

Don't Just Trust the Model, Test the Physics: Evaluating PyTorch Models with PhysicsNeMo-CFD

An open, composable harness to benchmark PyTorch CFD surrogates on engineering metrics and uncertainty

Motivation

CFD is slow
A high-fidelity simulation of one design takes hours to days, so exploring thousands of designs is expensive.

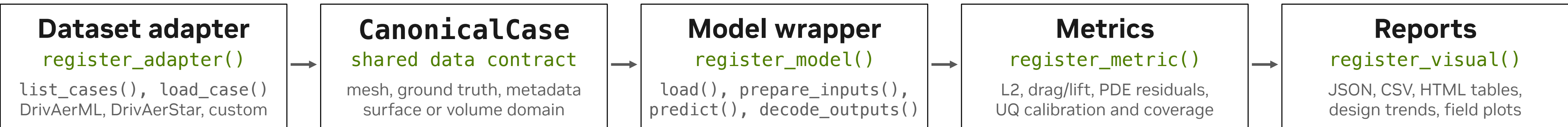
Surrogates are fast
PyTorch surrogates such as GeoTransolver, Transolver, DoMINO and X-MeshGraphNet predict flow fields in seconds.

Benchmarks don't line up
Teams hand-build their own splits, metrics and scripts, so results rarely reproduce and the state-of-the-art stays unclear.

Open datasets now exist: DrivAerML, DrivAerStar, AirfRANS, HiLiftAeroML. Models trained on them need a common yardstick.

A Composable Evaluation API

Four registries around one data contract: wrap a model once, adapt a dataset once, then score any model × dataset × metric combination. Extends the benchmarking framework of [1].



Bring your own PyTorch model

```
my_surrogate.py
import torch
from physicsnemo.cfd.evaluation.models import CFDMoel, register_model
from physicsnemo.cfd.evaluation.datasets import build_predictions_dict

class MySurrogate(CFDMoel):
    INFERENCE_DOMAIN = "surface" # or "volume"
    @property
    def output_location(self):
        return "cell" # or "point"
    def load(self, checkpoint_path, stats_path, device, **kw):
        self.net = MyNet().to(device).eval() # any nn.Module
        self.net.load_state_dict(torch.load(checkpoint_path))
        return self
    def prepare_inputs(self, case): # CanonicalCase in
        return featurize(case.mesh_path) # tensors out
    @torch.no_grad()
    def predict(self, x):
        return self.net(x)
    def decode_outputs(self, raw, case, model_input=None):
        return build_predictions_dict(pressure=raw["p"].cpu().numpy(),
                                     shear_stress=raw["wss"].cpu().numpy())

register_model("my_surrogate", MySurrogate)
```

Declare a benchmark matrix, run on N GPUs

```
conf/my_benchmark.yaml
benchmark:
  mode: matrix # every model × dataset
  models:
    - name: my_surrogate
      checkpoint: ckpt/my_model.pt
      stats_path: ckpt/stats.json
    - name: geotransolver_surface # from HF
    - name: transolver_surface
    - name: domino_surface
  datasets:
    - name: drivaml
      root: /data/drivaml
  metrics: [l2_pressure, l2_shear_stress, drag, lift]

terminal
$ torchrun --nproc_per_node=8 main.py --config-name=my_benchmark
```

