AI Co-Designer for Cyber-physical Systems

Susmit Jha

Neurosymbolic Computing and Intelligence (NuSCI)

Computer Science Laboratory, SRI International

Cyber-physical Systems

Distribution Statement A. Approved for Public Release, Distribution Unlimited

Cyber-physical Systems

Talk Outline

- Key Challenges in CPS Design and Desiderata of a neuro-symbolic AI Co-Designer

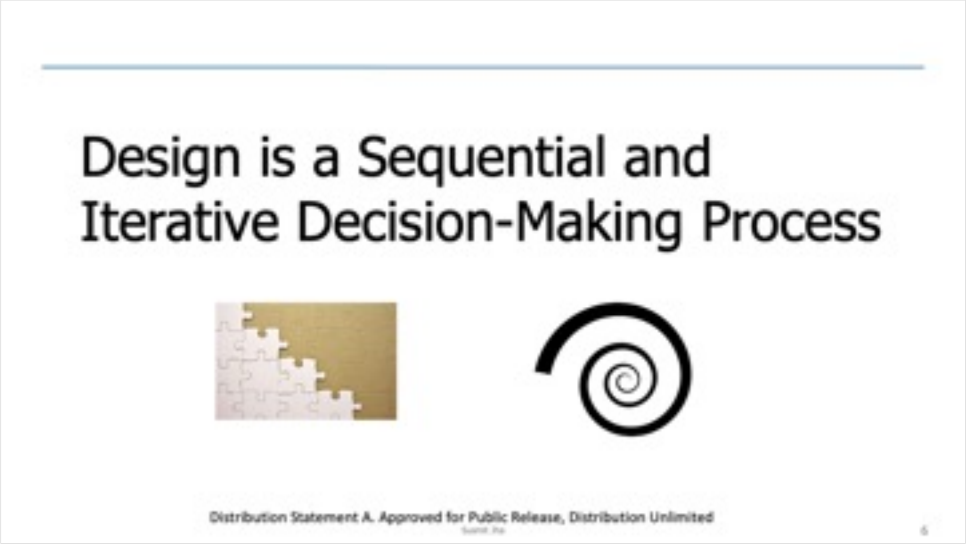
- Role of AI
  - Learning-based approaches
  - Neuro-symbolic Generator
  - Satisfiability-based Combinatorial Search Methods

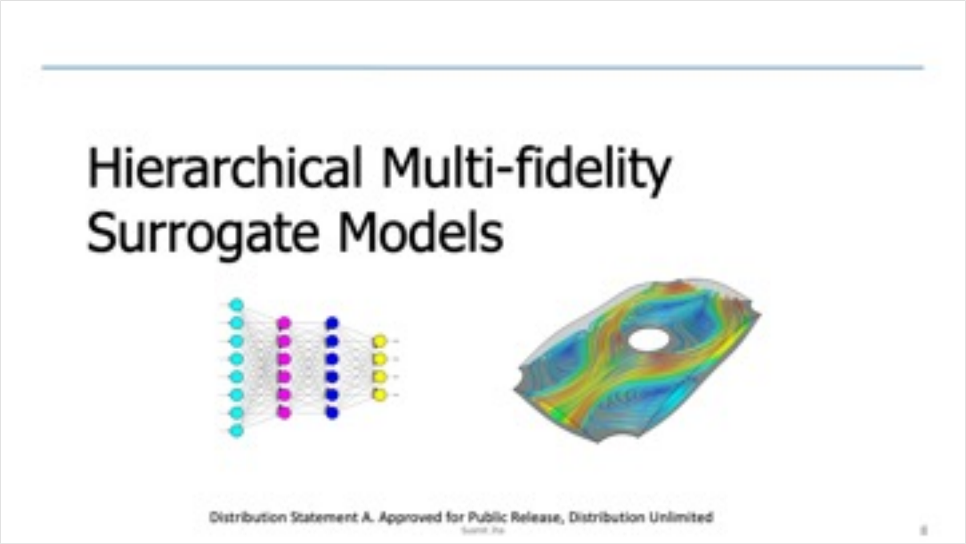
- Two real-world Example Applications
  - Air-domain: AircraftVerse – a new dataset of 27000 aircraft designs
  - Underwater-domain

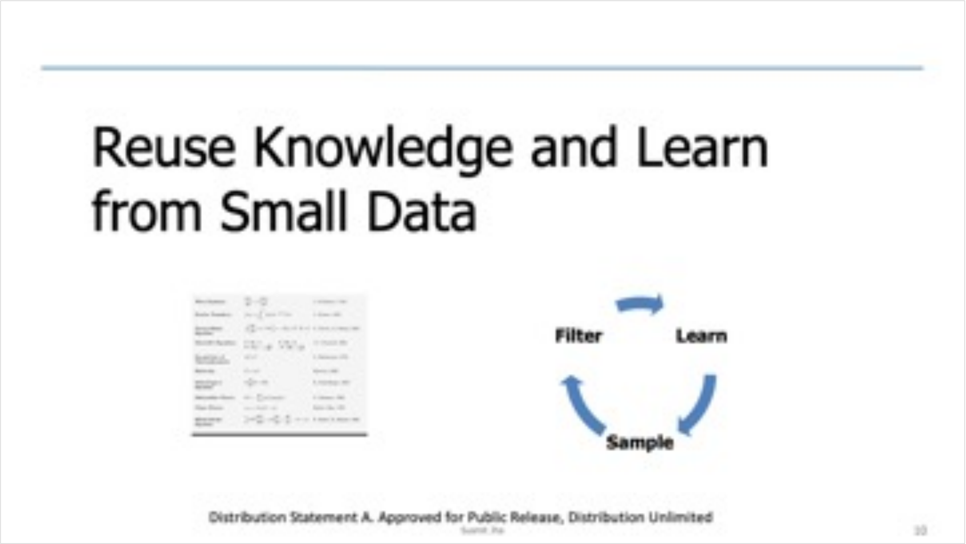
- Hands-on Exercises
  - Familiarization with AircraftVerse Dataset: https://aircraftverse.onrender.com/
  - Satisfiability Solvers Based Planning
  - Use of Satisfiability Solving for Packing in Underwater Vehicles

Distribution Statement A. Approved for Public Release, Distribution Unlimited

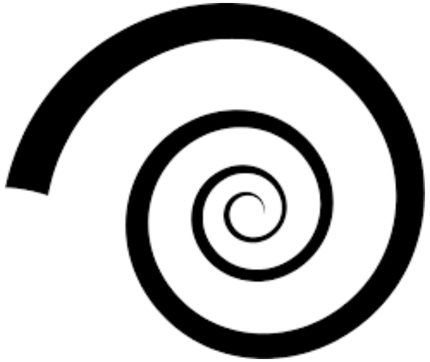
Key Challenges in CPS Design and Desiderata of an AI Co-Designer







Design is a Sequential and Iterative Decision-Making Process



Design is a Sequential and Iterative Decision-Making Process

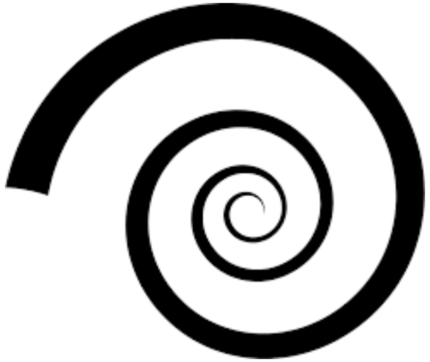
- Design process is a sequential and iterative decision-making process.

- Compositional representation and exploration of design

  - Hierarchy of parts and subsystems. E.g., faring and pressure-vessel design in UUV design
  - Composition of aspects / domains / physics. E.g., design of propulsion subsystem, pitch/roll subsystem

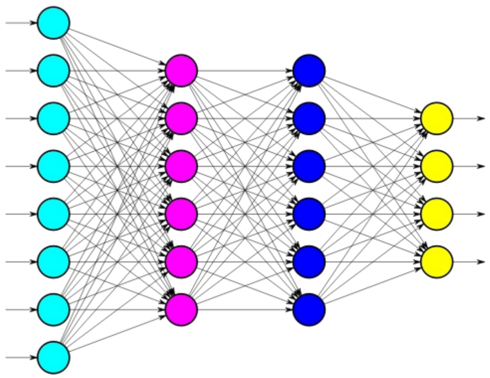
- Incomplete specification that is refined during the design process. Need to avoid over-concretization.

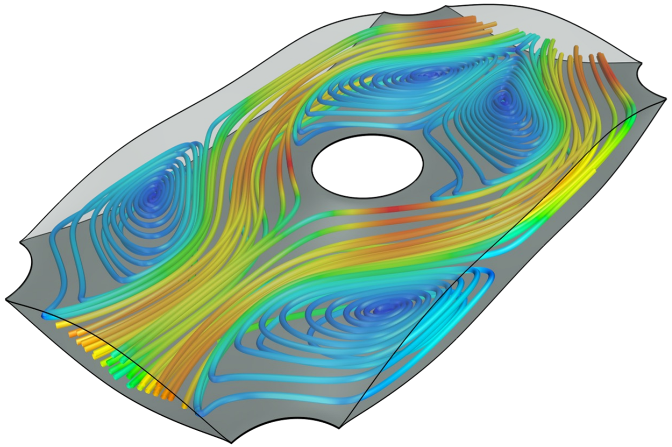
  - In place of single optimum design, discover a diverse set of designs.



Distribution Statement A. Approved for Public Release, Distribution Unlimited

Hierarchical Multi-fidelity Surrogate Models

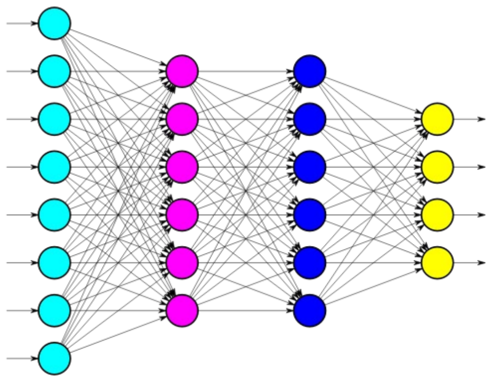


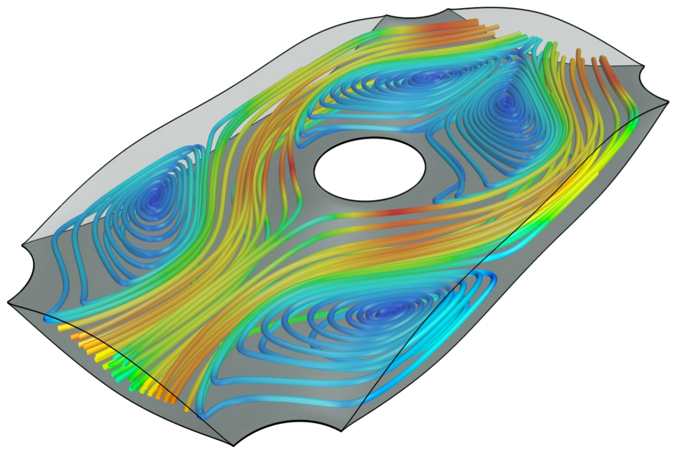


Distribution Statement A. Approved for Public Release, Distribution Unlimited

AI-driven Design needs Hierarchical Multi-fidelity Surrogate Models

- AI codesigner can query models (physics or ML-based) or humans as oracles to guide their decision steps.
- Physics-based scientific models (e.g., computational fluid dynamics models) are slow to evaluate.
- Need to accelerate these models with learned hierarchical surrogates.
    - The accuracy/fidelity/cost of learned surrogates are different at different stages of the design.
    - Some design-decisions need component/subsystem-specific surrogate model.

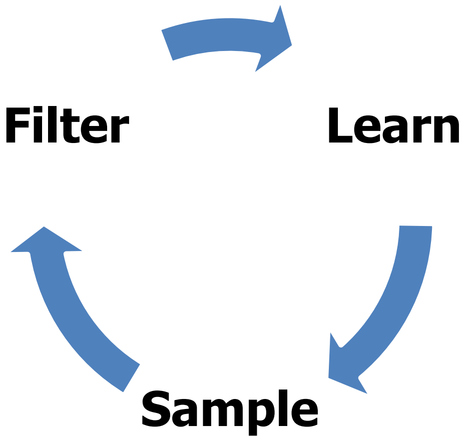




Distribution Statement A. Approved for Public Release, Distribution Unlimited

Reuse Knowledge and Learn from Small Data

Wave Equation	$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$	J. d'Almbert, 1746
Fourier Transform	$f(\omega) = \int_{\infty}^{\infty} f(x) e^{-2\pi i x \omega} \mathrm{d}\, x$	J. Fourier, 1822
Navier-Stokes Equation	$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \nabla \cdot \mathbf{T} + \mathbf{f}$	C. Navier, G. Stokes, 1845
Maxwell's Equations	$\nabla \cdot \mathbf{E} = 0 \quad \nabla \cdot \mathbf{H} = 0$ $\nabla \times \mathbf{E} = -\frac{1}{c}\frac{\partial \mathbf{H}}{\partial t} \quad \nabla \times \mathbf{H} = \frac{1}{c}\frac{\partial E}{\partial t}$	J.C. Maxwell, 1865
Second Law of Thermodynamics	$\mathrm{d}S \geq 0$	L. Boltzmann, 1874
Relativity	$E = mc^2$	Einstein, 1905
Schrodinger's Equation	$i\hbar \frac{\partial}{\partial t} \Psi = H \Psi$	E. Schrodinger, 1927
Information Theory	$H = -\sum p(x) \log p(x)$	C. Shannon, 1949
Chaos Theory	$x_{t+1} = k x_t (1 - x_t)$	Robert May, 1975
Black-Scholes Equation	$\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} - rV = 0$	F. Black, M. Scholes, 1990



Distribution Statement A. Approved for Public Release, Distribution Unlimited

AI-codesigner needs to reuse knowledge and learn from small data

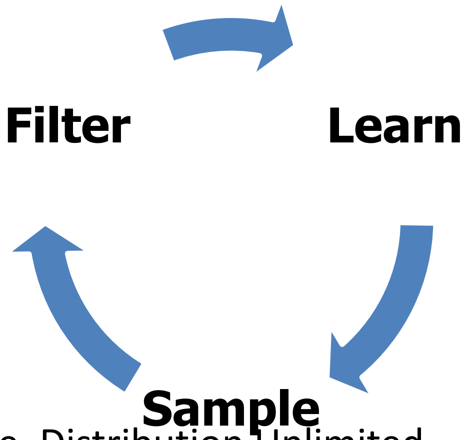
- AI codesigner needs to reuse existing domain knowledge : basic physics, domain-specific design rules, human intuition such as seed designs.

- The AI designer needs to not just optimize for a single design problem against a fixed specification but instead be an adaptive agent learning across design iterations and problems.

- Evaluation of design is costly, and the number of design epochs are few. Hence, AI co-designer needs to learn from small data.

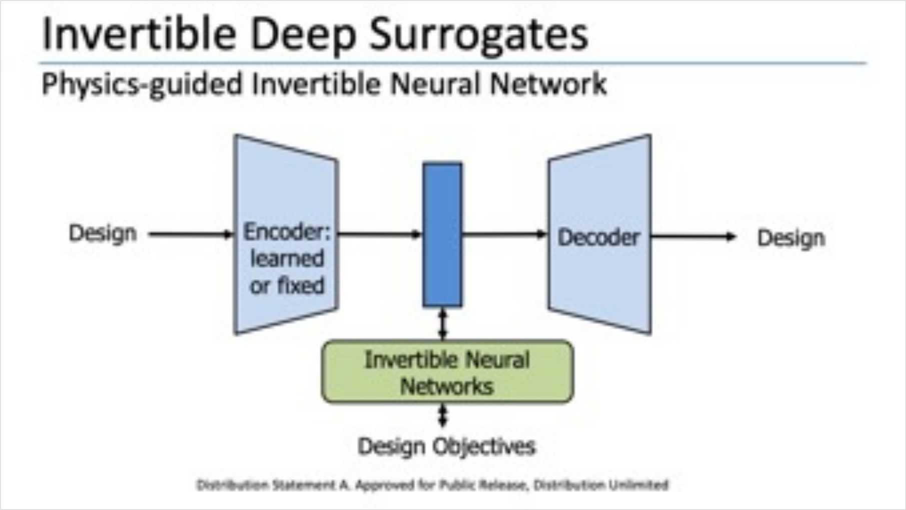
  - Need methods to filter and lift failures to rules.
  - Need for smarter and adaptive data-generation

Wave Equation	$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$	J. d'Almbert, 1746
Fourier Transform	$f(\omega) = \int_{\infty}^{\infty} f(x) e^{-2\pi i x \omega} \mathrm{d}\, x$	J. Fourier, 1822
Navier-Stokes Equation	$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -\nabla p + \nabla \cdot \mathbf{T} + \mathbf{f}$	C. Navier, G. Stokes, 1845
Maxwell's Equations	$\nabla \cdot \mathbf{E} = 0$ $\quad \nabla \cdot \mathbf{H} = 0$ $\nabla \times \mathbf{E} = -\frac{1}{c}\frac{\partial \mathbf{H}}{\partial t}$ $\quad \nabla \times \mathbf{H} = \frac{1}{c}\frac{\partial E}{\partial t}$	J.C. Maxwell, 1865
Second Law of Thermodynamics	$\mathrm{d}S \geq 0$	L. Boltzmann, 1874
Relativity	$E = mc^2$	Einstein, 1905
Schrodinger's Equation	$i\hbar \frac{\partial}{\partial t}\Psi = H\Psi$	E. Schrodinger, 1927
Information Theory	$H = -\sum p(x) \log p(x)$	C. Shannon, 1949
Chaos Theory	$x_{t+1} = k x_t (1 - x_t)$	Robert May, 1975
Black-Scholes Equation	$\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} - rV = 0$	F. Black, M. Scholes, 1990

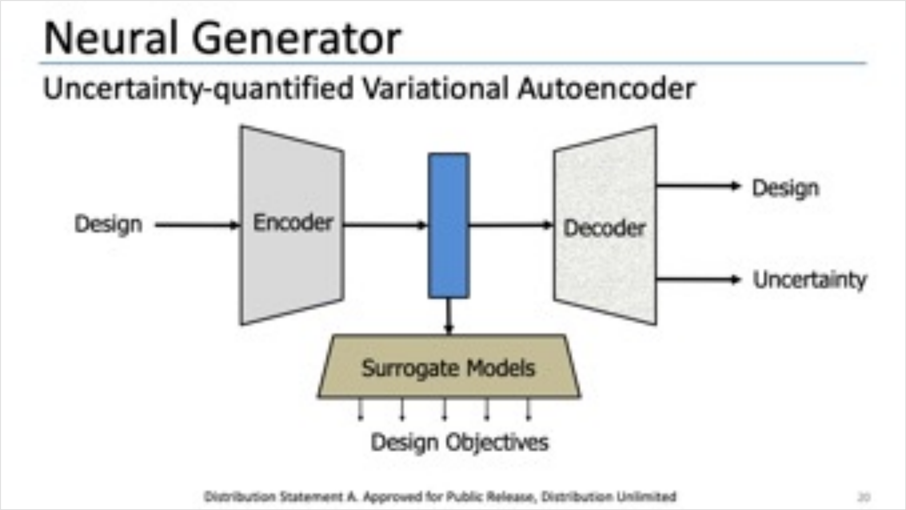


Distribution Statement A. Approved for Public Release, Distribution Unlimited

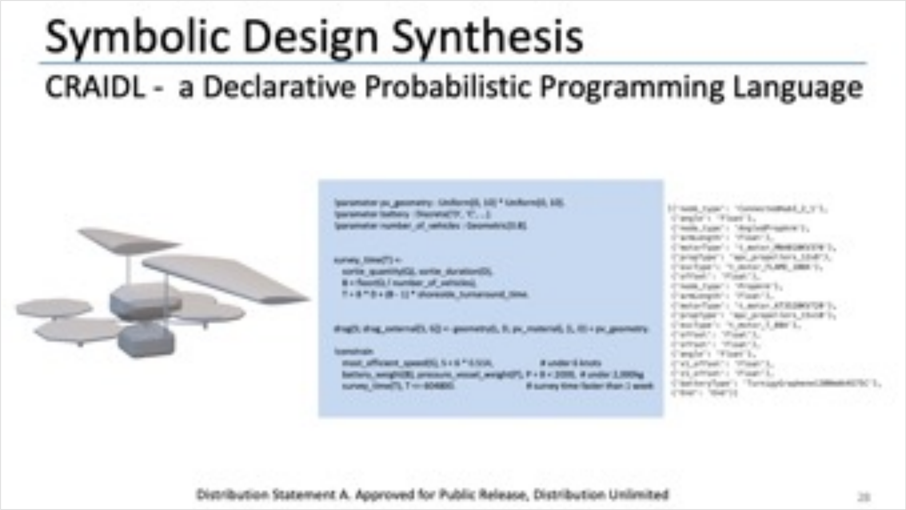
AI in Design



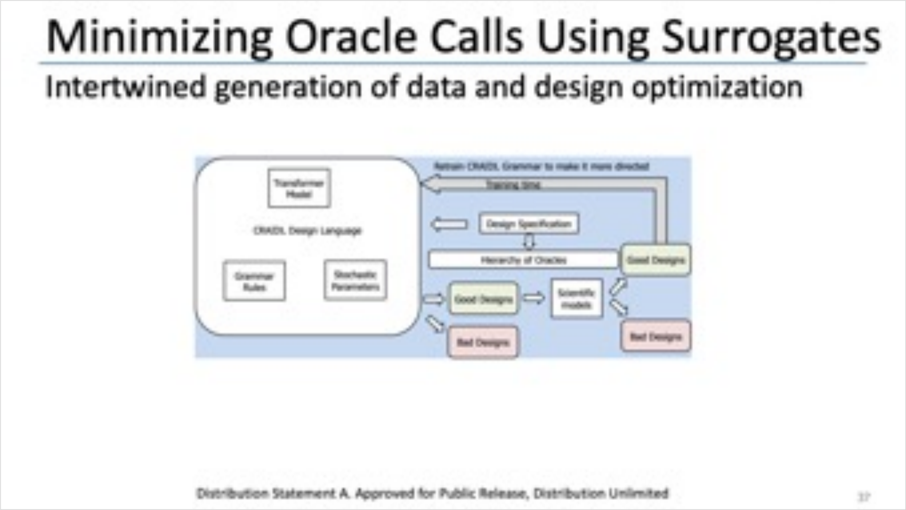
Invertible Deep Surrogates
Physics-guided Invertible Neural Network
Design
Encoder: learned or fixed
Decoder
Design
Invertible Neural Networks
Design Objectives
Distribution Statement A. Approved for Public Release, Distribution Unlimited



Neural Generator
Uncertainty-quantified Variational Autoencoder
Design
Encoder
Decoder
Design
Uncertainty
Surrogate Models
Design Objectives
Distribution Statement A. Approved for Public Release, Distribution Unlimited



