., Rossetti, M. & Campana, E. F. Design optimization of ship hulls via CFD techniques. *Journal of ship research* **45,** 140–149 (2001).
17. Demo, N., Tezzele, M., Mola, A. & Rozza, G. Hull shape design optimization with parameter space and model reductions, and self-learning mesh morphing. *Journal of Marine Science and Engineering* **9,** 185 (2021).
18. Abbas, A., Rafiee, A. & Haase, M. DeepMorpher: deep learning-based design space dimensionality reduction for shape optimisation. *Journal of Engineering Design* **34,** 254–270 (2023).
19. Khan, S., Kaklis, P., Serani, A., Diez, M. & Kostas, K. Shape-supervised dimension reduction: Extracting geometry and physics associated features with geometric moments. *Computer-Aided Design* **150,** 103327 (2022).
20. Khan, S., Kaklis, P., Serani, A. & Diez, M. Geometric moment-dependent global sensitivity analysis without simulation data: application to ship hull form optimisation. *Computer-Aided Design* **151,** 103339 (2022).
21. Khan, S., Goucher-Lambert, K., Kostas, K. & Kaklis, P. ShipHullGAN: A generic parametric modeller for ship hull design using deep convolutional generative model. *Computer Methods in Applied Mechanics and Engineering* **411,** 116051 (2023).
22. Shaeffer, A. *Application of Artificial Neural Networks to Early-Stage Hull Form Design* PhD thesis (George Mason University, 2023).
23. Shaeffer, A. K., Wilson, W. & Yang, C. *Application of Machine Learning to Early-Stage Hull Form Design* in *SNAME Maritime Convention* (2020), D043S019R002.
24. Hollenbach, K. U. Estimating resistance and propulsion for single-screw and twin-screw ships-ship technology research 45 (1998). *Schiffstechnik* **45,** 72 (1998).

25. Hollenbach, U. & Friesch, J. *Efficient hull forms–What can be gained* in *Proceedings of the 1st International Conference on Ship Efficiency, Hamburg, Germany* (2007), 8–9.
26. Savitsky, D. Hydrodynamic Design of Planing Hulls. *Marine Technology and SNAME News* **1,** 71–95. ISSN: 0025-3316. eprint: `https://onepetro.org/MTSN/article-pdf/1/04/71/2198785/sname-mtsn-1964-1-4-71.pdf`. `https://doi.org/10.5957/mt1.1964.1.4.71` (Oct. 1964).
27. Michell, J. H. XI. The wave-resistance of a ship. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **45,** 106–123 (1898).
28. Tuck, E. O. The wave resistance formula of JH Michell (1898) and its significance to recent research in ship hydrodynamics. *The ANZIAM Journal* **30,** 365–377 (1989).
29. Mantzaris, D. A. *A Rankine panel method as a tool for the hydrodynamic design of complex marine vehicles* PhD thesis (Massachusetts Institute of Technology, 1998).
30. Dawson, C. *A practical computer method for solving ship-wave problems* in *Proceedings of Second International Conference on Numerical Ship Hydrodynamics* (1977), 30–38.
31. Noblesse, F., Huang, F. & Yang, C. The Neumann–Michell theory of ship waves. *Journal of Engineering Mathematics* **79,** 51–71 (2013).
32. Yang, C., Huang, F. & Noblesse, F. Practical evaluation of the drag of a ship for design and optimization. *Journal of Hydrodynamics* **25,** 645–654 (2013).
33. Huang, F., Yang, C. & Noblesse, F. Numerical implementation and validation of the Neumann–Michell theory of ship waves. *European Journal of Mechanics-B/Fluids* **42,** 47–68 (2013).
34. Marlantes, K. & Maki, K. *Modeling Vertical Planing Boat Motions using a Neural-Corrector Method* in *SNAME International Conference on Fast Sea Transportation* **Day 1 Tue, October 26, 2021** (Oct. 2021). eprint: `https://onepetro.org/snamefast/proceedings-pdf/FAST21/1-FAST21/D011S002R005/2520674/sname-fast-2021-014.pdf`. `https://doi.org/10.5957/FAST-2021-014`.
35. Silva, K. M. & Maki, K. J. Implementation of the Critical Wave Groups Method with Computational Fluid Dynamics and Neural Networks. *arXiv preprint arXiv:2301.09834* (2023).
36. Dhariwal, P. & Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* **34,** 8780–8794 (2021).
37. Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33,** 6840–6851 (2020).
38. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. & Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).
39. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. *High-Resolution Image Synthesis with Latent Diffusion Models* in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 10674–10685.
40. Liu, R. *et al. Zero-1-to-3: Zero-shot one image to 3d object* 2023.
41. Mazé, F. & Ahmed, F. Diffusion Models Beat GANs on Topology Optimization. *Proceedings of the AAAI Conference on Artificial Intelligence* **37,** 9108–9116. `https://ojs.aaai.org/index.php/AAAI/article/view/26093` (June 2023).
42. Giannone, G., Srivastava, A., Winther, O. & Ahmed, F. Aligning Optimization Trajectories with Diffusion Models for Constrained Design Generation. *arXiv preprint arXiv:2305.18470* (2023).
43. Giannone, G., Regenwetter, L., Srivastava, A., Gutfreund, D. & Ahmed, F. Learning from Invalid Data: On Constraint Satisfaction in Generative Models. *arXiv preprint arXiv:2306.15166* (2023).
44. Arechiga, N., Permenter, F., Song, B. & Yuan, C. Drag-guided diffusion models for vehicle image generation. *arXiv preprint arXiv:2306.09935* (2023).
45. Zubaly, R. *Applied Naval Architecture* ISBN: 9780870334757 (Cornell Maritime Press, 1996).
46. Newman, J. N. *Marine hydrodynamics* (The MIT press, 2018).
47. Dalle, D. Comparison of numerical techniques for Euclidean curvature. *Rose-Hulman Undergraduate Mathematics Journal* **7,** 12 (2006).
48. Nelder, J. A. & Mead, R. A simplex method for function minimization. *The computer journal* **7,** 308–313 (1965).
49. Regenwetter, L., Srivastava, A., Gutfreund, D. & Ahmed, F. Beyond Statistical Similarity: Rethinking Metrics for Deep Generative Models in Engineering Design. *Computer-Aided Design* **165,** 103609. ISSN: 0010-4485. `https://www.sciencedirect.com/science/article/pii/S0010448523001410` (2023).
50. Xu, L., Skoularidou, M., Cuesta-Infante, A. & Veeramachaneni, K. *Modeling Tabular data using Conditional GAN* in *Advances in Neural Information Processing Systems* (2019).
51. Kotelnikov, A., Baranchuk, D., Rubachev, I. & Babenko, A. *Tabddpm: Modelling tabular data with diffusion models* in *International Conference on Machine Learning* (2023), 17564–17579.

52. Reuther, A. *et al. Interactive Supercomputing on 40,000 Cores for Machine Learning and Data Analysis* in *2018 IEEE High Performance extreme Computing Conference (HPEC)* (2018), 1–6.