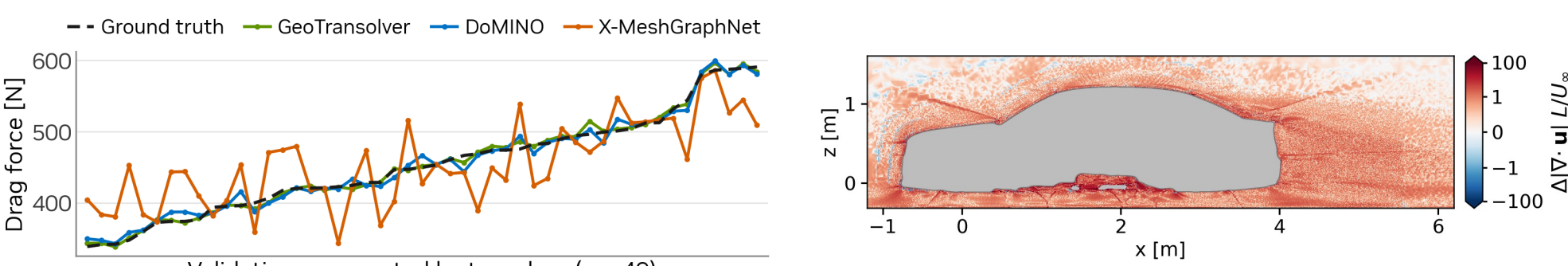
Included: pretrained GeoTransolver, Transolver, DoMINO, X-MeshGraphNet and FIGConvNet checkpoints on Hugging Face; multi-GPU runs with torchrun; agent skills that scaffold new wrappers, dataset adapters and metrics.

Benchmark Results

Pretrained checkpoints on DrivAerML validation cars [2]. Best in bold.

Model	L2 pressure ↓	L2 drag ↓	R² drag ↑	Spearman drag ↑
GeoTransolver	0.0373	0.0087	0.9943	0.9944
Transolver	0.0437	0.0107	0.9924	0.9944
DoMINO	0.1024	0.0166	0.9818	0.9903
FIGConvNet	0.2367	0.0637	0.7811	0.9743
X-MeshGraphNet	0.2782	0.0893	0.5333	0.7190

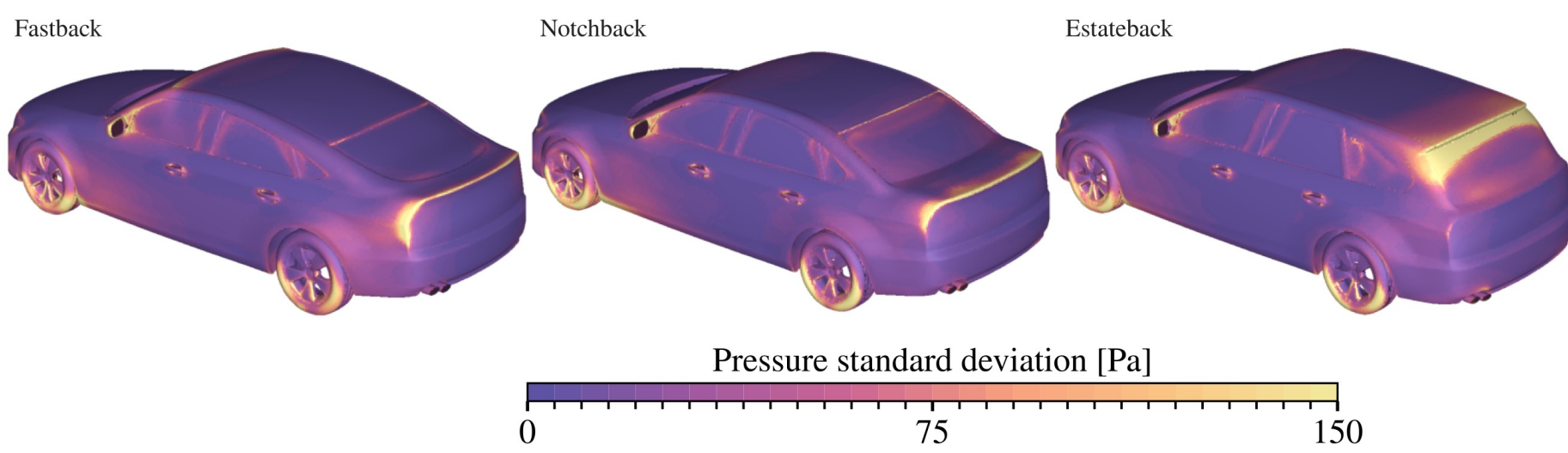
Beyond L2: engineering and physics diagnostics



Left: predicted vs. true drag for three surrogates. Right: continuity residual difference of the prediction relative to CFD.