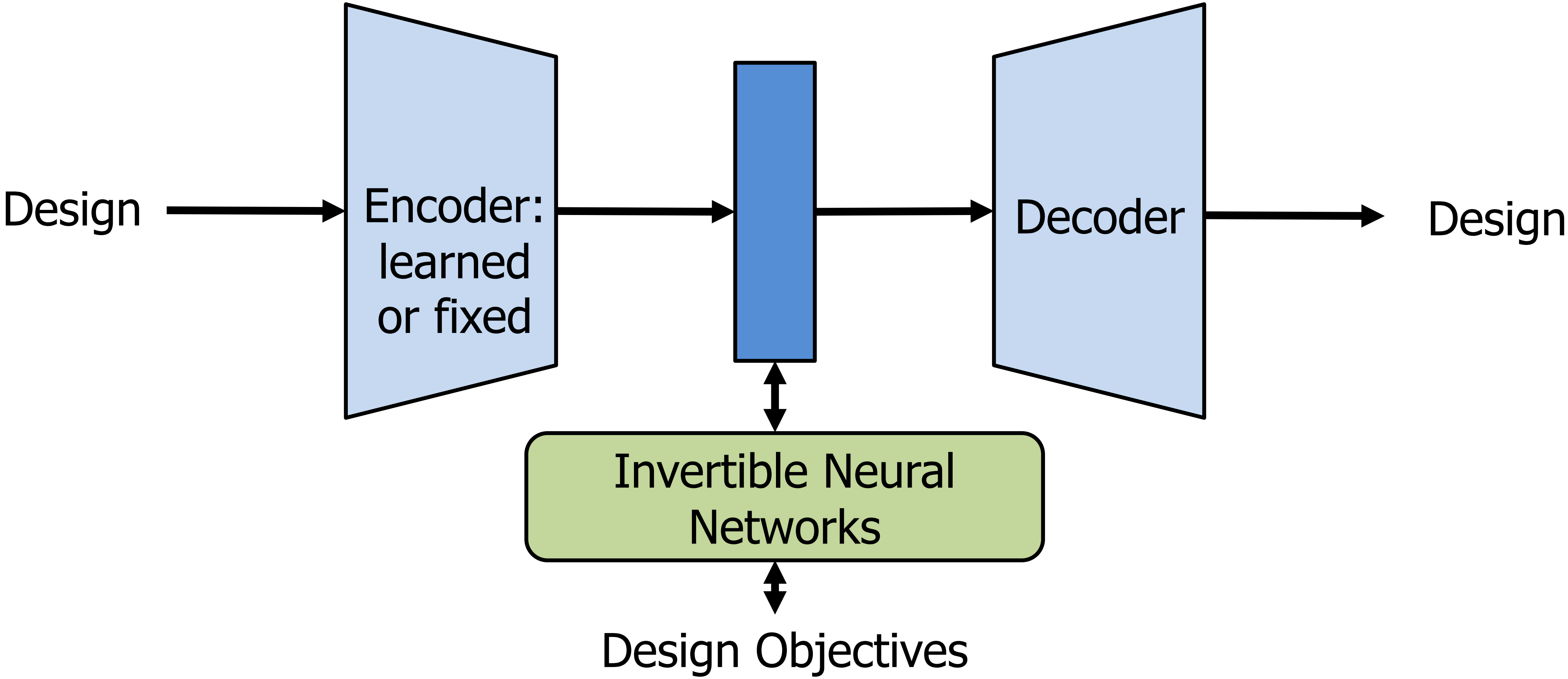
Symbolic Design Synthesis
CRAIDL - a Declarative Probabilistic Programming Language
Distribution Statement A. Approved for Public Release, Distribution Unlimited



Minimizing Oracle Calls Using Surrogates
Intertwined generation of data and design optimization
Distribution Statement A. Approved for Public Release, Distribution Unlimited

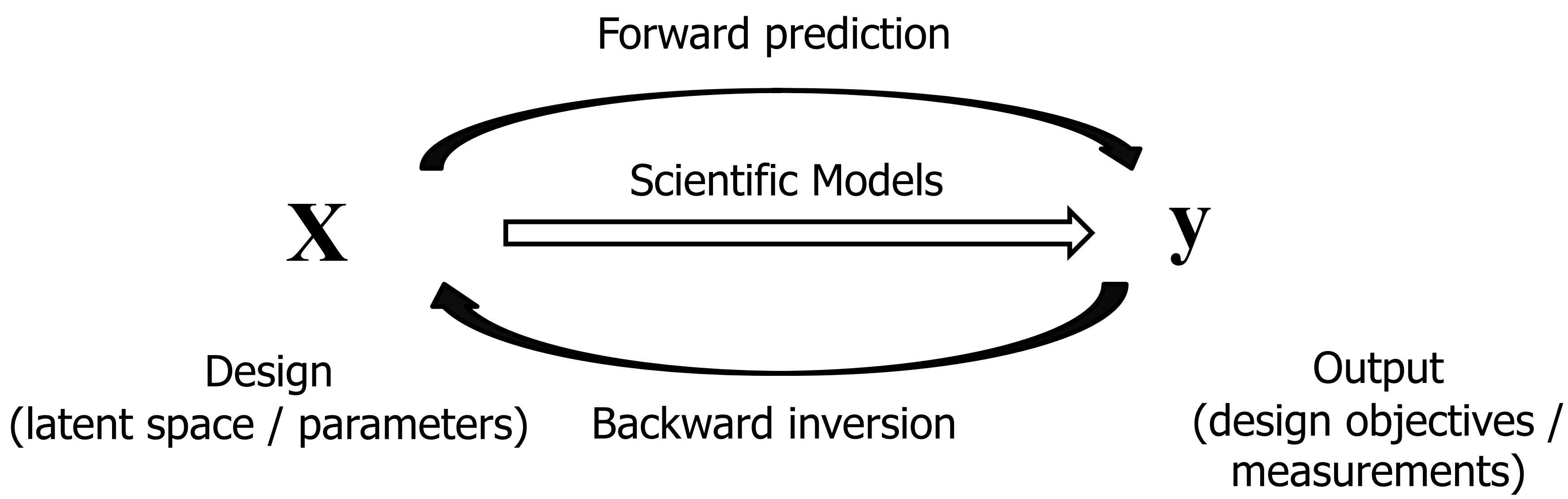
Invertible Deep Surrogates

Physics-guided Invertible Neural Network



Distribution Statement A. Approved for Public Release, Distribution Unlimited

Design as an Inverse Problem

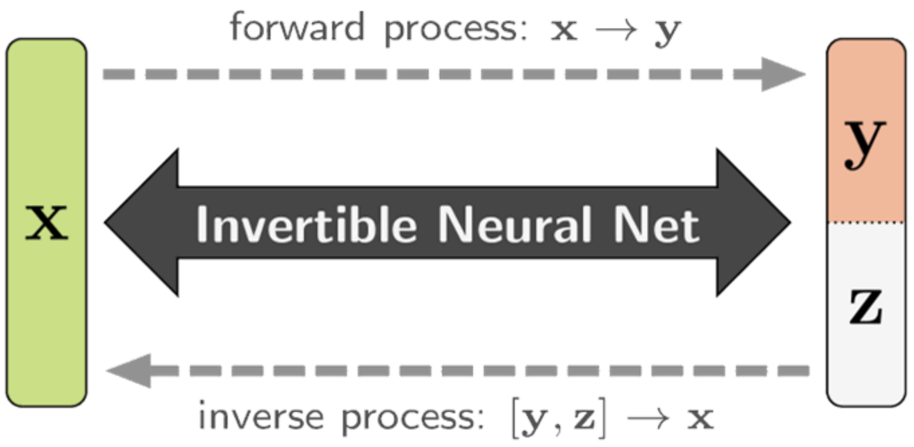


Inverse problem: Generate design corresponding to a given design objective

In many applications, the **forward scientific model can be slow/expensive** and can only be sparsely sampled, thus an inverse model can help generate novel designs with desired specifications

While the forward model is well-defined and can be approximated in a supervised learning setup, the inv**erse problem is ill-defined due to the 'many-to-one' mapping** from inputs to outputs.

Invertible Neural Networks



Invertible neural network framework with inputs (X), outputs (y) and latent variable (z)

As there is an information loss during the forward process *f(x ->y)*, latent variable (z) is introduced to capture the information about X that is not present in y

This allows to learn the inverse process as:

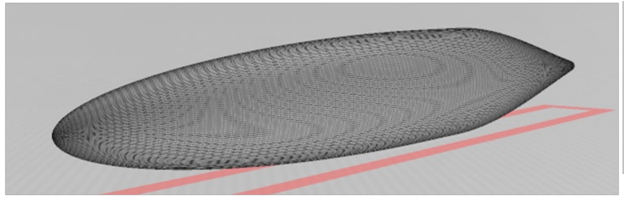
$$\mathbf{x} = f^{-1}(\mathbf{y}, \mathbf{z}) = g(\mathbf{y}, \mathbf{z})$$

where *f()* and *g()* are modeled by invertible neural networks

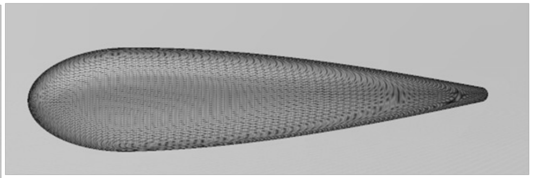
Distribution Statement A. Approved for Public Release, Distribution Unlimited

Example Application: Fairing design

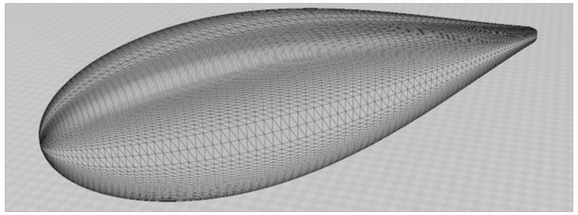
- The goal of fairing design is to generate shapes that
  - experience **less drag** forces,
  - **pack** the required components, and
  - efficiently navigate (**good controllability**) through the mission path
- We consider a **parametric representation** of fairing shapes
  - Parameters control various aspects of the fairing such as nose shape and length, tail shape and length, size and length of the body, curvatures of the body etc.
- We consider the fairing design tool via parameterized **Myring hulls.**



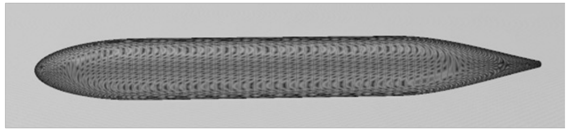
Angle



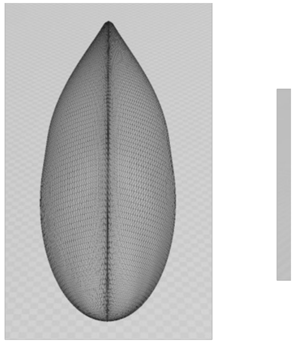
Tear drop



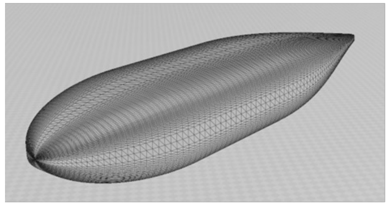
Box fish



Torpedo



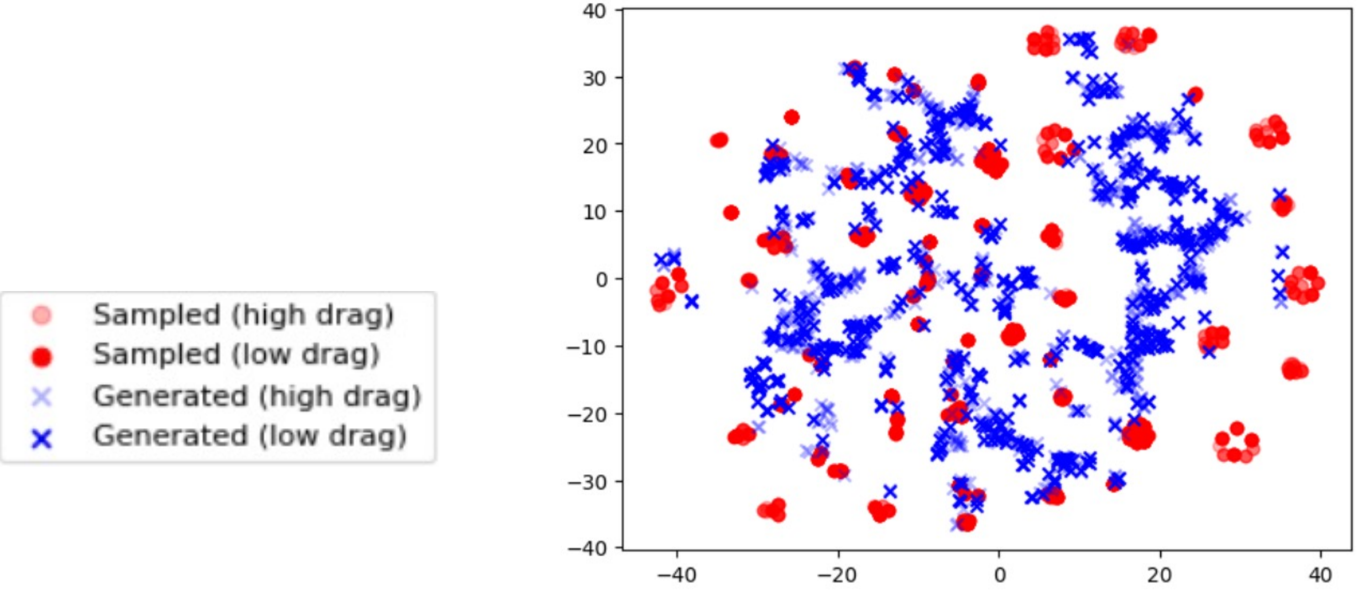
Manta



Whale

INNs can reliably generate diverse fairing designs for given type of shapes

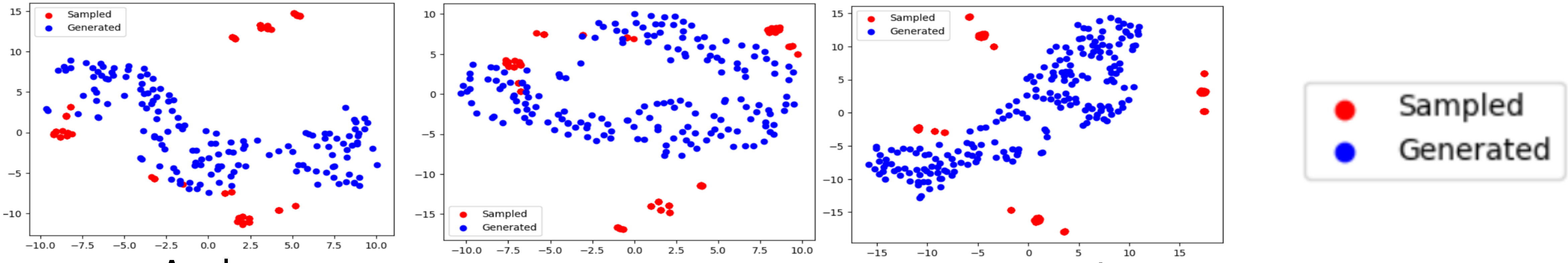
- Average error: 7.48%
- Speed up: order of 1000X



T-SNE plot to compare sampled vs generated fairing shapes

High Diversity

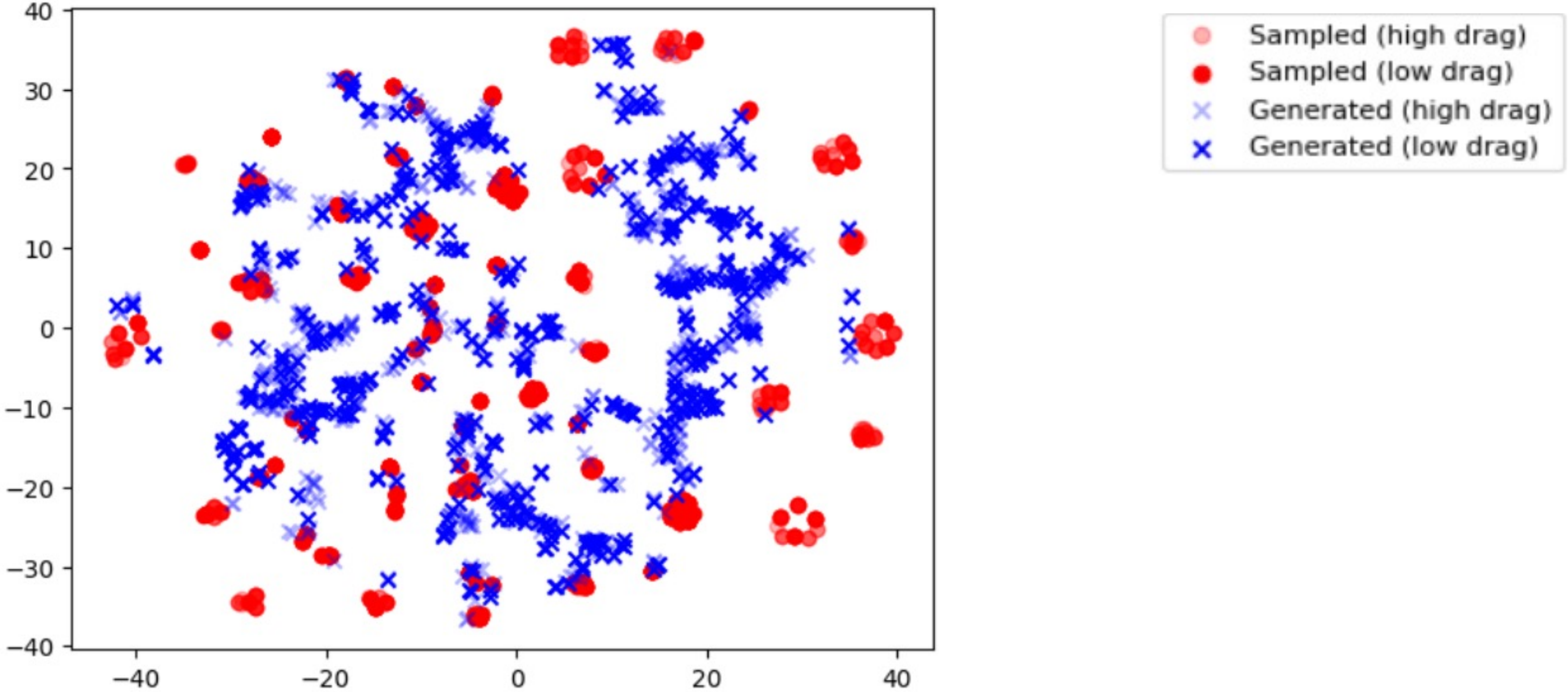
Comparison of diversity between sampled and generated designs



Angle Manta Torpedo

INNs can reliably generate fairing designs for given drag properties

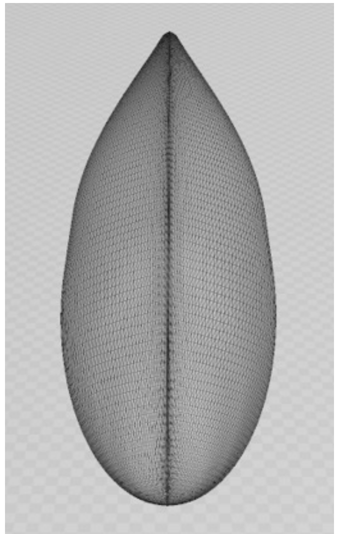
T-SNE plot to compare sampled vs generated fairing shapes



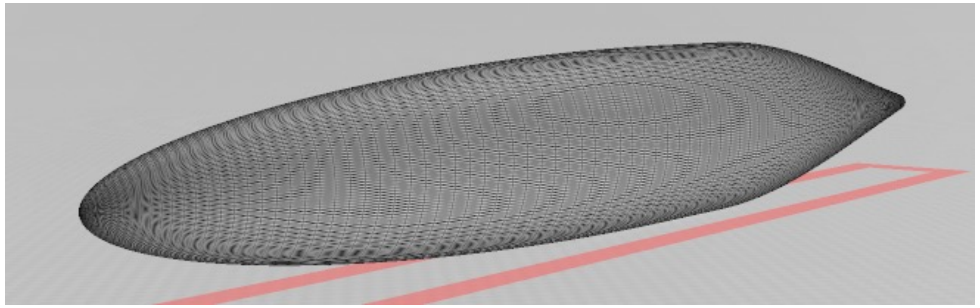
Distribution Statement A. Approved for Public Release, Distribution Unlimited

Discovery of diverse shapes (UUV fairings) of comparable volume with low drag

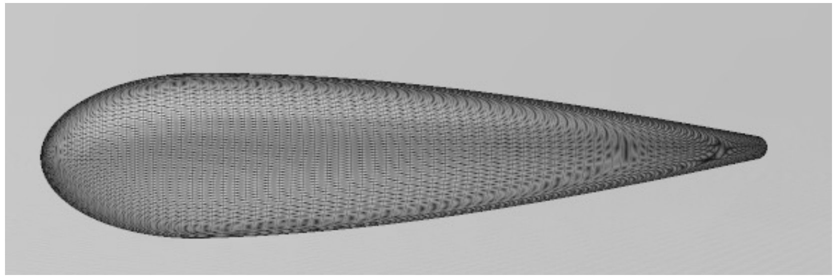
	V = 1	V = 2	V = 5
Angle	0.094 / 57.76	0.087 / 214.66	0.078 / 1197.91
Box	0.048 / 58.38	0.044 / 217.14	0.039 / 1214.79
Manta	0.062 / 56.55	0.058 / 210.75	0.052 /1182.66
Teardrop	0.052 / 53.85	0.049 / 200.70	0.044 / 1126.55
Torpedo	0.090 / 52.82	0.085 / 197.85	0.076/ 1106.09
Whale	0.063 / 58.83	0.0.58 / 217.50	0.052 / 1220.82



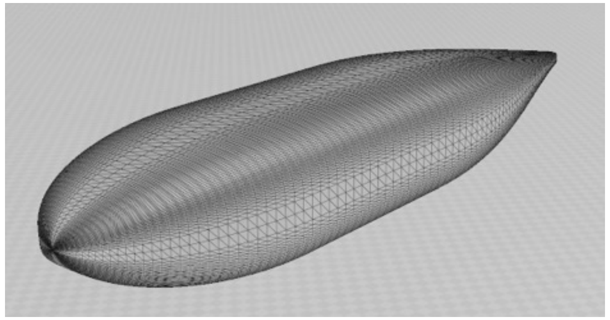
Manta



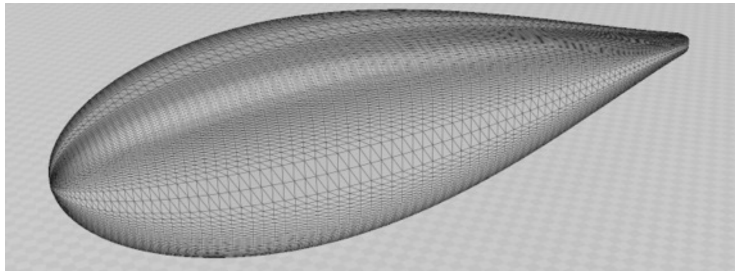
Angle



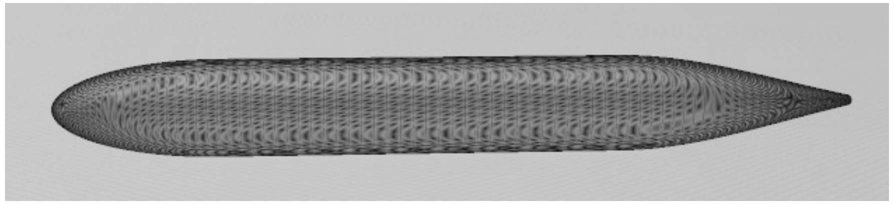
Tear drop



Whale



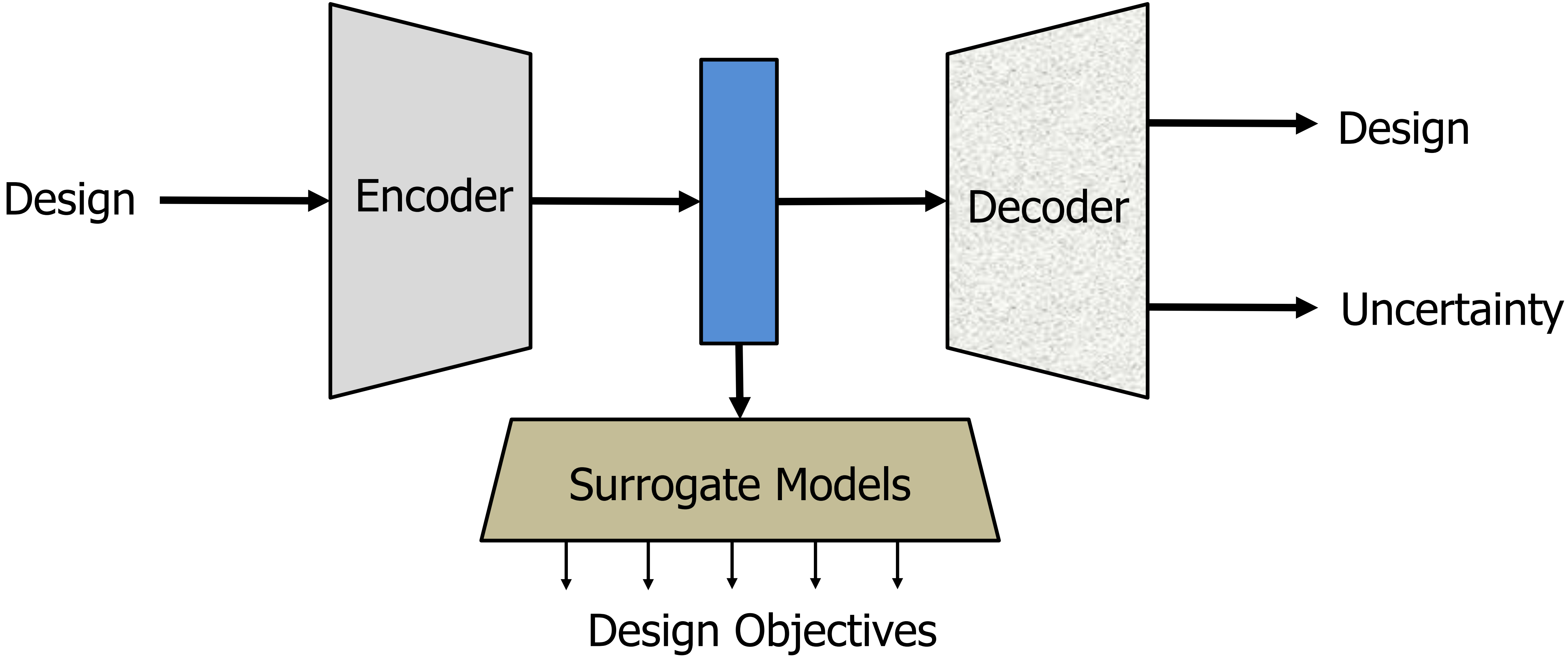
Box fish



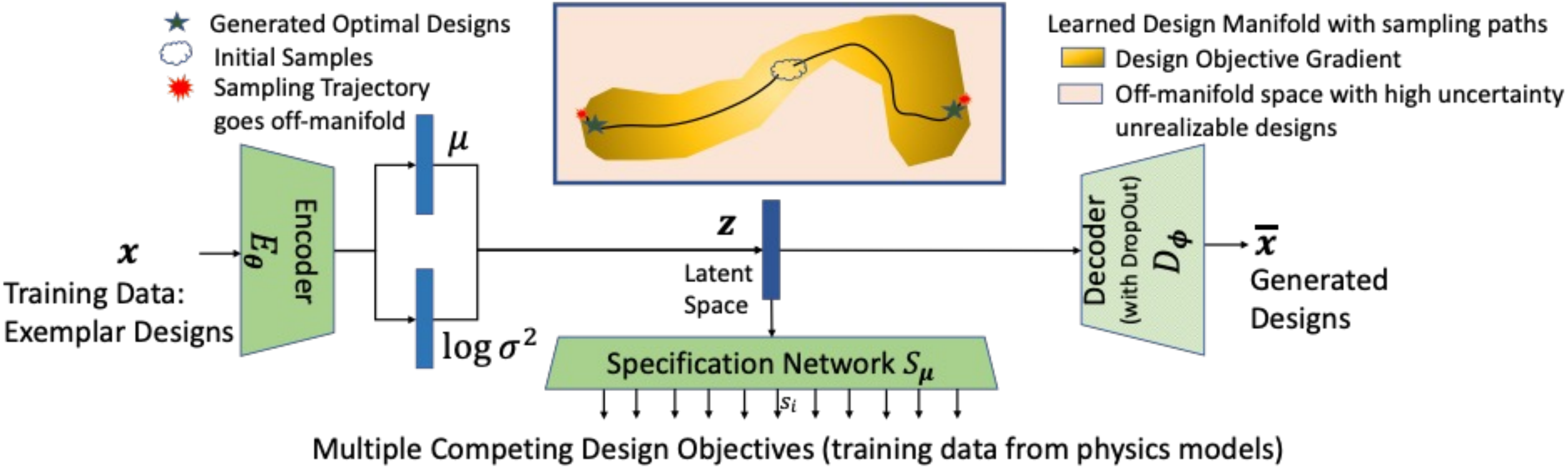
Torpedo

Neural Generator

Uncertainty-quantified Variational Autoencoder

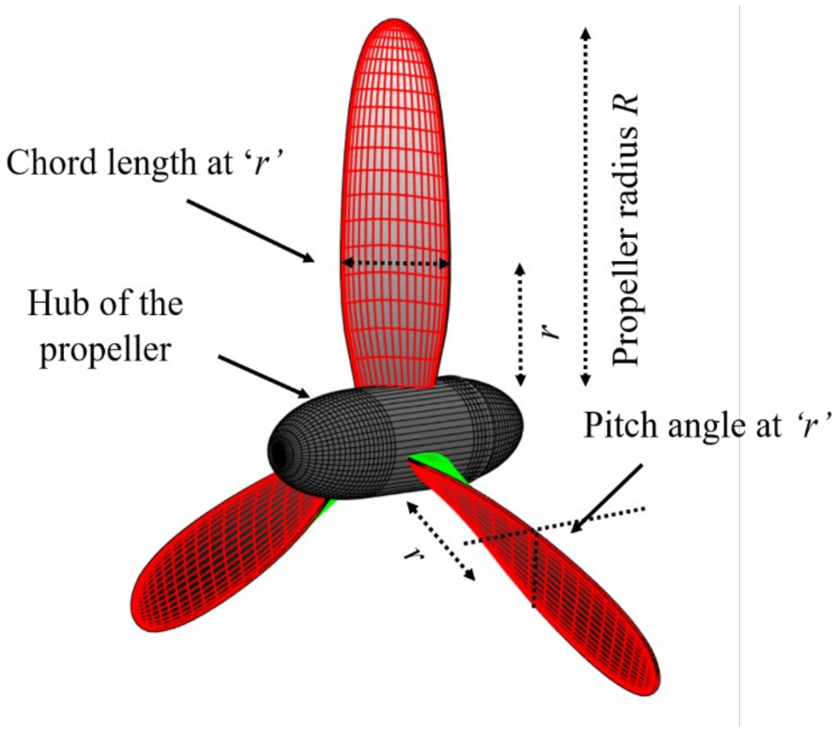


Hamiltonian MCMC over Design Manifold

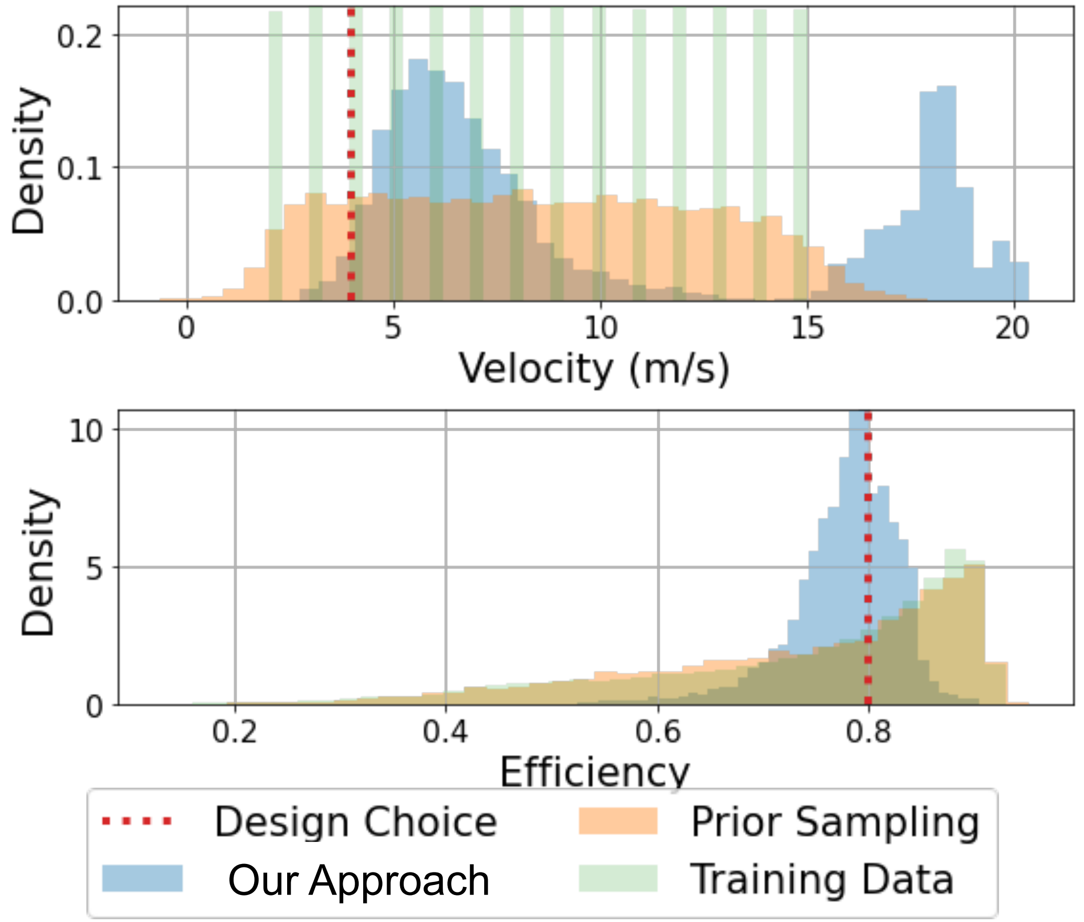


- Our approach uses exemplar designs to learn a variational encoder (VAE) where the decoder is trained with dropout. The specification network predicts the design objectives from the latent space.
- The VAE and the specification network are jointly trained on the exemplar designs and their evaluation on physics models. In the design exploration stage, we condition on the new target design objectives and use temperature annealed HMC to sample the latent space, moving towards optimal designs exploiting the gradient information.
- High variance/uncertainty implies off-manifold designs that may not be unrealizable.
- Controlling HMC walk yields diverse designs.

OpenProp Propeller Design

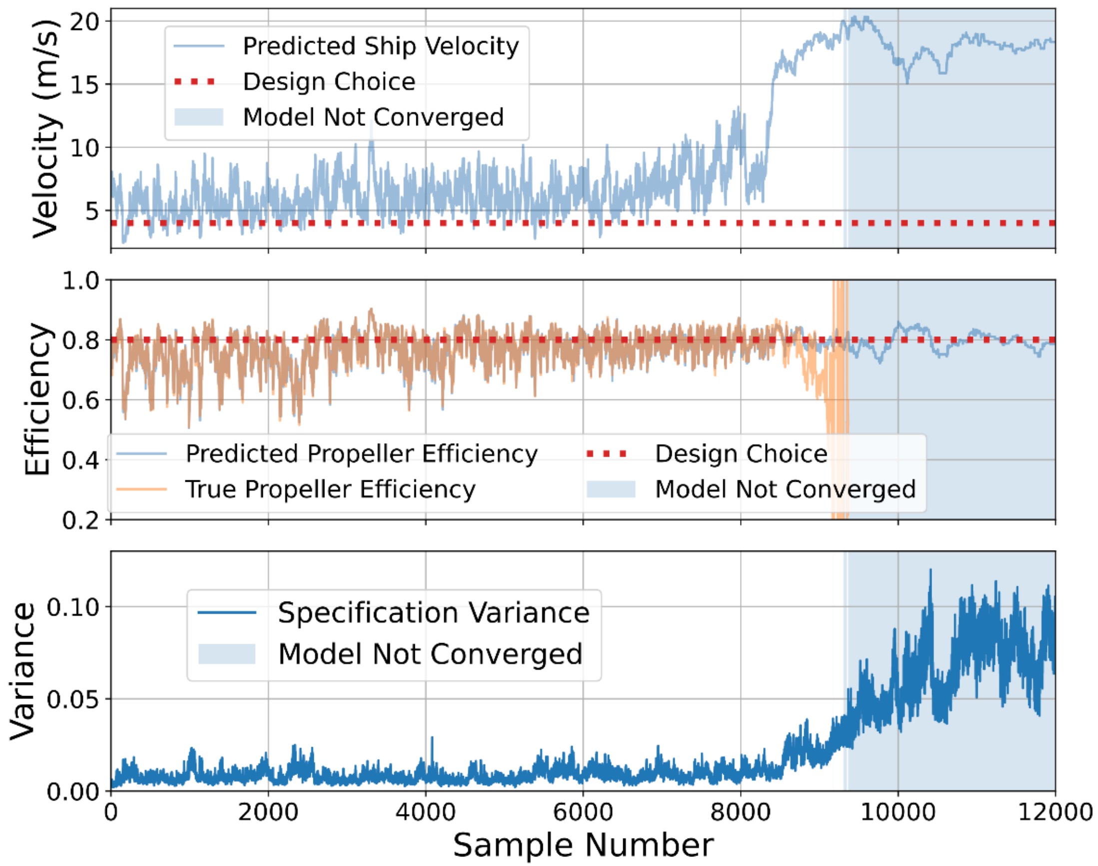


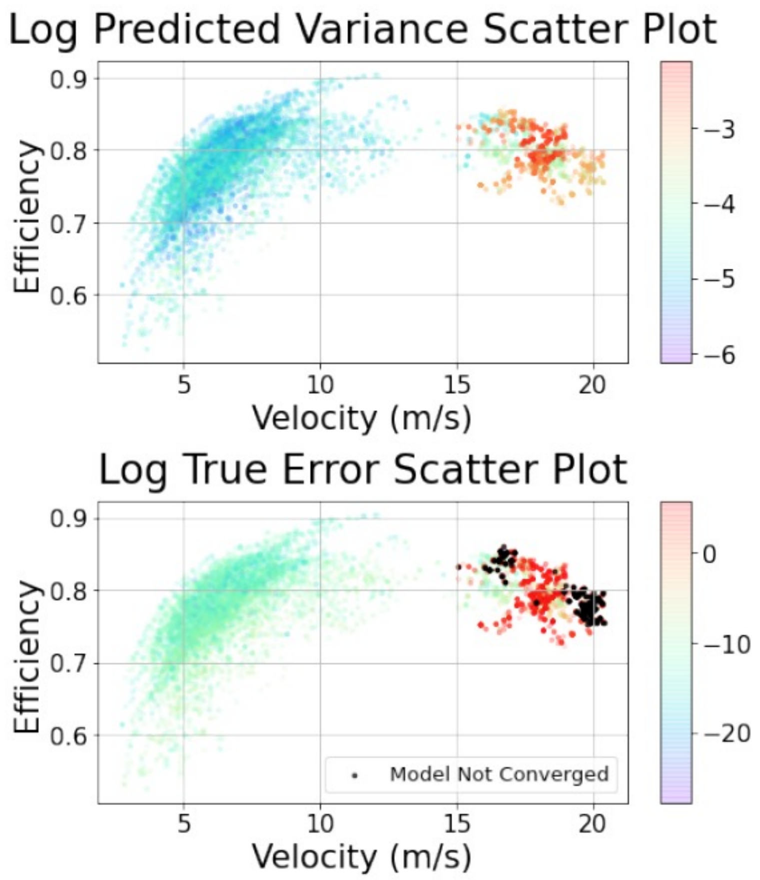
High efficiency at low velocity



Histograms of two competing design objectives. Simply sampling from the Gaussian prior in the latent space is not sufficient.

OpenProp Propeller Design





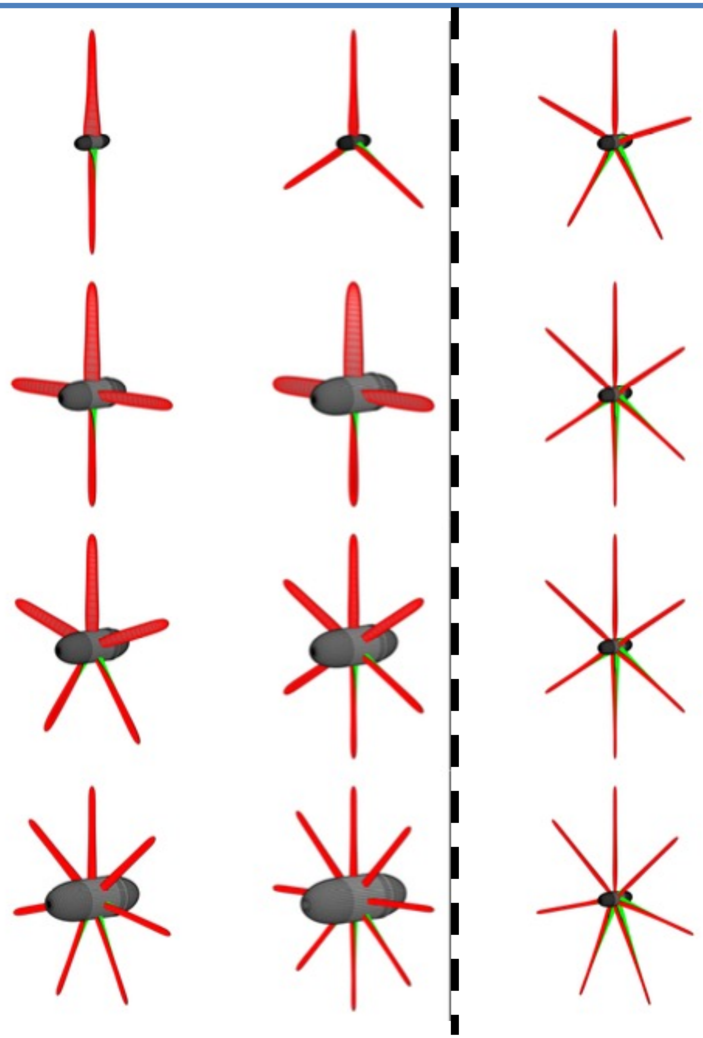
Sample trajectories of the velocity and the propeller efficiency, as well as the corresponding variance on the objectives. Around sample ID 9000, we see high velocities with high efficiency, but the corresponding variance is high, suggesting these are unreliable designs.

Diversity in Propeller Design

Here, our design objective was to generate efficient propellers at a low velocity (efficiency higher than 75% and velocity lower than 4.5 m/s).

Our Approach:

Creates a diverse set of propeller designs in terms number of blades, the shape of blades, pitch angles, and hub diameter.



Direct Optimization (EAs):

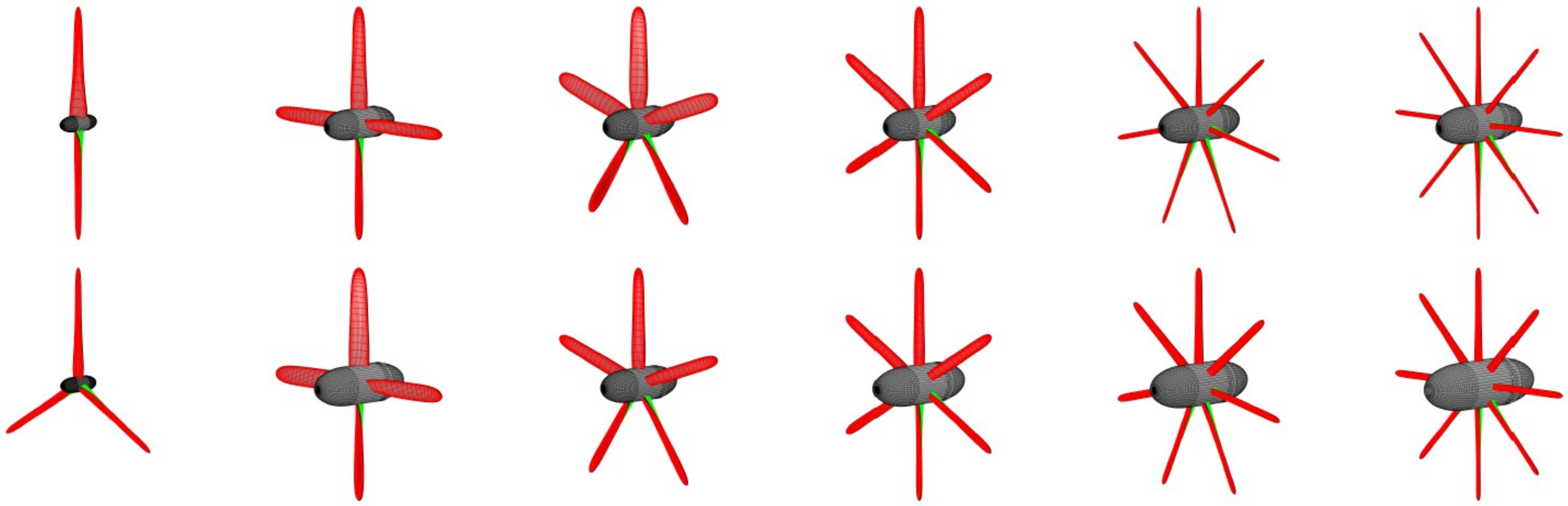
Creates a set of designs with a slight variation in number of blades, but the other geometric properties are similar.

Diversity in Propeller Design

Here, our design objective was to generate efficient propellers at a low velocity (efficiency higher than 75% and velocity lower than 4.5 m/s).

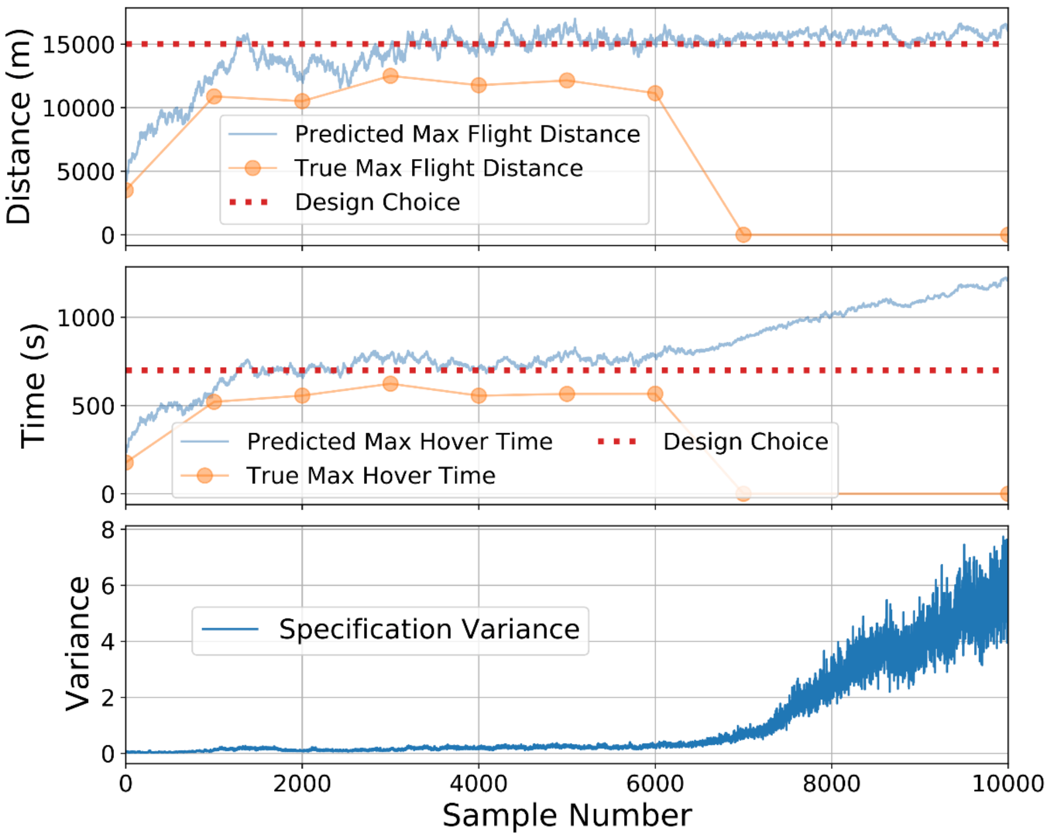
Our Approach:

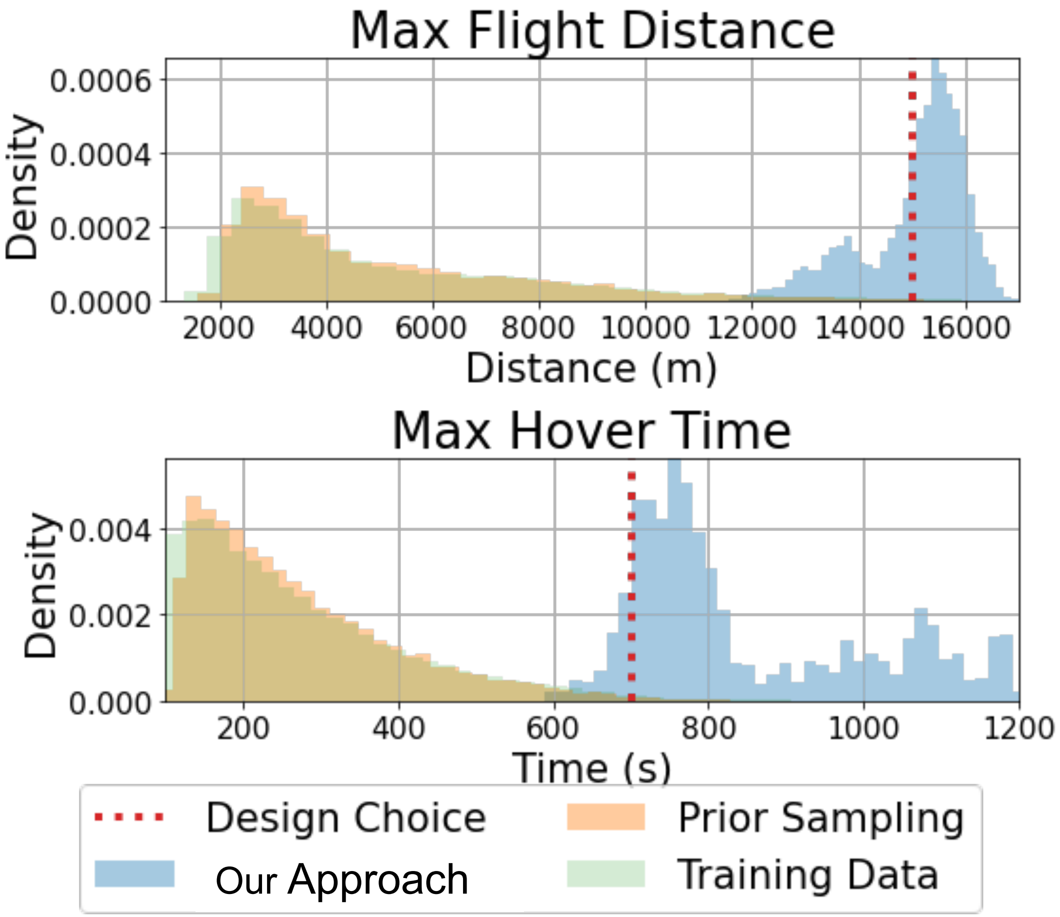
Creates a diverse set of propeller designs in terms number of blades, the shape of blades, pitch angles, and hub diameter.