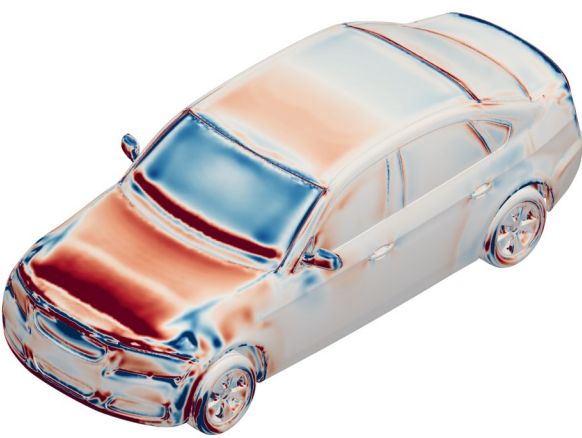
Uncertainty Quantification

- One distribution type:** Wrappers set SUPPORTS_UQ and return a FieldDistribution (mean, epistemic/aleatoric std, samples).
- Closed-form or sampling:** Different UQ methods like GP heads, MC dropout and deep ensembles share one engine: seeded passes, streaming aggregation and runs on multi-GPU.
- Metrics for design decisions:** NLPD, calibration, 95% coverage, sharpness, sparsification (AUSE), drag uncertainty.



GP surrogate trained on Fastbacks, tested on unseen Notchback and Estateback cars (DrivAerStar): uncertainty concentrates on the changed rear body [3].

From Benchmarks to Engineering Workflows

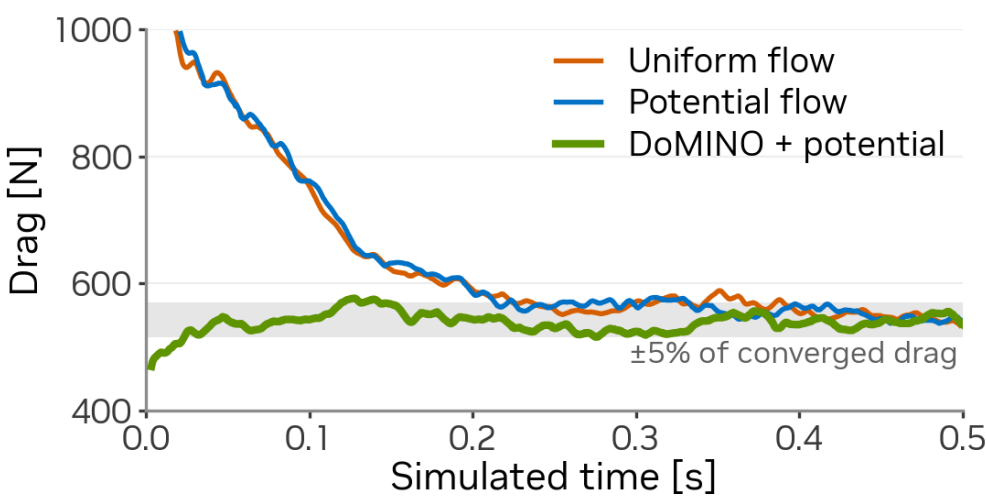


Design sensitivities

The same GeoTransolver checkpoint that predicts drag also shows where small shape changes would lower it most.

Solver initialization

Warm-start transient CFD runs from an AI surrogate prediction to achieve faster convergence [4].



PhysicsNeMo-CFD
github.com/NVIDIA/physicsnemo-cfd



Live demo

Try the pretrained models in your browser (Hugging Face)



PhysicsNeMo

github.com/NVIDIA/physicsnemo

[1] Tangsali et al. A Benchmarking Framework for AI Models in Automotive Aerodynamics. NeurIPS ML4PS 2025, arXiv:2507.10747
[2] Ashton et al. DrivAerML: High-fidelity CFD dataset for road-car external aerodynamics. arXiv:2408.11969, 2024
[3] Tangsali et al. Predictive Uncertainty for Neural CAE Surrogates. arXiv:2609.25430, 2026
[4] Sharpe et al. Accelerating Transient CFD through Machine Learning-Based Flow Initialization. arXiv:2503.15766, 2025