Noise as a late specification: A 2-blade propeller produces two pressure pulses per revolution, whereas a 3-blade propeller will produce three smaller pulses per revolution for the same amount of total thrust.

Air Vehicle Design



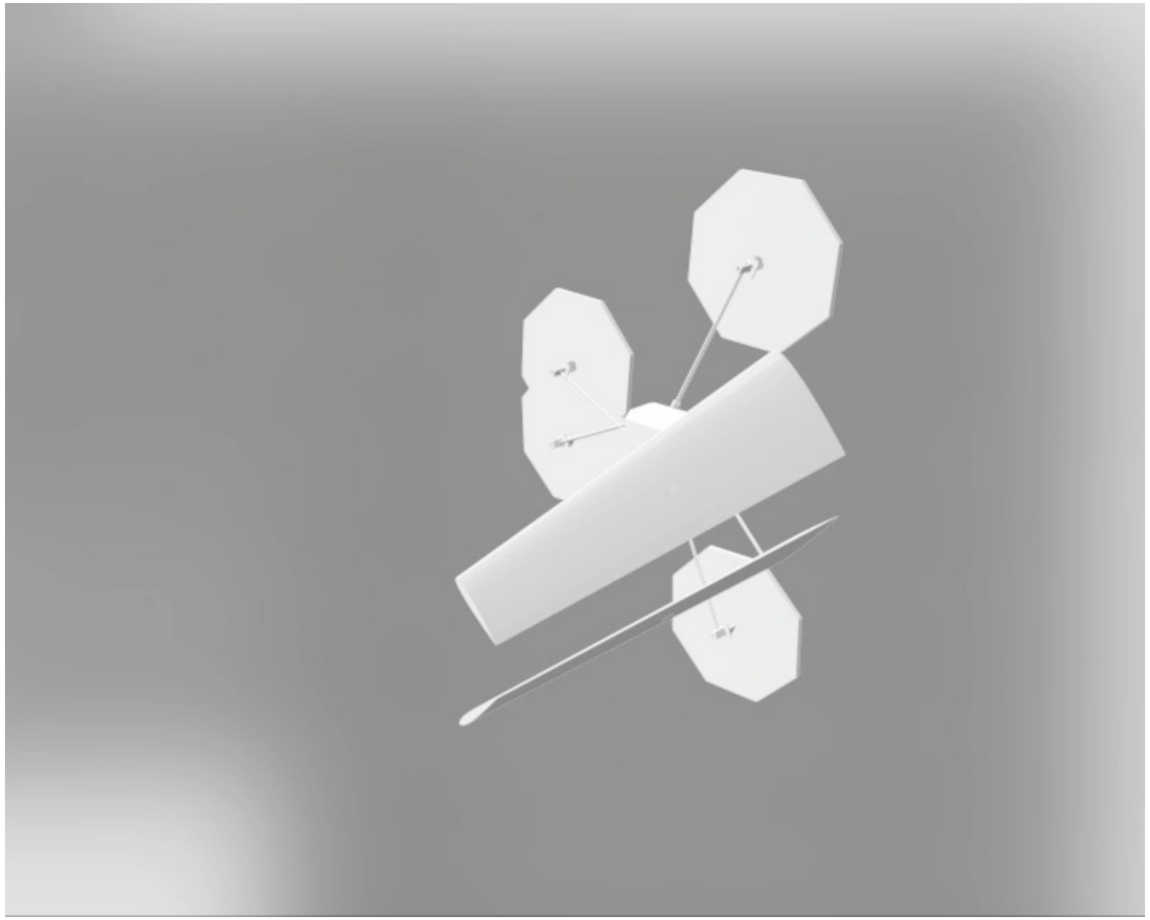


Our approach detects when the generates designs cannot be trusted using the specification variance.

Our approach samples designs that converge around the design objectives even though the training data and the prior distribution are far from it.

Distribution Statement A. Approved for Public Release, Distribution Unlimited

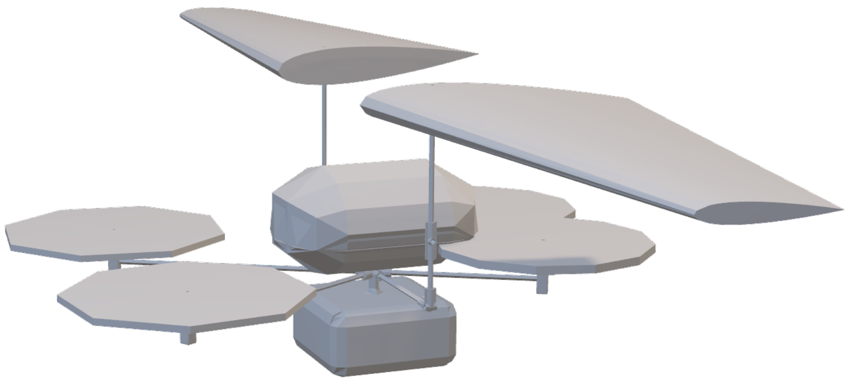
Air Vehicle Design



Distribution Statement A. Approved for Public Release, Distribution Unlimited

Symbolic Design Synthesis

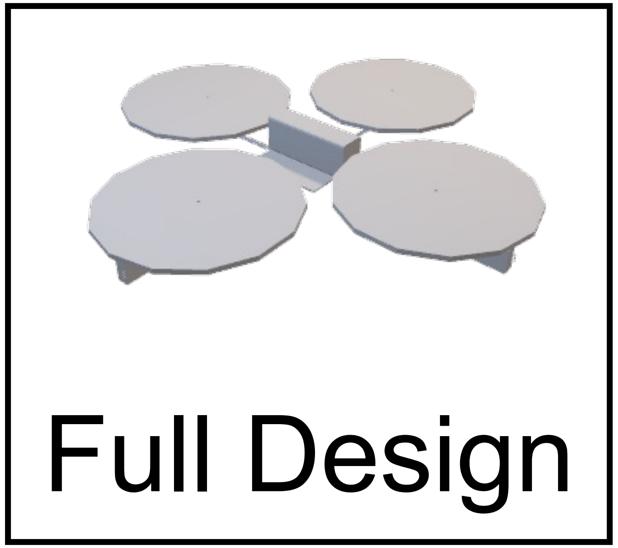
CRAIDL - a Declarative Probabilistic Programming Language

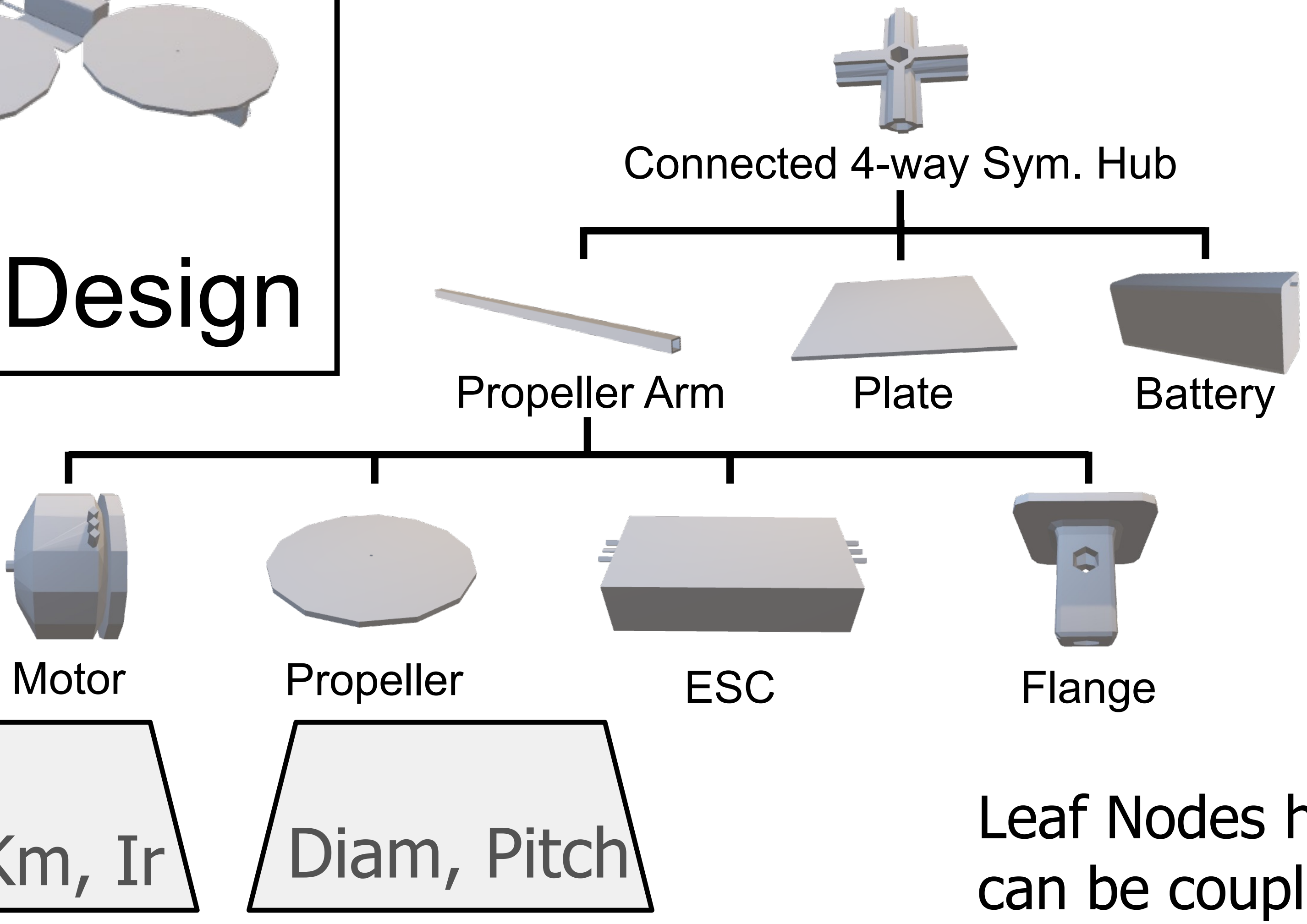


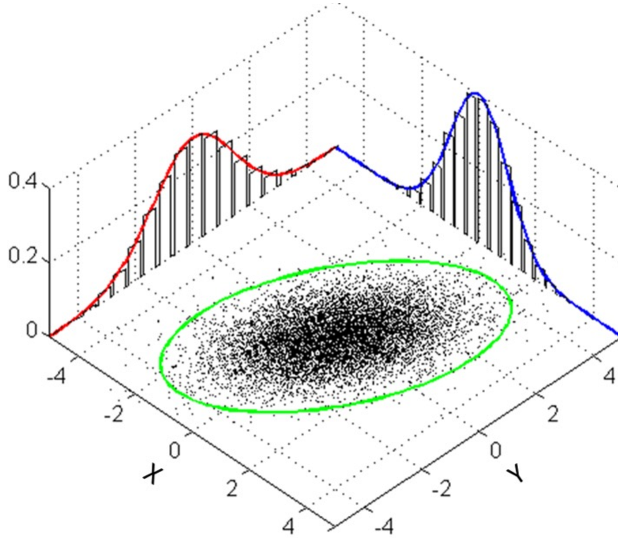
```
!parameter pv_geometry : Uniform[0, 10] * Uniform[0, 10].
!parameter battery : Discrete['D', 'C', ...].
!parameter number_of_vehicles : Geometric[0.8].


survey_time(T) <-
   sortie_quantity(Q), sortie_duration(D),
   B = floor(Q / number_of_vehicles),
   T = B * D + (B - 1) * shoreside_turnaround_time.


drag(S; drag_external[S, G]) <- geometry(L, D, pv_material), (L, D) = pv_geometry.

!constrain
   most_efficient_speed(S), S < 6 * 0.514,                  # under 6 knots
   battery_weight(B), pressure_vessel_weight(P), P + B < 2000,  # under 2,000kg
   survey_time(T), T <= 604800.                             # survey time faster than 1 week
```


```
[{'node_type': 'ConnectedHub3_2_1'},
 {'angle': 'Float'},
 {'node_type': 'AngledPropArm'},
 {'armLength': 'Float'},
 {'motorType': 't_motor_MN4010KV370'},
 {'propType': 'apc_propellers_12x8'},
 {'escType': 't_motor_FLAME_100A'},
 {'offset': 'Float'},
 {'node_type': 'PropArm'},
 {'armLength': 'Float'},
 {'motorType': 't_motor_AT3520KV720'},
 {'propType': 'apc_propellers_15x10'},
 {'escType': 't_motor_T_80A'},
 {'offset': 'Float'},
 {'offset': 'Float'},
 {'angle': 'Float'},
 {'x1_offset': 'Float'},
 {'z1_offset': 'Float'},
 {'batteryType': 'TurnigyGraphene1300mAh4S75C'},
 {'End': 'End'}]
```

Symbolic Generator Using CRAIDL - a Declarative Probabilistic Programming Language







Probability Assigned to Each Design Rule

Leaf Nodes have Attributes that can be coupled

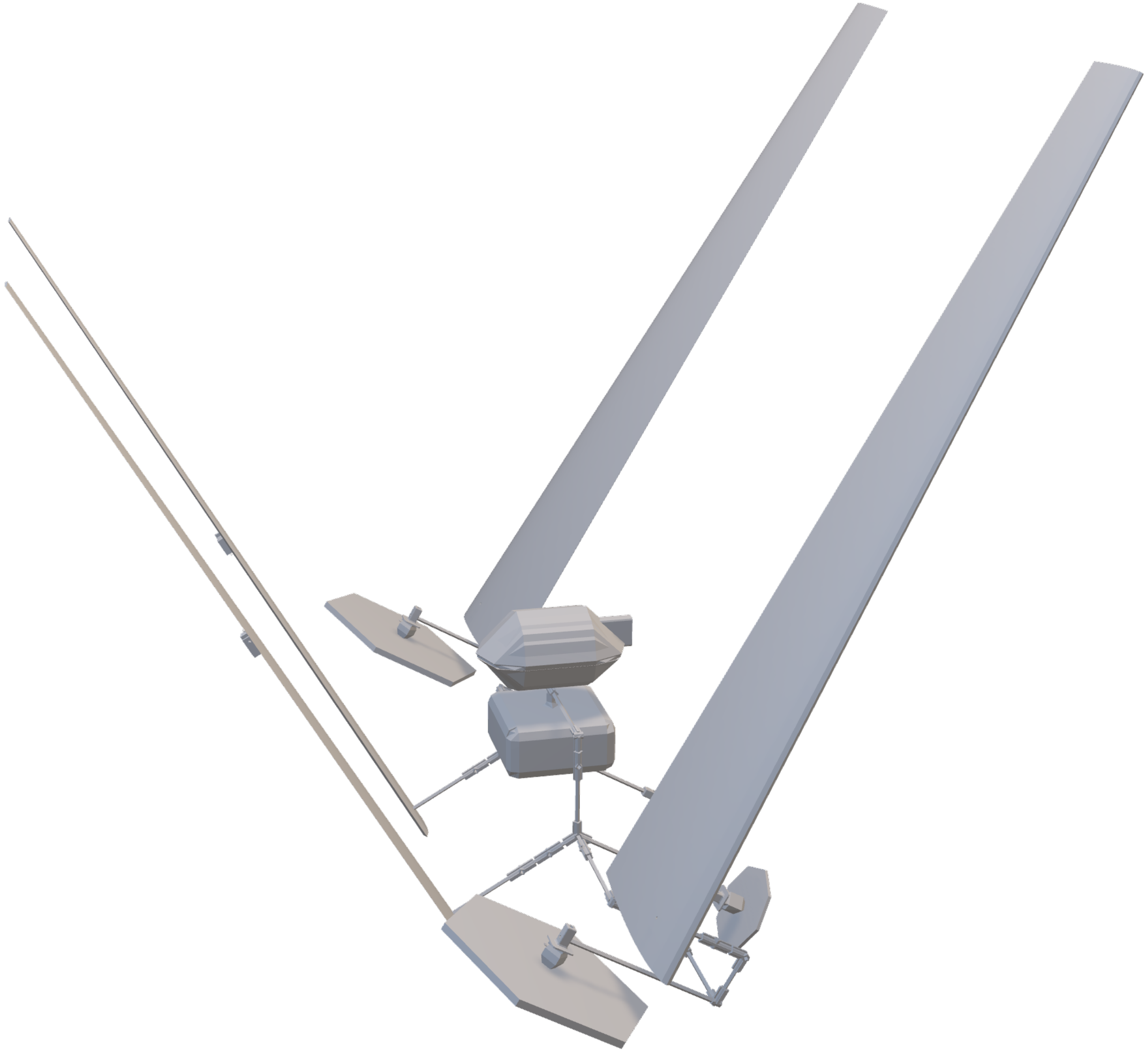
Distribution Statement A. Approved for Public Release, Distribution Unlimited

Symbolic Generator Using CRAIDL - a Declarative Probabilistic Programming Language

Map leaf nodes to corpus components

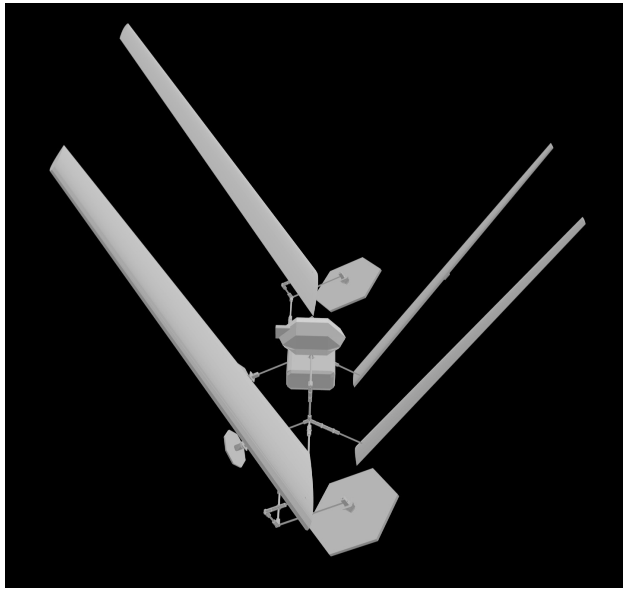
```
def fromSeq(seq: DesignSeq): (ConnectedHub,DesignSeq) = {

def fromSeqBF(seq: DesignSeq): (BFParseResult[ConnectedHub], DesignSeq) = {
  val (node_type, seq2) = readKeyValue[String](seq, "node_type")
  node_type match {
    case "ConnectedHub2_Sym_Long" => ConnectedHub2_Sym_Long.fromSeqBF(seq2)
    case "ConnectedHub2_Sym_Wide" => ConnectedHub2_Sym_Wide.fromSeqBF(seq2)
    case "ConnectedHub2_Asym" => ConnectedHub2_Asym.fromSeqBF(seq2)
    case "ConnectedHub3_Sym" => ConnectedHub3_Sym.fromSeqBF(seq2)
    case "ConnectedHub3_2_1" => ConnectedHub3_2_1.fromSeqBF(seq2)
    case "ConnectedHub4_Sym" => ConnectedHub4_Sym.fromSeqBF(seq2)
    case "ConnectedHub4_Sym_Aligned" => ConnectedHub4_Sym_Aligned.fromSeqBF(seq2)
    case "ConnectedHub4_2_2" => ConnectedHub4_2_2.fromSeqBF(seq2)
    case "ConnectedHub4_1_2_1" => ConnectedHub4_1_2_1.fromSeqBF(seq2)
    case "ConnectedHub5_Sym" => ConnectedHub5_Sym.fromSeqBF(seq2)
    case "ConnectedHub5_4_1" => ConnectedHub5_4_1.fromSeqBF(seq2)
    case "ConnectedHub5_2_2_1" => ConnectedHub5_2_2_1.fromSeqBF(seq2)
    case "ConnectedHub6_Sym" => ConnectedHub6_Sym.fromSeqBF(seq2)
    case "ConnectedHub6_Sym_Aligned" => ConnectedHub6_Sym_Aligned.fromSeqBF(seq2)
    case "ConnectedHub6_2_2_2" => ConnectedHub6_2_2_2.fromSeqBF(seq2)
    case "ConnectedHub6_1_2_2_1" => ConnectedHub6_1_2_2_1.fromSeqBF(seq2)
    case _ => throw new IllegalArgumentException(s"Unexpected node_type for ConnectedHub: $node_type")
  }
}
```

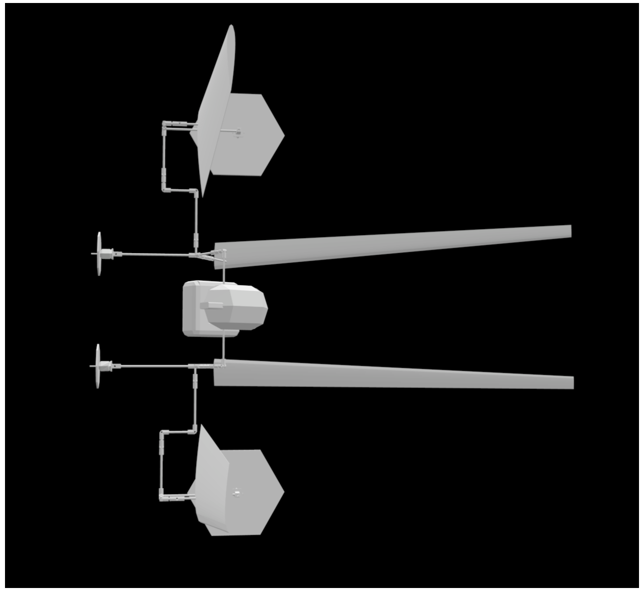
Symbolic Generator Using CRAIDL - a Declarative Probabilistic Programming Language



Symbolic Generator Using CRAIDL - a Declarative Probabilistic Programming Language

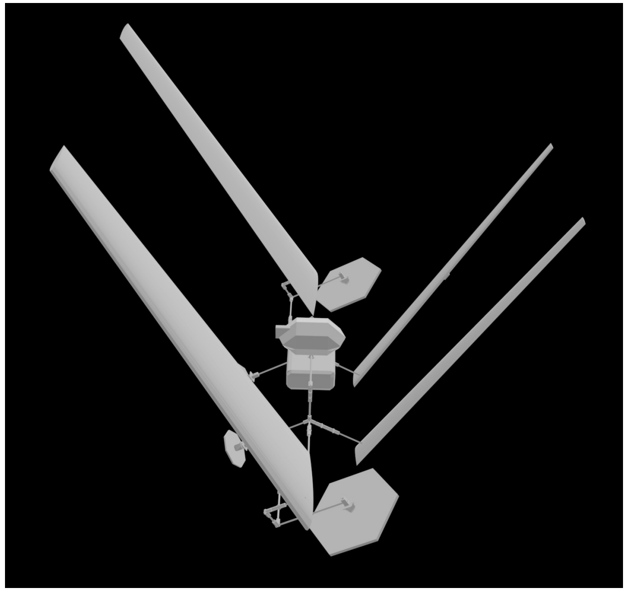

```
generator_version:          "UAV2_gen12"
name:                       "design_296_24d147e5dc334866a789085e817cedee"
▼ hub:
    node_type:              "ConnectedHub2_Sym_Wide"
  ▼ mainSegment:
      node_type:            "BendSegment"
      angle:                -90
      armLength:            239.05569937868356
    ▶ mainSegment:          {…}
▼ fuselageWithComponents:
    node_type:              "SingleBatteryFuselageWithComponents"
  ▶ battery:                {…}
  ▶ fuselage:               {…}
```

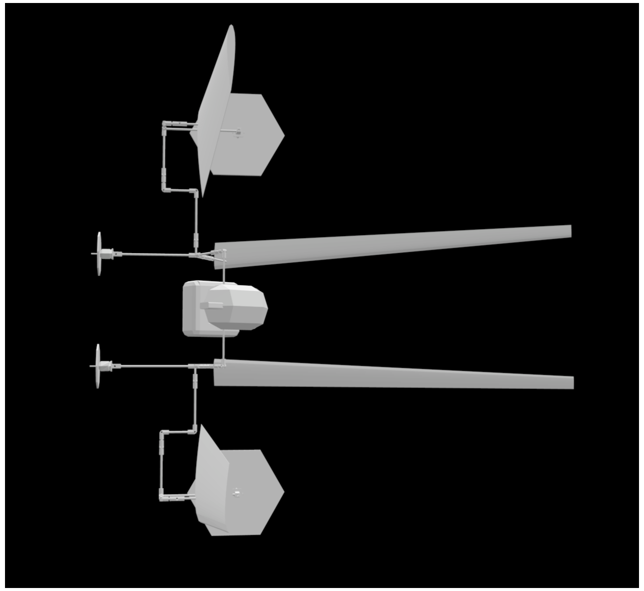





Symbolic Generator Using CRAIDL - a Declarative Probabilistic Programming Language

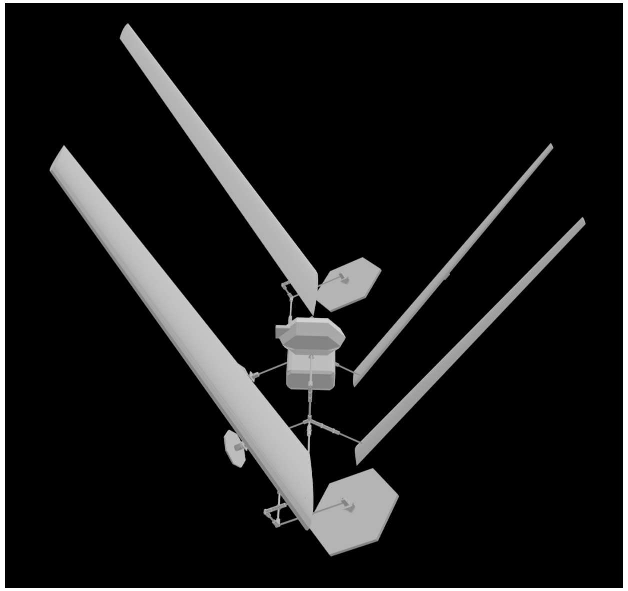

```
generator_version:        "UAV2_gen12"
name:                     "design_296_24d147e5dc334866a789085e817cedee"
▾ hub:
    node_type:            "ConnectedHub2_Sym_Wide"
  ▾ mainSegment:
      node_type:          "BendSegment"
      angle:              -90
      armLength:          239.05569937868356
    ▾ mainSegment:
        node_type:        "BranchWithTopSegment_Asym"
        armLength:        145.4677982914925
        angle:            120
      ▸ leftSegment:      {…}
      ▸ rightSegment:     {…}
      ▸ topSegment:       {…}
▾ fuselageWithComponents:
    node_type:            "SingleBatteryFuselageWithComponents"
  ▸ battery:              {…}
  ▸ fuselage:             {…}
```

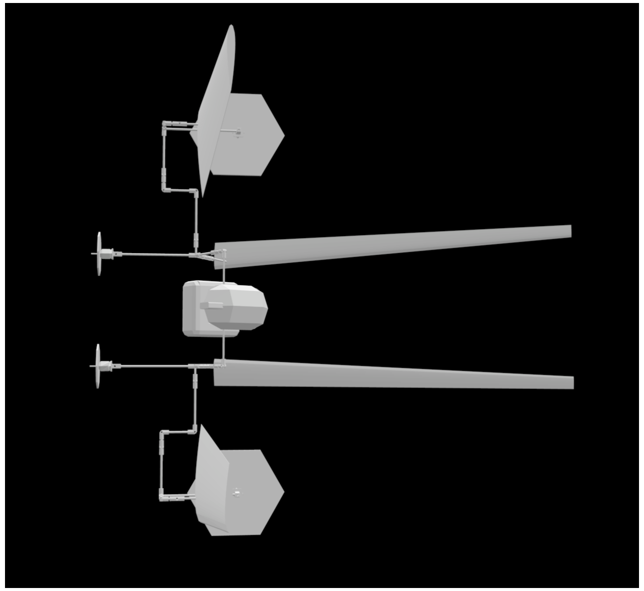





Symbolic Generator Using CRAIDL - a Declarative Probabilistic Programming Language

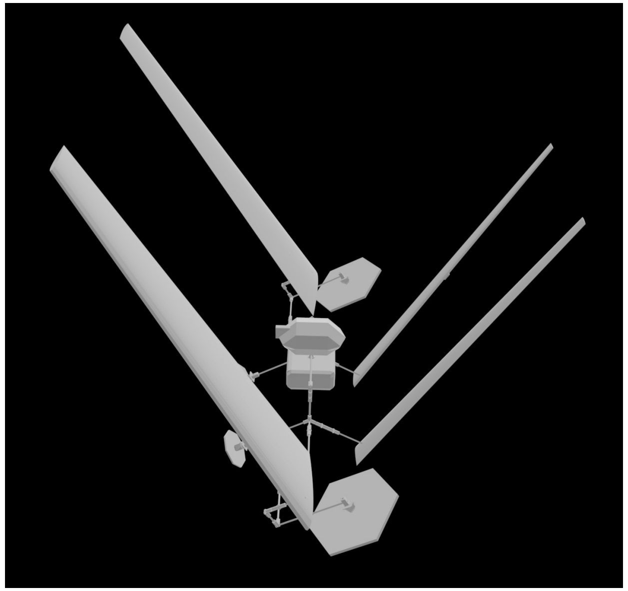

```
generator_version:        "UAV2_gen12"
name:                     "design_296_24d147e5dc334866a789085e817cedee"
▼ hub:
    node_type:            "ConnectedHub2_Sym_Wide"
  ▼ mainSegment:
      node_type:          "BendSegment"
      angle:              -90
      armLength:          239.05569937868356
    ▼ mainSegment:
        node_type:        "BranchWithTopSegment_Asym"
        armLength:        145.4677982914925
        angle:            120
      ▼ leftSegment:
          node_type:      "AngledPropArm"
          armLength:      353.7693911129445
        ▶ motor:          {…}
        ▶ prop:           {…}
      ▼ rightSegment:
          node_type:      "AngledWingArm"
          arm1Length:     52.535344464774575
          arm2Length:     180.60733766227716
        ▶ wing:           {…}
        ▶ servo:          {…}
      ▶ topSegment:       {…}
▼ fuselageWithComponents:
    node_type:            "SingleBatteryFuselageWithComponents"
  ▶ battery:              {…}
  ▶ fuselage:             {…}
```

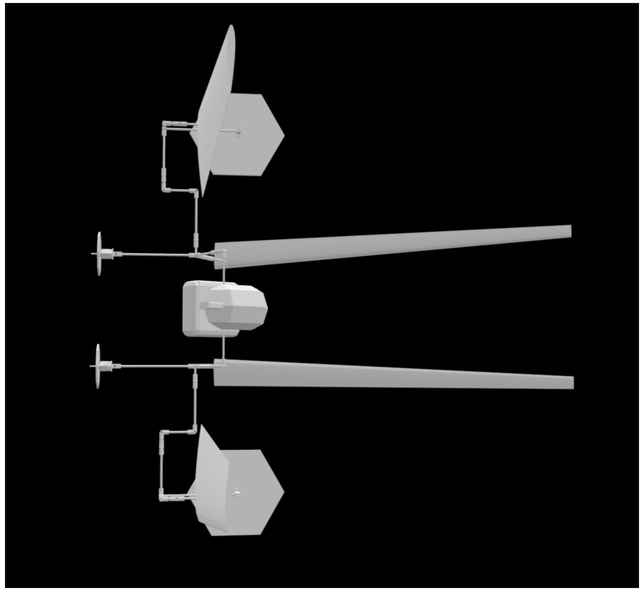





Symbolic Generator Using CRAIDL - a Declarative Probabilistic Programming Language

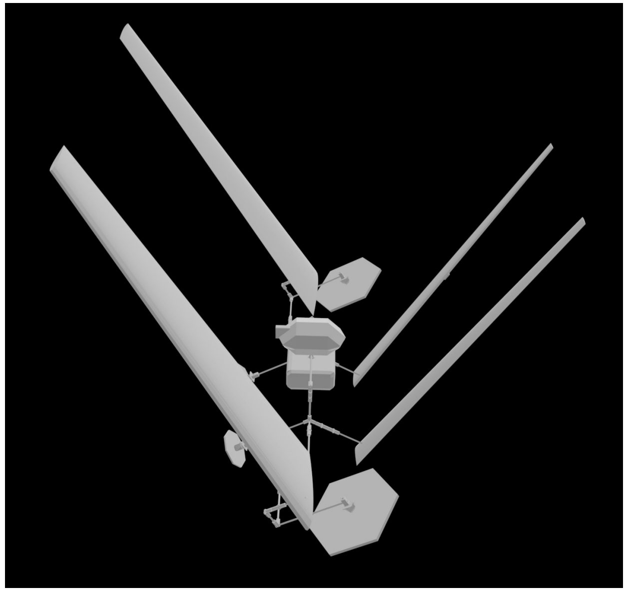
```
▼ mainSegment:
    node_type:              "BranchWithTopSegment_Asym"
    armLength:              145.4677982914925
    angle:                  120
  ▶ leftSegment:            {…}
  ▶ rightSegment:           {…}
  ▼ topSegment:
      node_type:            "DoubleBendSegment"
      angle:                -90
      arm1Length:           213.9896273386132
      arm2Length:           132.35846236720778
    ▼ mainSegment:
        node_type:          "SidewaysBendWithTopSegment"
        angle:              -90
        armLength:          199.99986531517303
      ▼ mainSegment:
          node_type:        "AngledWingArm"
          arm1Length:       89.06951101606187
          arm2Length:       125.9417328495726
        ▶ wing:             {…}
        ▼ servo:
            servoType:      "Hitec_HS_5087MH"
      ▼ topSegment:
          node_type:        "PropArm"
          armLength:        350.27592082572664
        ▶ motor:            {…}
        ▶ prop:             {…}
        ▶ flange:           {…}
```

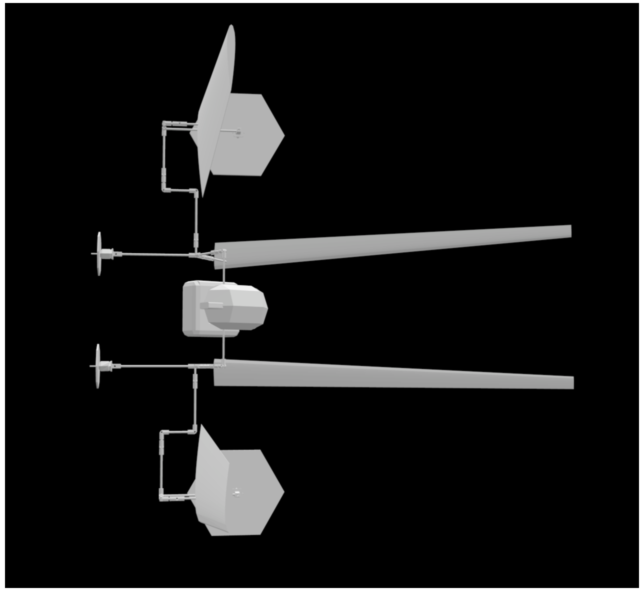




Symbolic Generator Using CRAIDL - a Declarative Probabilistic Programming Language

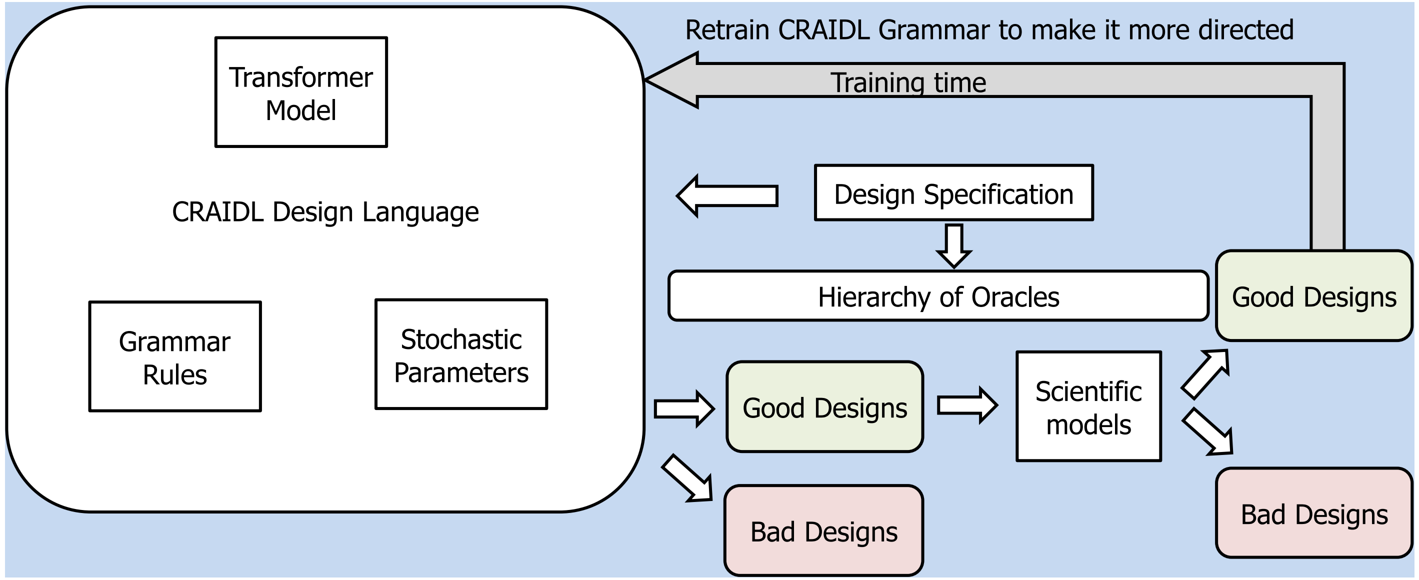

```
▼ mainSegment:
    node_type:        "AngledWingArm"
    arm1Length:       89.06951101606187
    arm2Length:       125.9417328495726
  ▼ wing:
      wingType:       "Wing_horiz_hole"
      nacaProfile:    "4312"
      span:           1868.9591009651886
      chordInner:     447.59088752358923
      chordOuter:     190.49774585767557
      taperOffset:    false
      aileronBias:    0.46673040537379185
      flapBias:       0.6267633918626601
      load:           393.5496771272593
      tubeOffset:     70.66926820238568
      tubeRot:        270
  ▼ servo:
      servoType:      "Hitec_HS_5087MH"
▼ topSegment:
    node_type:        "PropArm"
    armLength:        350.27592082572664
  ▼ motor:
      motorType:      "t_motor_AT4120KV560"
  ▼ prop:
      propType:       "apc_propellers_18x8"
```



Minimizing Oracle Calls Using Surrogates

Intertwined generation of data and design optimization



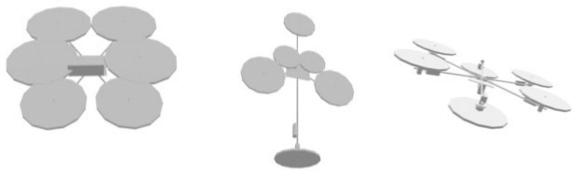
Distribution Statement A. Approved for Public Release, Distribution Unlimited

Iterative Feedback to Data Generator from Exploration

NS Generator: Stochastic Grammar for CRAIDL design language

- Grammar Rules
- Stochastic Parameters

Example Designs



Oracles for evaluating design

- Design Specification
- Scientific models

Good Designs

Bad Designs

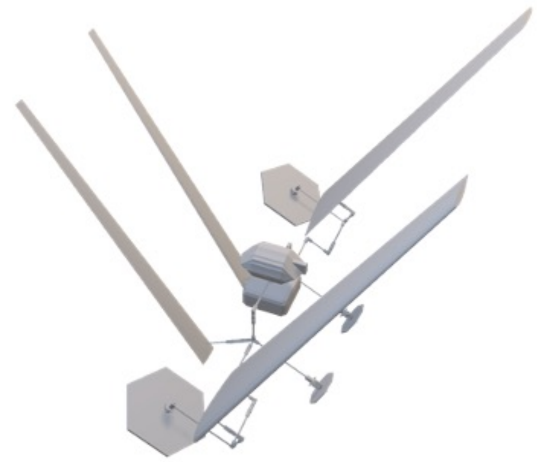
1. **Need to make data generator objective-aware** – Only a small percentage of designs meet the specification when the initial stochastic grammar. E.g. SwRI pipeline – out of 5000 designs, only 70 were valid. Direct tuning can lead to collapse of diversity.

2. **Need to make evaluation faster** - Scientific models can be slow to evaluate.

3. **Need more valid designs for learning of rules and parameters** – Feed valid designs back to learn the generator to improve valid generation but without losing diversity.

Distribution Statement A. Approved for Public Release, Distribution Unlimited

System-level Surrogate Model: Inputs

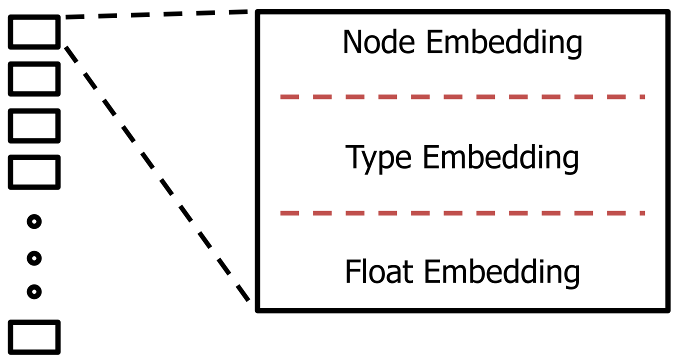


1. **3D STL CAD Model as a Point Cloud**

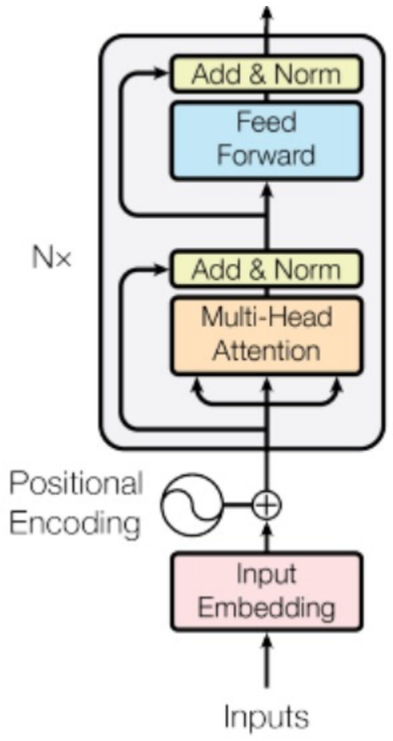
2. **Symbolic Design turned into a Sequence**

```
[{'node_type': 'ConnectedHub3_2_1'},
 {'angle': 'Float'},
 {'node_type': 'AngledPropArm'},
 {'armLength': 'Float'},
 {'motorType': 't_motor_MN4010KV370'},
 {'propType': 'apc_propellers_12x8'},
 {'escType': 't_motor_FLAME_100A'},
 {'offset': 'Float'},
 {'node_type': 'PropArm'},
 {'armLength': 'Float'},
 {'motorType': 't_motor_AT3520KV720'},
 {'propType': 'apc_propellers_15x10'},
 {'escType': 't_motor_T_80A'},
 {'offset': 'Float'},
 {'offset': 'Float'},
 {'angle': 'Float'},
 {'x1_offset': 'Float'},
 {'z1_offset': 'Float'},
 {'batteryType': 'TurnigyGraphene1300mAh4S75C'},
 {'End': 'End'}]
```

Design Sequence



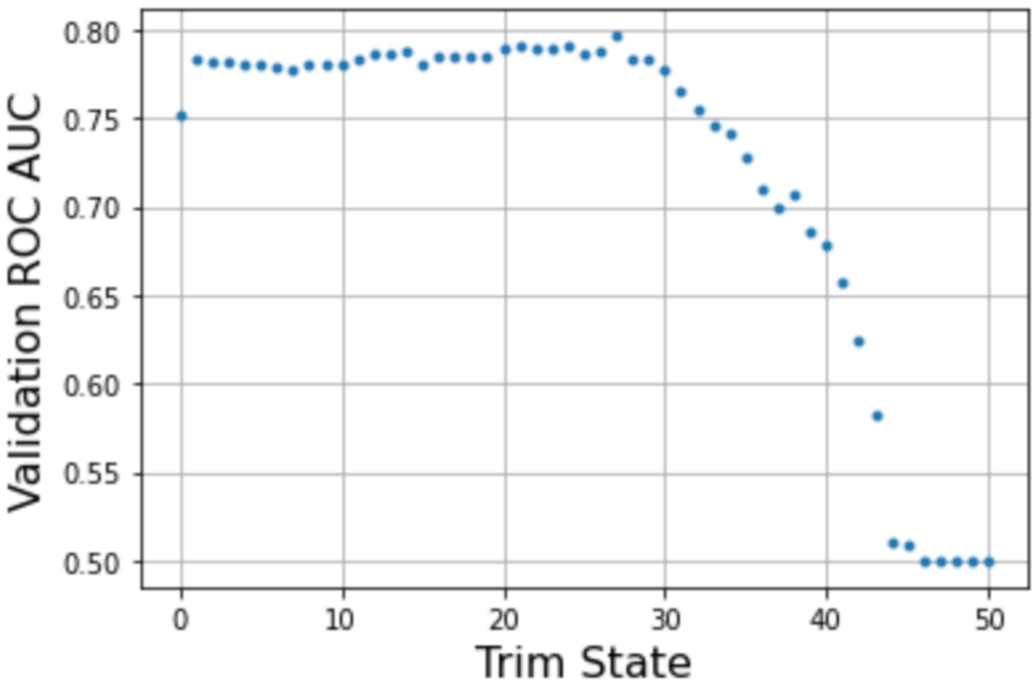
Single Token



Distribution Statement A. Approved for Public Release, Distribution Unlimited

System-level Surrogate Model: Outputs

Approach	Data	Hover (Acc. ↑), (F1 Score ↑)	Mass (MSE ↓)	Flight Distance (MSE ↓)	Interference (F1 Score ↑)
Transformer Enc.	Sequence	0.8551, 0.7605	0.1319	0.5141	0.7706
LSTM	Sequence	0.7931, 0.6836	0.2576	0.7617	0.0
Graph conv. net	Point Cloud	0.8585, 0.7660	0.170	0.582	0.8207



Distribution Statement A. Approved for Public Release, Distribution Unlimited

Generative Models for Design Topology

The transformer model saved 137.2 days of compute time

CRAIDL design language

Grammar Rules

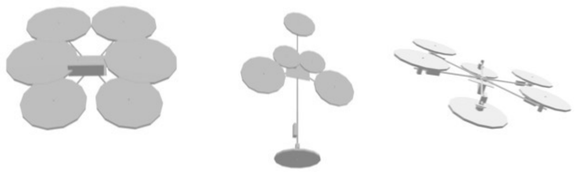
Stochastic Parameters

Automate retraining of Stochastic Grammar to make it more directed
Enables successful learning design patterns

Design Specification
(Ability to hover)

Hierarchy of Oracles

Example Designs



100,000 ~ order of minutes

ML models

Symbolic rules

21,000

Good Designs

Bad Designs

79,000

Scientific models

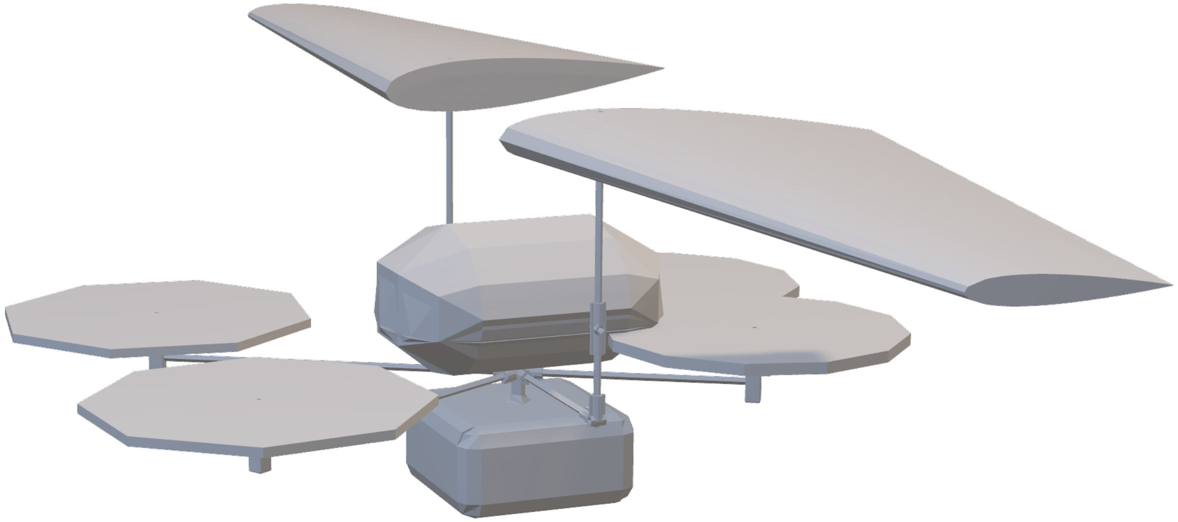
36.5 days compute

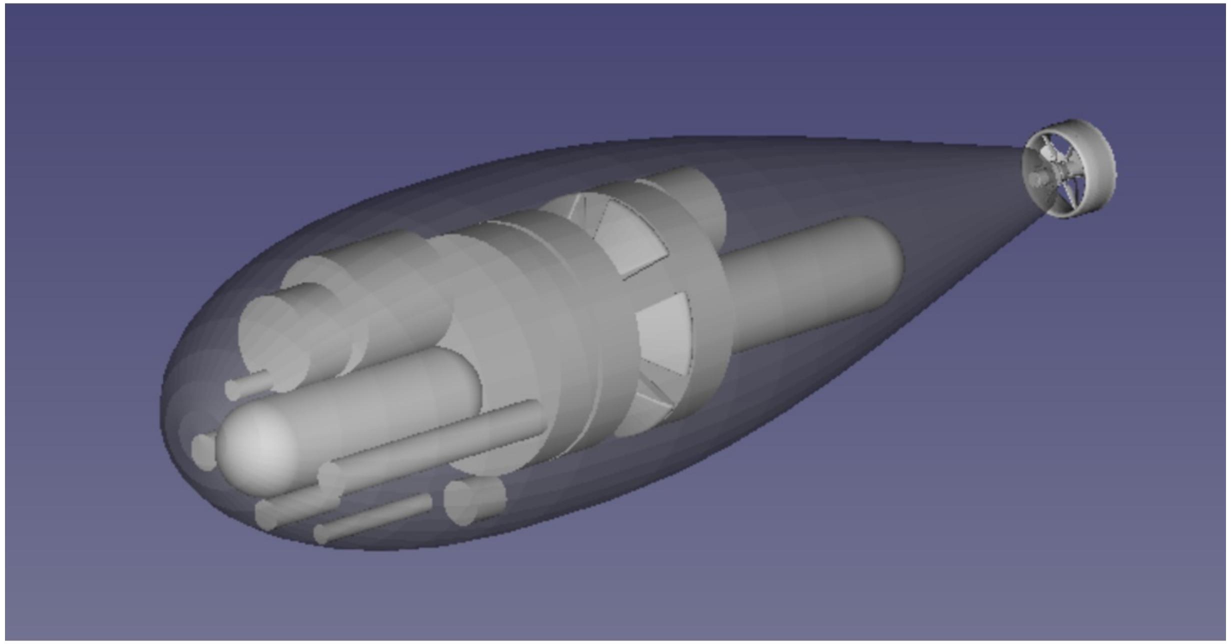
Good Designs

Bad Designs

Distribution Statement A. Approved for Public Release, Distribution Unlimited

Applications

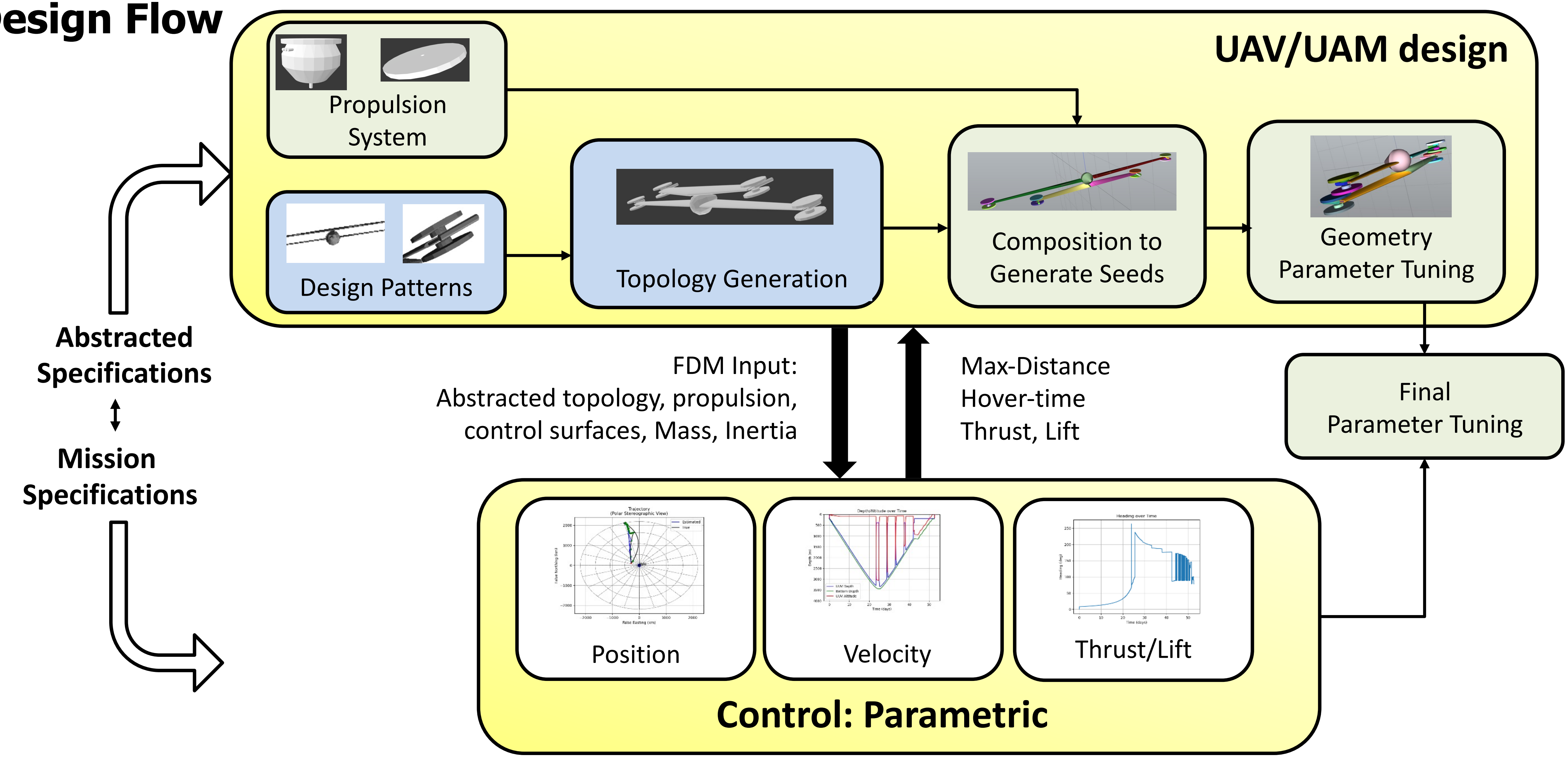




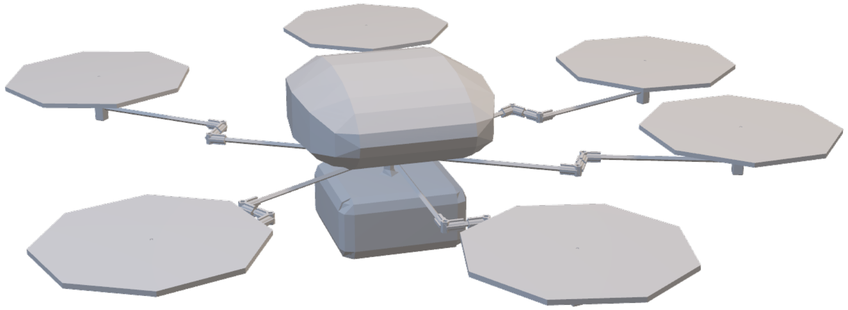
Distribution Statement A. Approved for Public Release, Distribution Unlimited

Air Corpus: UAV/UAM

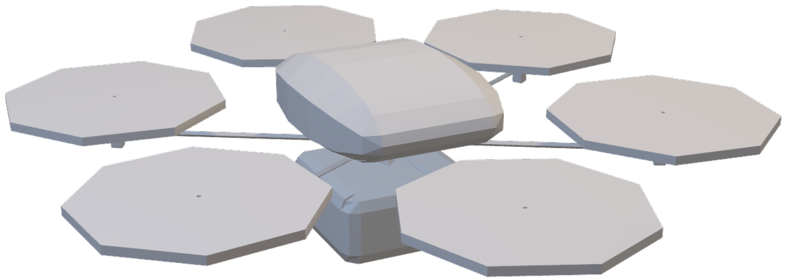
Design Flow



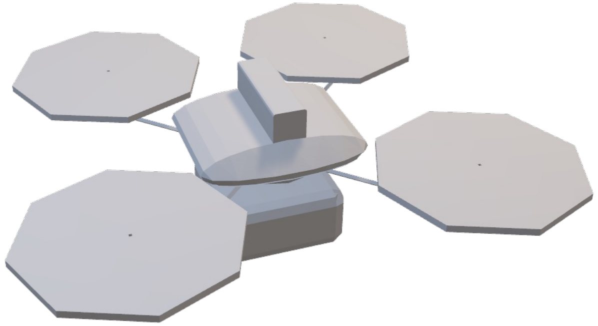
5 High Performing Designs



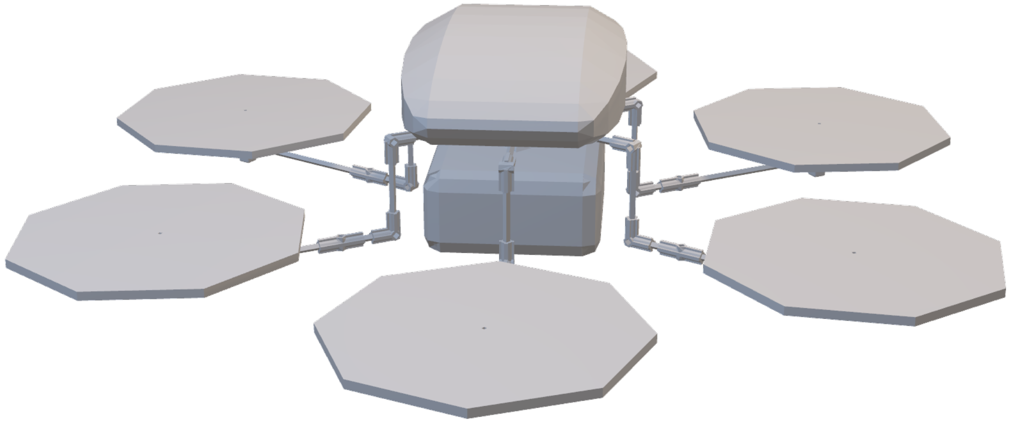
Beetle, Score: 7962



Daisy, Score: 5218



Frog, Score: 7920

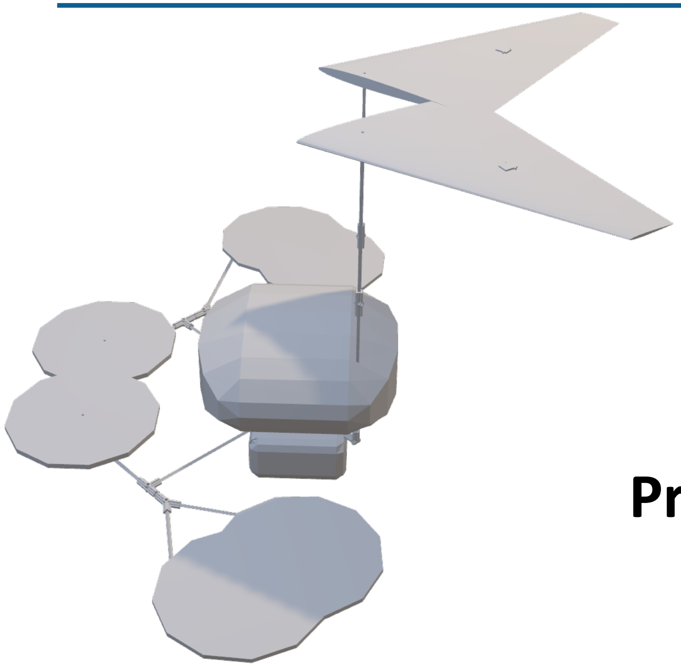


Frog, Score: 5263

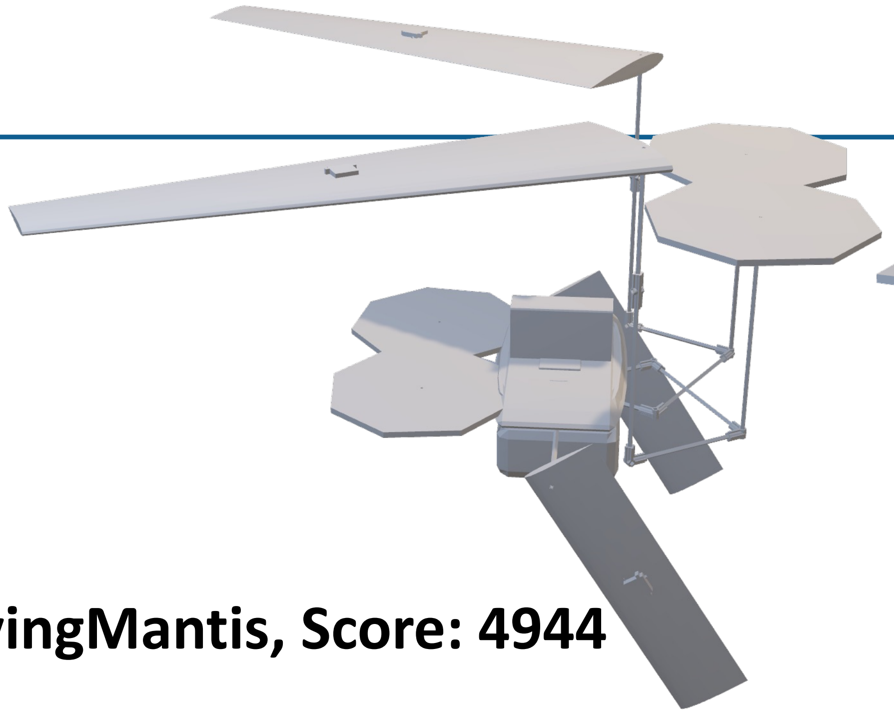
Water strider, Score: 5218

Distribution Statement A. Approved for Public Release, Distribution Unlimited

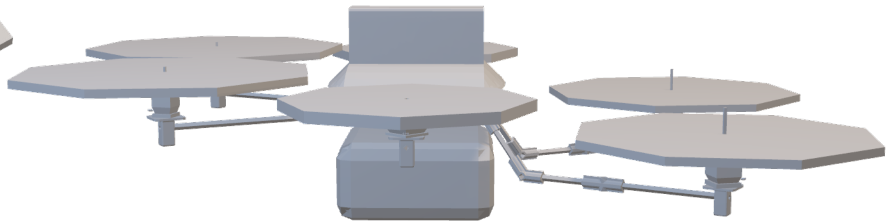
5 Top Creative Designs



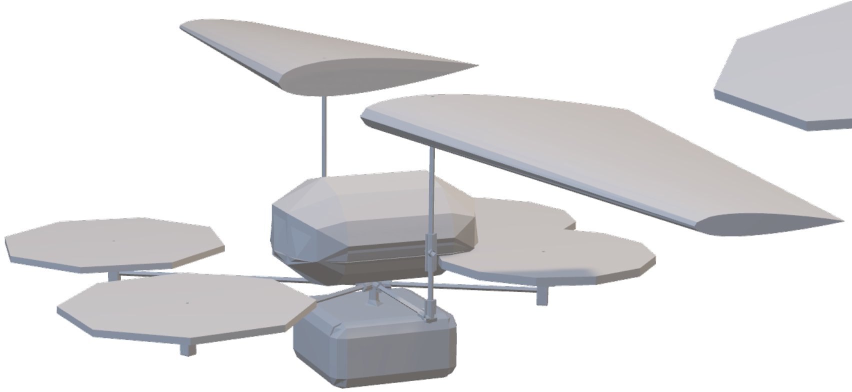
Kite, Score: 4308



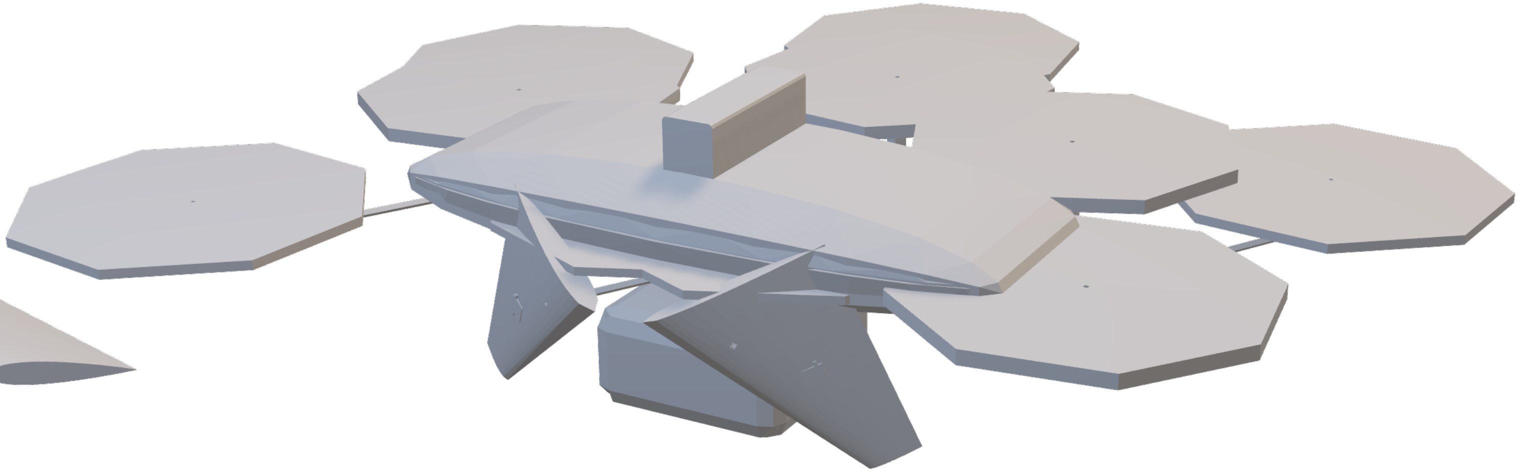
PrayingMantis, Score: 4944



Frog, Score: 3550



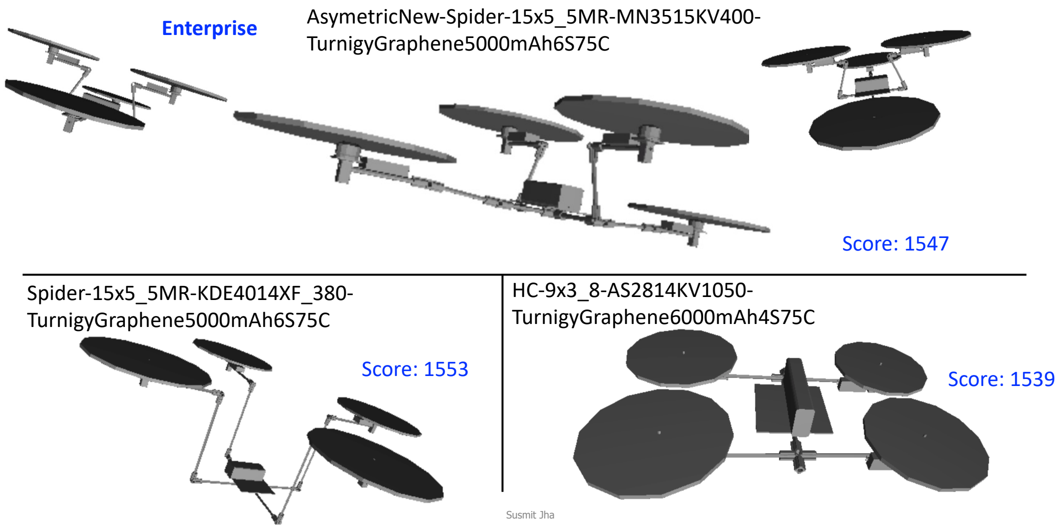
Vulture, Score: 5263

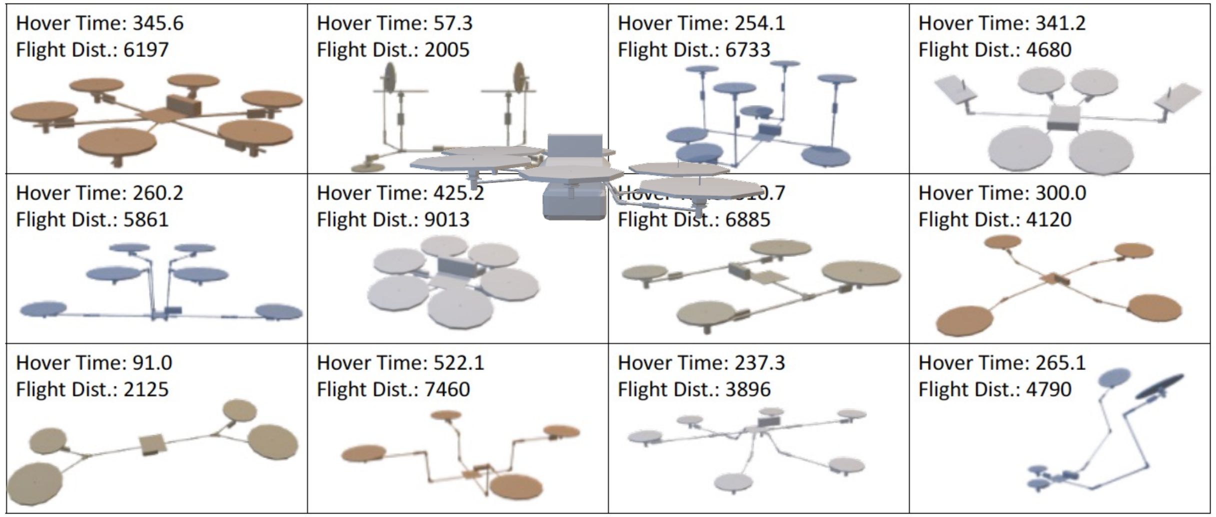


Walrus, Score: 5306

Distribution Statement

UAV Designs over the course of DARPA SDCPS Program (2020-2023)





Fall 2021

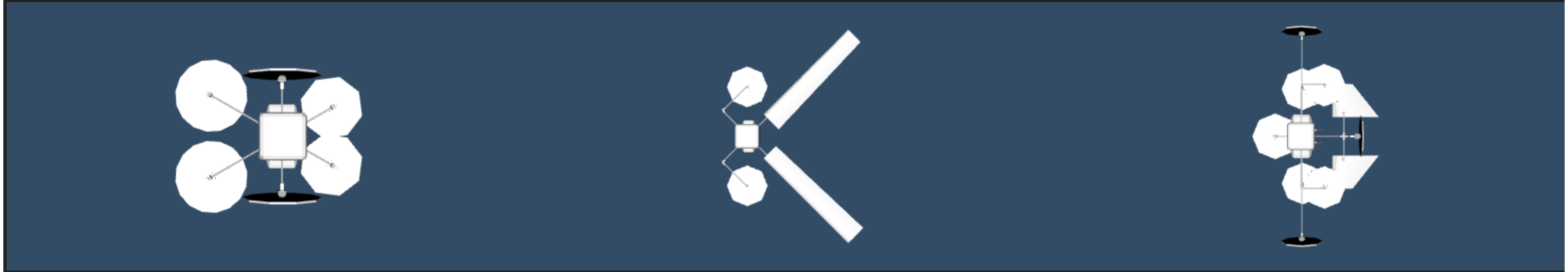
Spring 2022

Fall 2022

Aircraft Design Dataset: https://aircraftverse.onrender.com/

AircraftVerse

We present AircraftVerse, a publicly available dataset with over 27,000 diverse set of air vehicle designs.



Use the mouse to manipulate the Aircraft designs (STL Files) in the figures above.

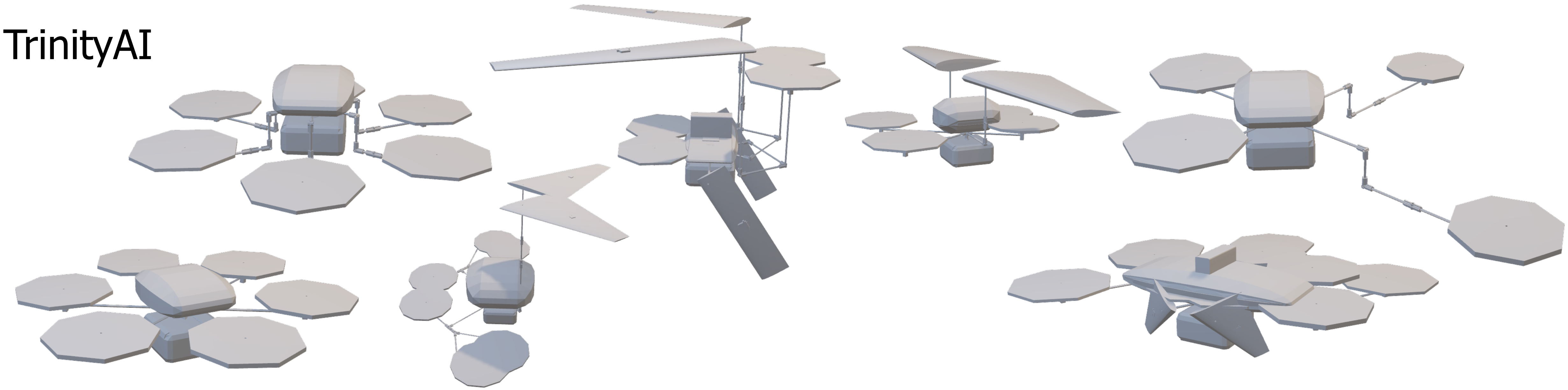
Aircraft design encompasses different physics domains and, hence, multiple modalities of representation. The evaluation of these designs requires the use of scientific analytical and simulation models ranging from computer-aided design tools for structural and manufacturing analysis, computational fluid dynamics tools for drag and lift computation, battery models for energy estimation, and simulation models for flight control and dynamics. AircraftVerse contains over 27,000 diverse set of air vehicle designs - the largest corpus of designs with this level of complexity. Each design comprises the following artifacts: a symbolic design tree describing topology, propulsion subsystem, battery subsystem, and other design details; a STandard for the Exchange of Product (STEP) model data; a 3D CAD design using Standard Tessellation Languages (STL); a 3D point cloud for the shape of the design; and evaluation results from high fidelity state-of-the-art physics models that characterize performance metrics such as maximum flight distance and hover-time. We also present baseline surrogate models that use different modalities of design representation to predict design performance metrics, which we provide as part of our dataset release.

Dataset | Paper | Explore Designs

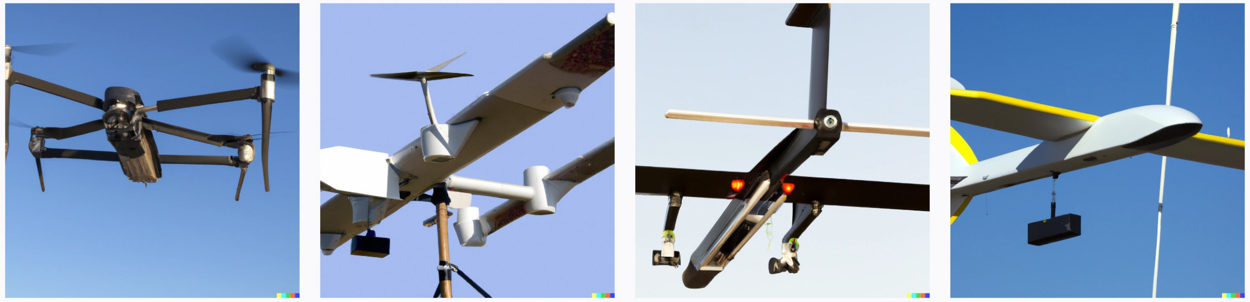
Aircraft Design Dataset: https://aircraftverse.onrender.com/

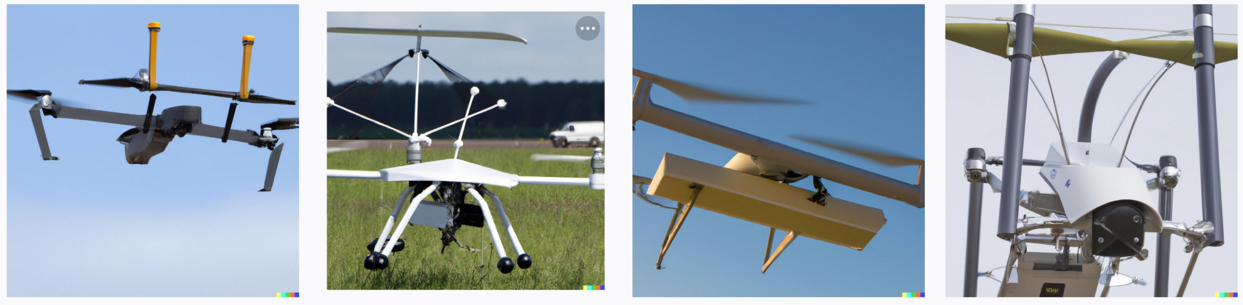
TrinityAI produced AircraftVerse Designs vs DALL-E-2

TrinityAI



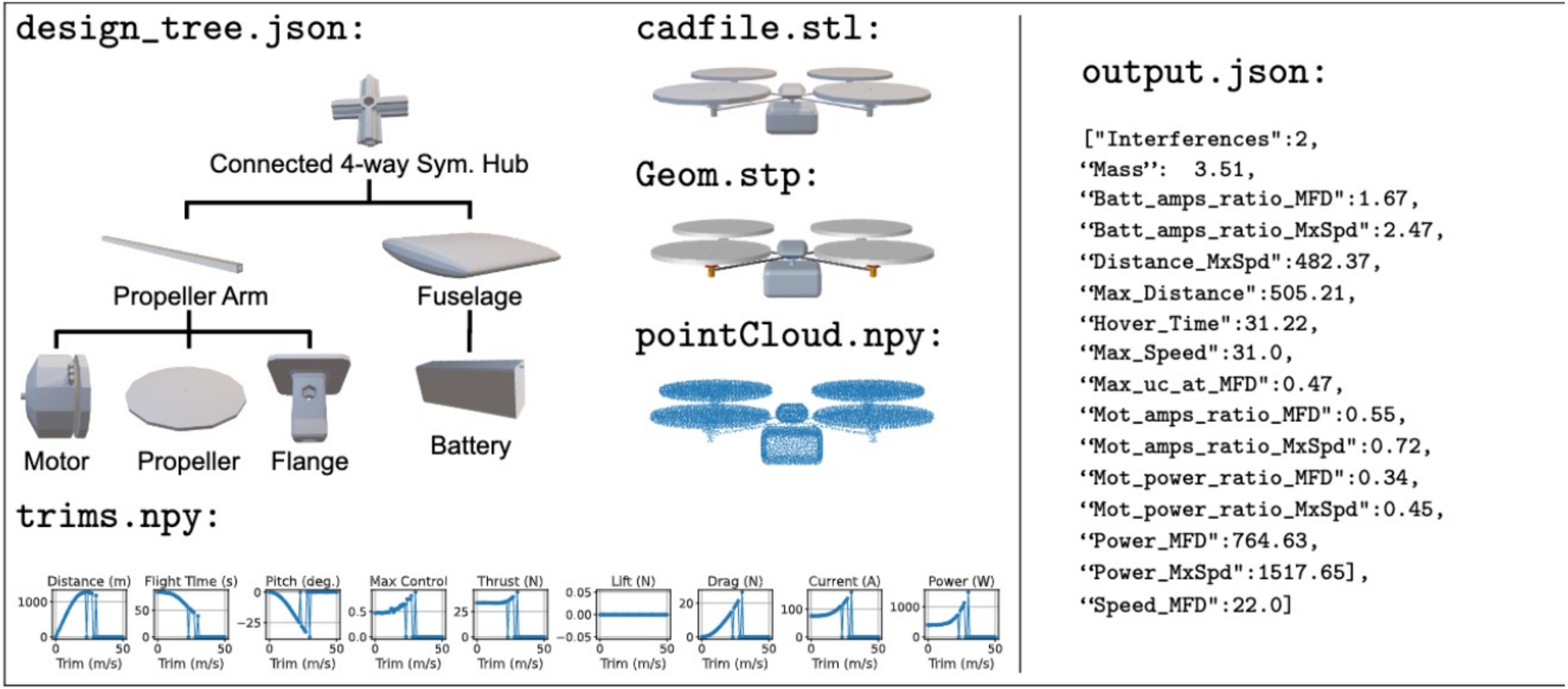
DALL-E-2





Distribution Statement A. Approved for Public Release, Distribution Unlimited

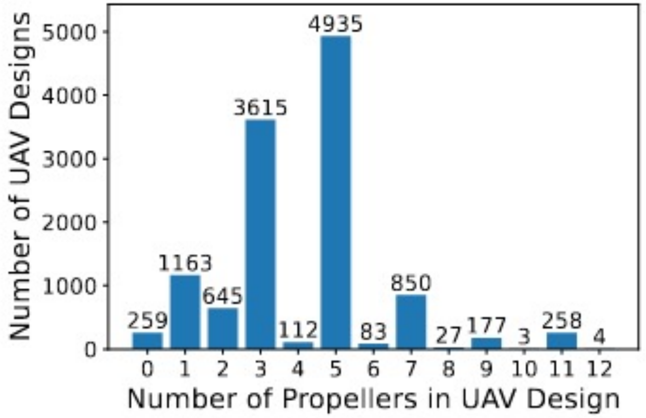
Aircraft Design Dataset: https://aircraftverse.onrender.com/



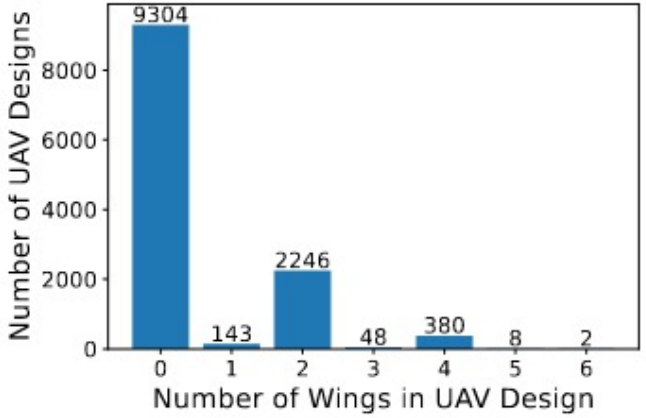
Aircraft Design Dataset Code at Github: https://github.com/SRI-CSL/AircraftVerse/

- notebooks contains:
  - `DataDemo.ipynb`: This notebook demonstrates how to read the raw data in the dataset.
  - `DataSetPlot.ipynb`: This notebook contains the summary plots of the full data set.
  - `ModelBenchmark.ipynb`: This notebook contains the benchmark experiments for the sequence data. To run this notebook you may need to build the data set according to the encoding of Cobb et al. 2022 (https://arxiv.org/pdf/2211.08138.pdf). This code is provided in `build_transformer_data.py`.
- data contains:
  - 15 illustrative designs taken from AircraftVerse (http://doi.org/10.5281/zenodo.6525446).
  - The corpus dictionary.
  - A example data set build by `build_transformer_data.py`.
- code contains:
  - `build_transformer_data.py`: Apply this file to the folder structure as in the zip file available on http://doi.org/10.5281/zenodo.6525446 to build the dataset as we did in the paper.
  - `ssm.py`: File containing torch models and related helper functions.
  - `util.py`: Some useful plotting functions for the notebooks.

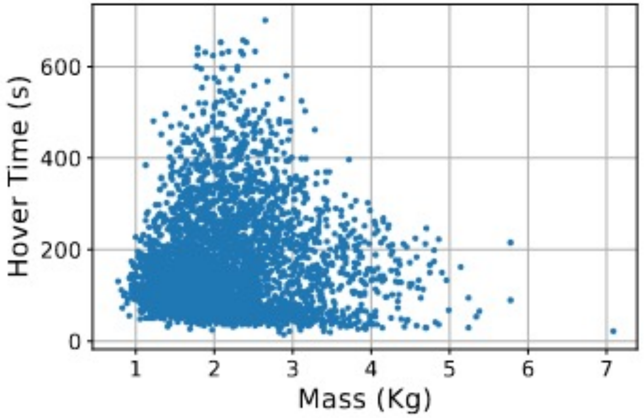
Diversity Summary Statistics



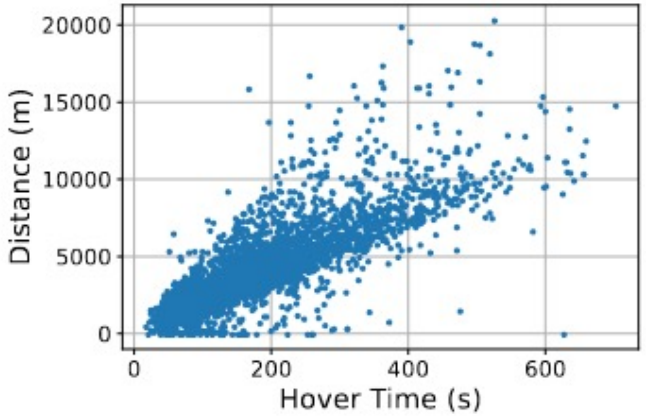
(a) Number of Propellers



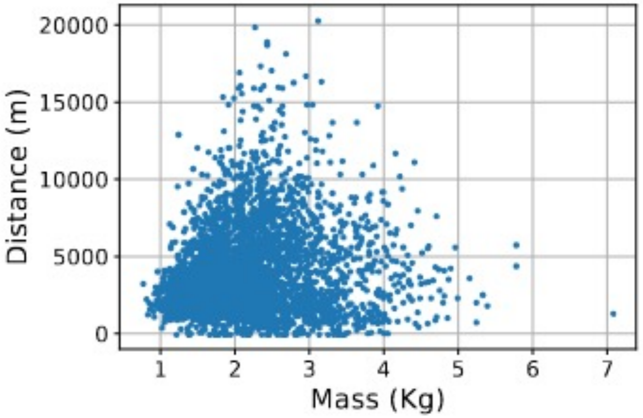
(b) Number of Wings



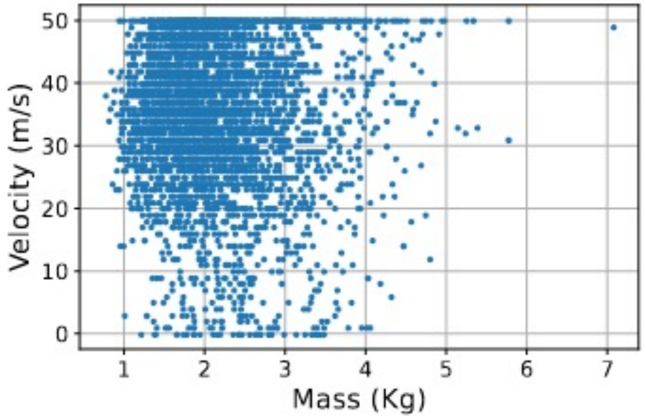
(c) Hover Time vs. Mass



(d) Distance vs. Hover Time



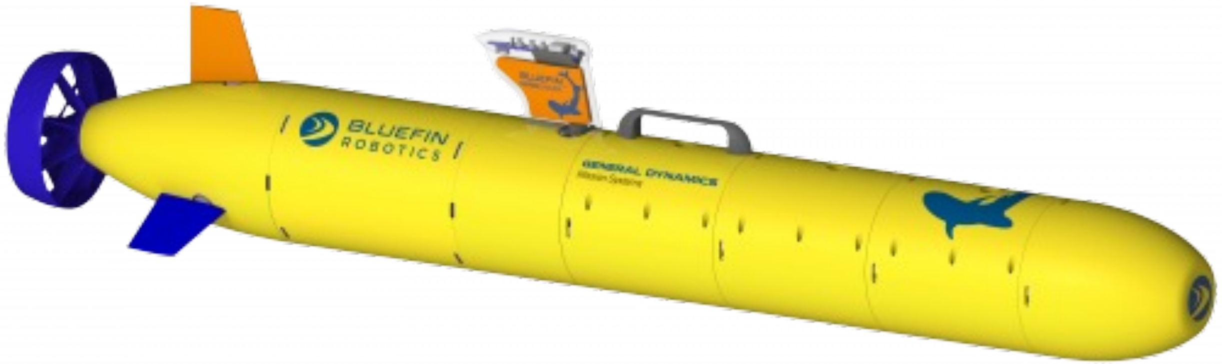
(e) Distance vs. Mass



(f) Max. Velocity vs. Mass

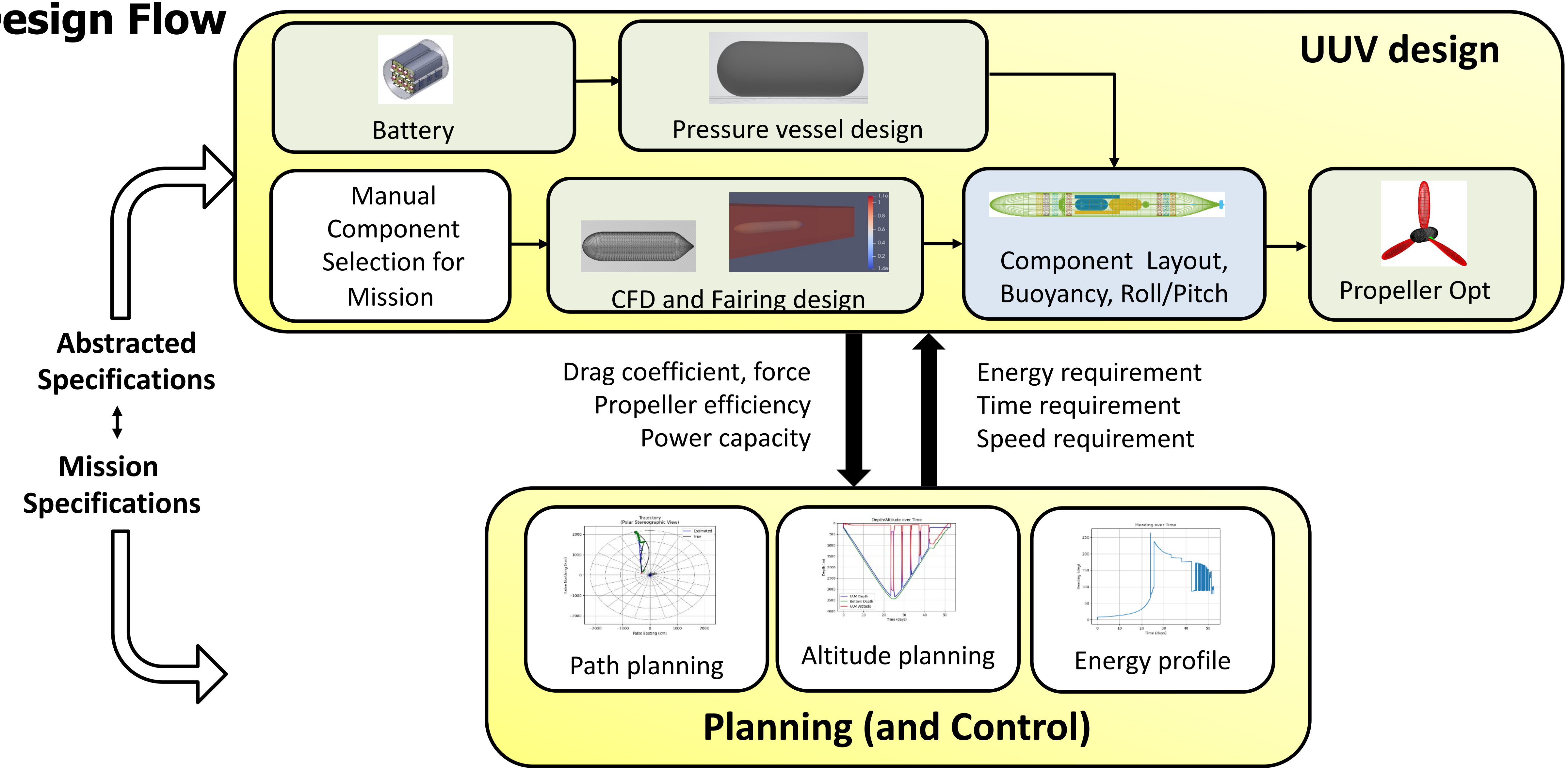
Distribution Statement A. Approved for Public Release, Distribution Unlimited

Underwater Domain: UUV



Underwater Domain: UUV

Design Flow



UUV Design

Total distance travelled: **4257km**

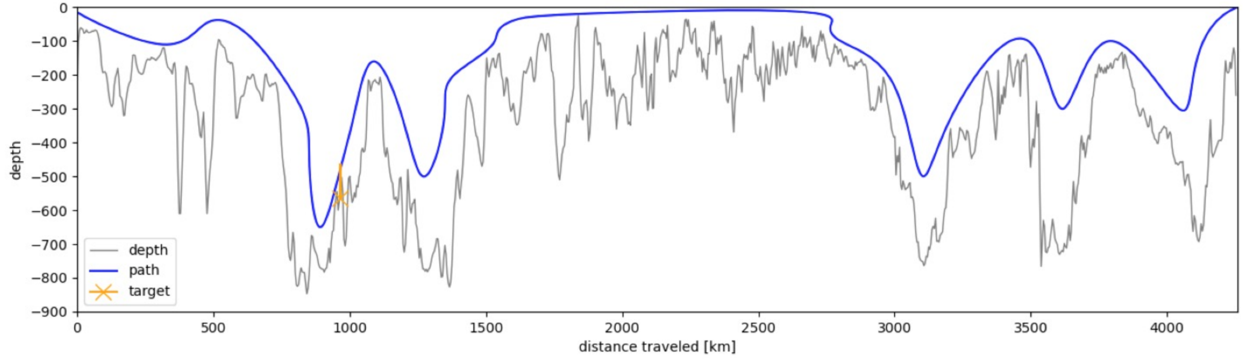
Minimum distance to shore: **4810m from (80.74, -79.36)**

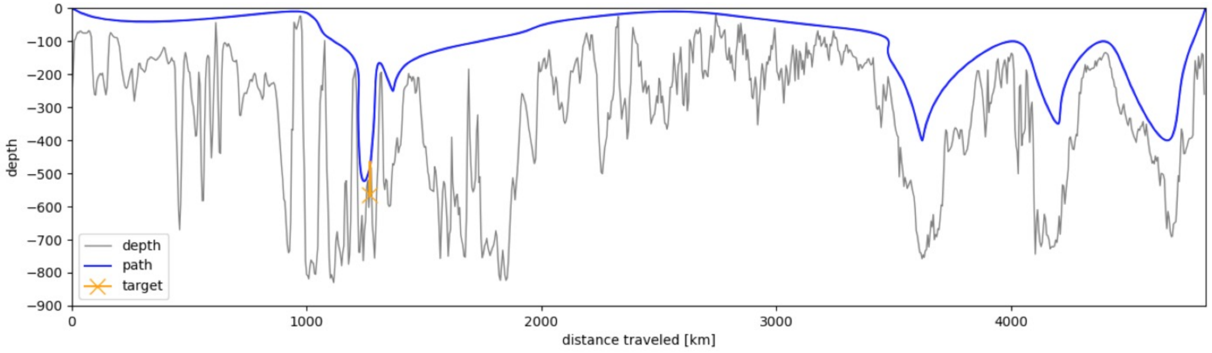
Minimum sensing distance: **7.5km**

Total distance travelled: **4827.5km**

Minimum distance to shore: **1445m from (81.34, -89.94)**

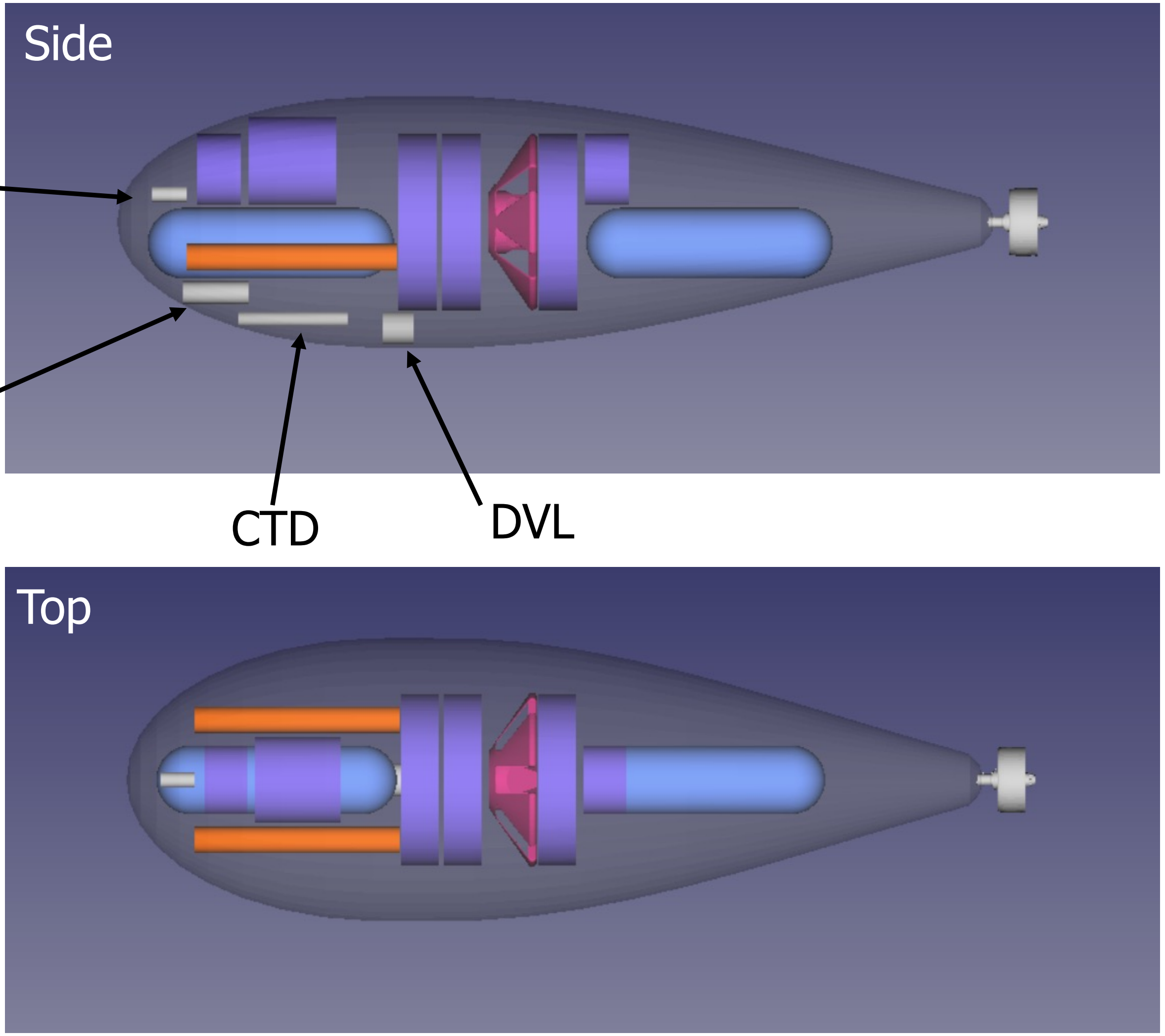
Robust Plan and Controller





Distribution Statement

UUV Design: Physical Design



CAD Model

Names are color coded with the color of the rendering.

PV

Foam

OBS

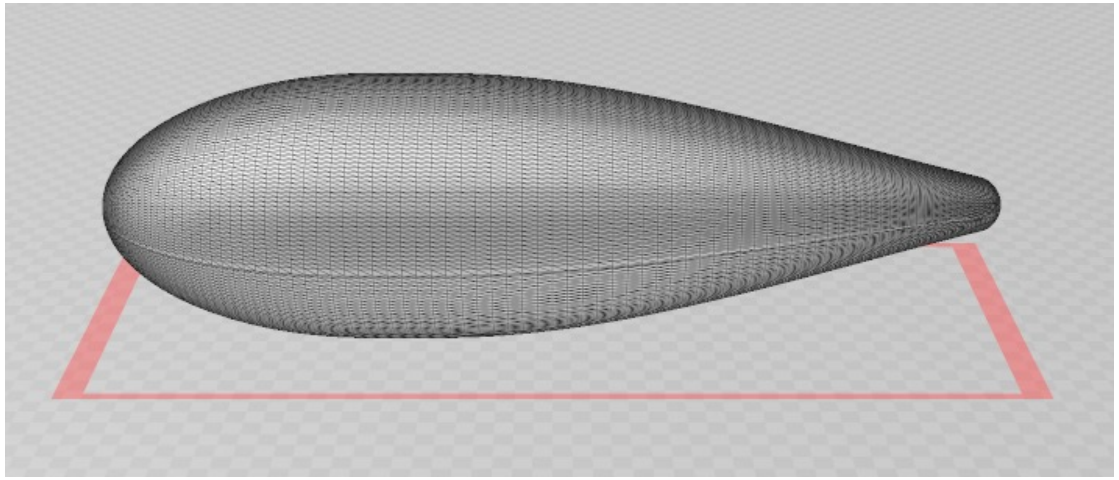
Pitch shifter

Key Vehicle Attributes	
Length (m)	4.99m
Max diameter (m)	1.65m
Mass (kg)	1357.1 Kg
CoB-CoG offset (m)	0.050
Hotel power (W)	19
Max vehicle speed (m/s)	1.1

Distribution Statement A. Approved for Public Release, Distribution Unlimited
Distribution Statement

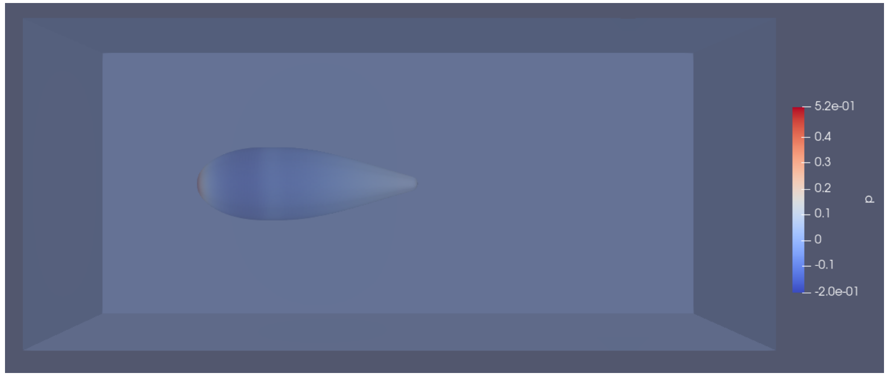
UUV Design: Faring

Length	4.990 m
Diameter	1.650 m
Nose length	1.5 m
Body length	0.378 m
Tail length	3.122 m

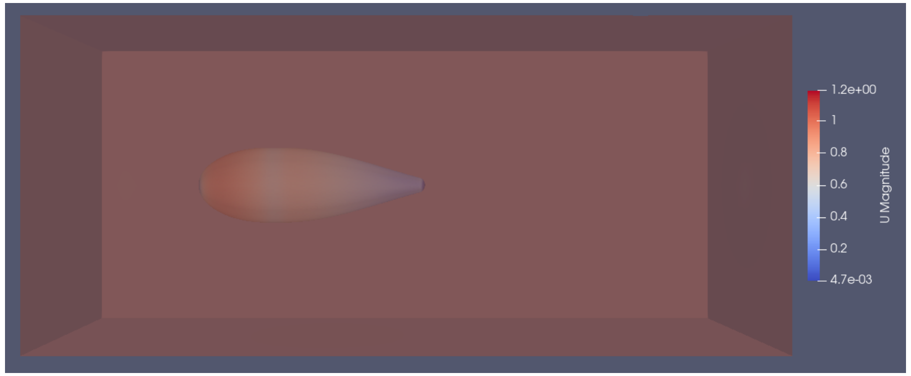


Teardrop fairing: lower drag, better packing

(Drag coeff/ Drag force)	velocity = 1	velocity = 1.1
aoa = 0	0.049 / 46.67	0.0485 / 55.78
aoa = 15	0.083 / 79.05	0.082/ 94.63



Pressure field



Velocity field

Fairing

Main observations on fairing choice.

- At the relatively low velocity (1-2 meter/sec) required for the mission, viscous drag dominate the pressure drag. Viscous drag contributes to approx. 67% of the overall drag force and pressure drag contribute to approx. 33% of the force.
- Pressure drag primarily depends on the frontal cross-section area of the vehicle and viscous drag primarily depends on the surface area.
- Drag force at various angles of attacks depends on the cross-section area along the direction

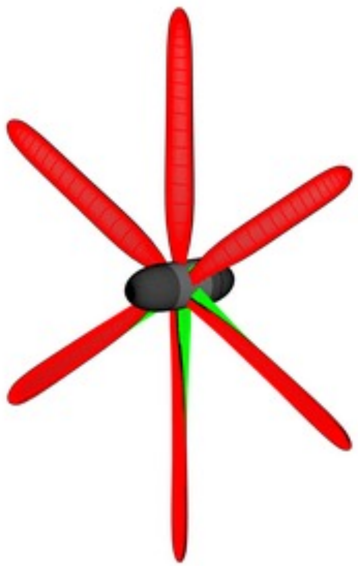
- We consider a fairing with a large diameter and a short length to reduce overall drag force keeping the volume fixed.
- Better drag properties at various angles of attack compared to torpedo while the volume is kept fixed.

Propulsion System

- We use a custom-designed propeller using Open Prop. We do not design motor or gearing, and use corpus components (Tecnadyne Model 2061) for these.
- Overall efficiency : 0.5

Propeller properties:

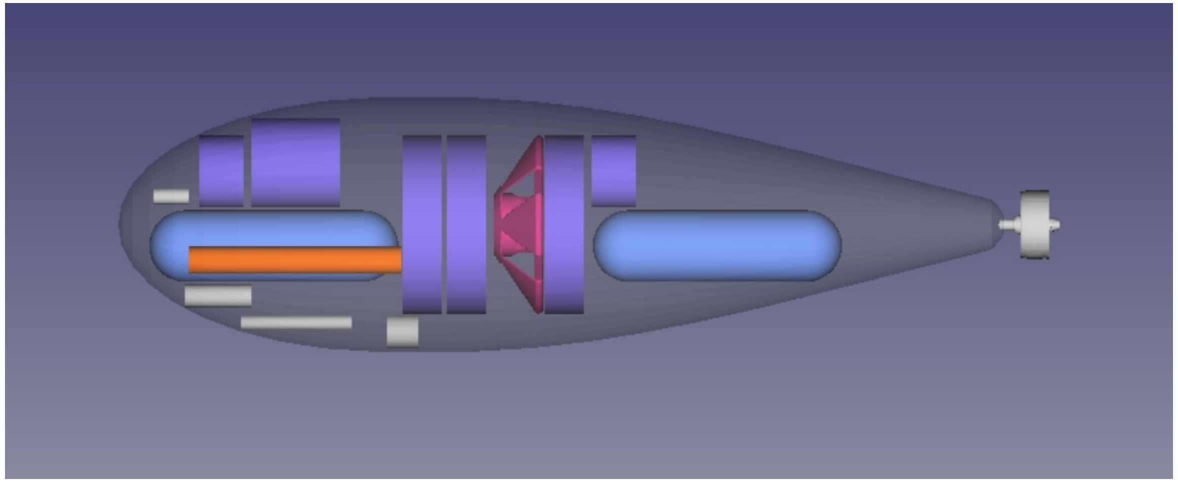
- Velocity = 1 – 1.5 m/s
- Water density = 1028 kg / m^3
- Number of blades = 6
- Diameter: 0.7 m
- Hub diameter: 0.055 m
- Propeller Efficiency at the speed 1 m/s: 70%
- Propeller Efficiency at the speed 1.5 m/s: 80%



Propeller rendering

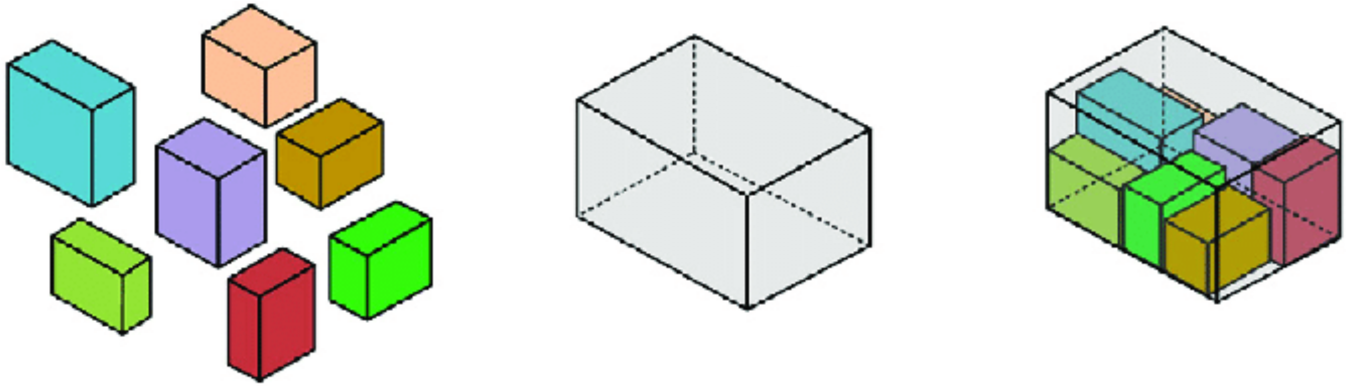
Packing and CB-CG / Pitch Analysis

- CB-CG separation – 50mm vertical separation



- Pitch range is –34 to +34 degrees
  - Achieved by two mass shifters, each has
  - 1.2m long Al. tube with 59 kg movable lead mass

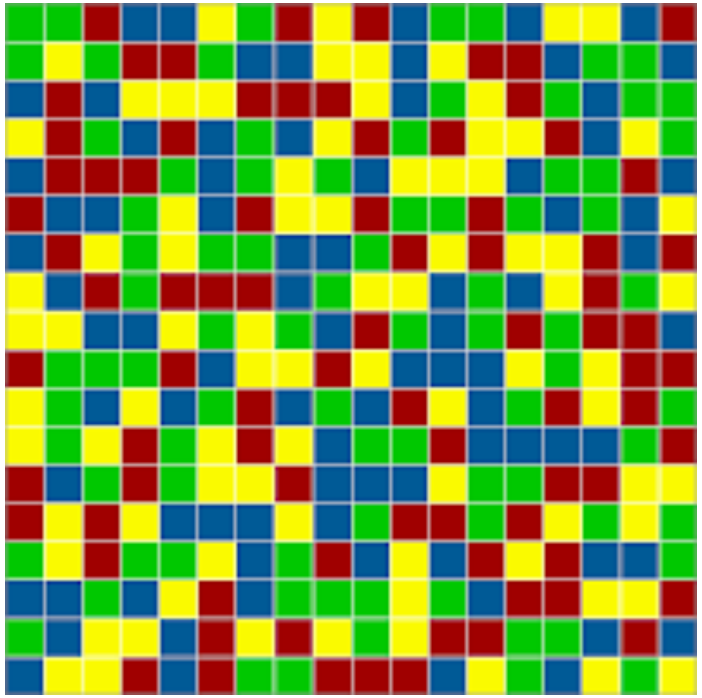
- OBS payload is packaged in a neutrally buoyant unit with two compensating syntactic foam discs, leaving CG/CB unaffected by payload delivery

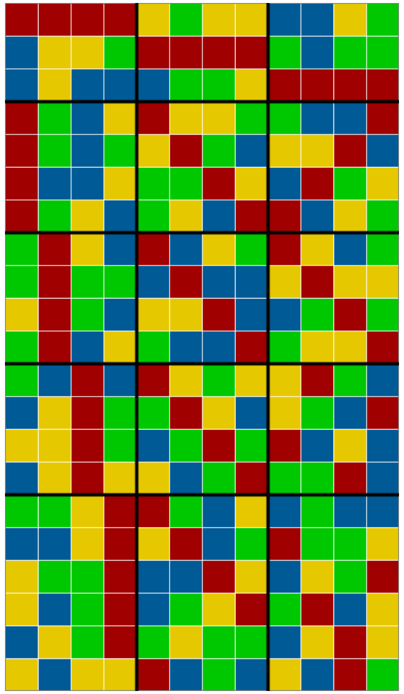


Combinatorial Search

If $n \in \mathsf{N}$ then $[n] = \{1, \ldots, n\}$. If $n, m \in \mathsf{N}$ then $G_{n,m}$ is the grid $[n] \times [m]$.

A *rectangle* of $G_{n,m}$ is a subset of the form $\{(a,b), (a+c_1, b), (a+c_1, b+c_2), (a, b+c_2)\}$ for some $a, b, c_1, c_2 \in \mathsf{N}$. A grid $G_{n,m}$ is *c-colorable* if there is a function $\chi : G_{n,m} \to [c]$ such that there are no rectangles with all four corners the same color.





Publications

1. Edmond Cunningham, Adam Cobb, Susmit Jha. **Principal Manifold Flows.** 39th International Conference on Machine Learning (ICML), 2022.
2. Chenxi Yang, Swarat Chaudhuri (2022). **Safe Neurosymbolic Learning with Differentiable Symbolic Execution**. International Conference on Learning Representations (ICLR), 2022
3. Cobb, Adam; Roy, Anirban; Elenius, Daniel; Koneripalli, Kaushik; Jha, Susmit. **On Diverse System-Level Design Using Manifold Learning and Partial Simulated Annealing**. 17th International Design Conference (DESIGN), 2022. **Identified as Top 10% of papers at the venue.**
4. Cobb, Adam; Roy, Anirban; Elenius, Daniel; Jha, Susmit. **Trinity AI Co-Designer.** Design Automation for CPS and IoT (DESTION 2022).
5. Agrawal, A., & McComb, C. **Comparison of Visualization Strategies for High-Dimensional Exploration Behavior of CPS Design Agents**. Design Automation for CPS and IoT (DESTION 2022).
6. Agrawal, A., & McComb, C (2022). Reinforcement learning for efficient design space exploration with variable fidelity analysis models. Journal of Computers and Information Science in Engineering, 2022.
7. Li, M., & McComb, C. **Using Physics-Informed Generative Adversarial Networks to Perform Super-Resolution for Multiphase Fluid Simulations**. Journal of Computers and Information Science in Engineering, 2021
8. Samuel Kessler, Adam Cobb, Stefan Zohren, Stephen J. Roberts. **Can Sequential Bayesian Inference Solve Continual Learning?** Advances in Approximate Bayesian Inference (ABI), 2021

Thank you!

Swap two bitvectors

Given x and y as two bitvectors, swap them without using any temporary variables.

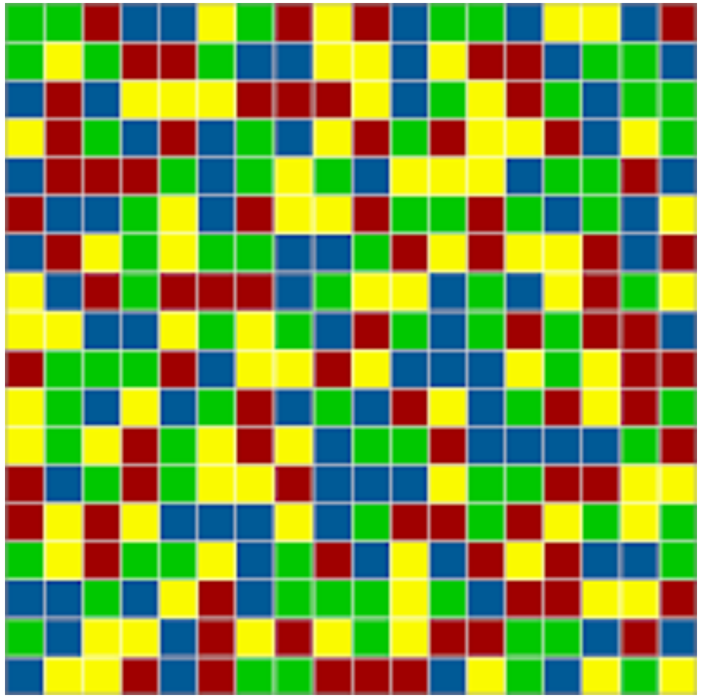
Use at most 3 lines in the program

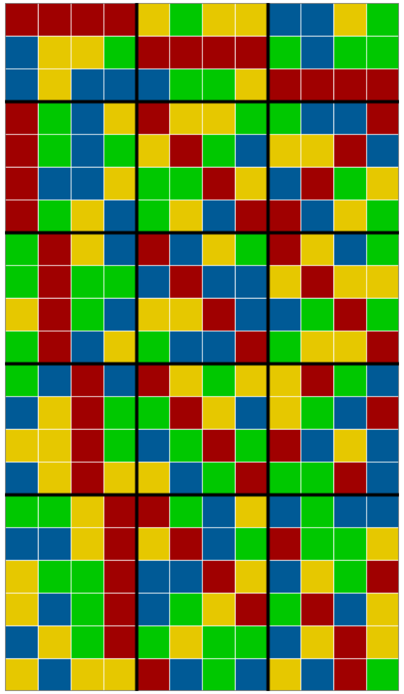
Use xor bitwise operators

Combinatorial Search

If $n \in \mathsf{N}$ then $[n] = \{1, \ldots, n\}$. If $n, m \in \mathsf{N}$ then $G_{n,m}$ is the grid $[n] \times [m]$.

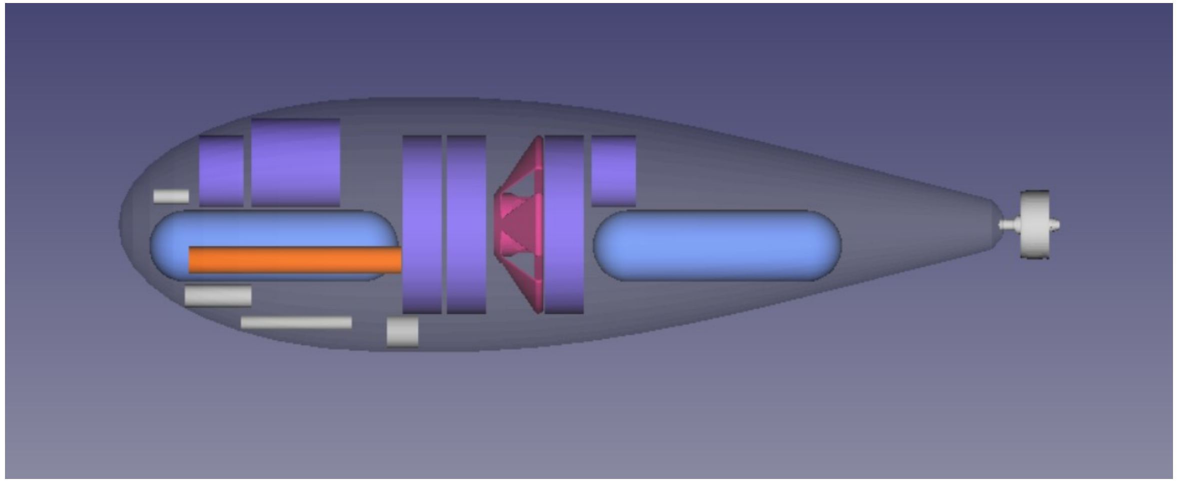
A *rectangle* of $G_{n,m}$ is a subset of the form $\{(a,b), (a+c_1,b), (a+c_1,b+c_2), (a,b+c_2)\}$ for some $a, b, c_1, c_2 \in \mathsf{N}$. A grid $G_{n,m}$ is *c-colorable* if there is a function $\chi : G_{n,m} \to [c]$ such that there are no rectangles with all four corners the same color.





Packing and CB-CG / Pitch Analysis

- CB-CG separation – 50mm vertical separation



- Pitch range is –34 to +34 degrees
  - Achieved by two mass shifters, each has
  - 1.2m long Al. tube with 59 kg movable lead mass

- OBS payload is packaged in a neutrally buoyant unit with two compensating syntactic foam discs, leaving CG/CB unaffected by payload delivery