Session 3

Optimisation and Design

S3-A2	Farzad Mashayek — Optimal Pressure Sensor Placement for Reward Computation in Reinforcement-Learning-Based Active Flow Control of Airfoils
S3-A3	Francesco Di Fiore — Active Learning for Fluid Dynamics Driven Design
S3-A4	Izuru Kambayashi — LBM-based Deep Learning Surrogate Modeling for Shape Optimization of Fans
S3-A5	Long Ma — A Consistency-Identifiable Conditional Variational Autoencoder for Reliable Generative Shape Design
S3-A7	Tianhan Zhang — Autonomous Aerodynamic Design and Stability Optimization of UAVs via a Skill-Based Multi-Agent LLM Framework
S3-P1	Kherlen Jigjid — Adjoint-Based Training of Kolmogorov–Arnold Networks for Interpretable Turbulence Modeling
S3-P6	Vladimir Krasilnikov — AI-driven optimization of hull–propeller systems using CFD and propeller design codes
S3-P8	Youzheng Yu — Generative Framework for Aerodynamic Configuration Conceptual Design
S3-P9	Alfredo Lopez — A physics informed reinforcement learning framework for efficient airflow diffuser design
S3-P10	Larissa Jeindl — Optimization with Differentiable Flow Solvers: A Comparative Study of Bayesian and Gradient-Based Methods

Optimal Pressure Sensor Placement for Reward Computation in Reinforcement-Learning-Based Active Flow Control of Airfoils

[1]P. Thoguluva Rajendran, [1]D.T. Larsson, [1]L. Tronstad, [1]B. Kinsey, [1]S.A. Craig, [1]F. Mashayek*

[1]*Department of Aerospace & Mechanical Engineering, University of Arizona, USA*

Abstract

The present study investigates reconstruction of unsteady lift and quarter-chord pitching moment over a three-dimensional airfoil using a limited number of surface pressure probes while the flow is modified by fluidic actuation. This requirement is anticipated for practical deep reinforcement learning (DRL) training in wind-tunnel-in-the-loop active flow control experiments. We consider a three-dimensional NACA0024 airfoil at 8° incidence and chord-based Reynolds number 60,000, where the baseline flow exhibits a laminar separation bubble and shear-layer transition. Wall-resolved implicit large-eddy simulations are performed with the high-order Flux Reconstruction solver PyFR to provide time-resolved surface pressure fields for uncontrolled and actuated cases, with unsteady blowing through rectangular slots. Using these data, we train regression models that map sparse pressure-tap signals to estimates of lift and pitching moment over control intervals consistent with anticipated DRL decision times. Sensor placement is posed as a constrained optimization problem that minimizes reconstruction error subject to a fixed tap budget. The resulting placement guidelines and reconstruction error bounds quantify when pressure-only sensing is sufficient or insufficient for reward calculation and inform sensor layout and sampling requirements for future wind-tunnel DRL experiments.

Keywords: Sensor Placement Optimization; Deep Reinforcement Learning; Active Flow Control

*Corresponding Author, F. Mashayek mashayek@arizona.edu

Active Learning for Fluid Dynamics Driven Design

[1,2]Francesco Di Fiore*; [1,2]Hayriye Pehlivan Solak; [3]Andrea Serani; [3]Matteo Diez; [1,2]Laura Mainini

[1]*Department of Aeronautics, Imperial College London, London, UK*

[2]*Brahmal Vasudevan Institute for Sustainable Aviation, Imperial College London, London, UK*

[3]*Institute of Marine Engineering, National Research Council, Rome, Italy*

Abstract

Design efficiency plays a central role in reducing the environmental impact of air and sea transportation systems, as aerodynamic and hydrodynamic performance directly influence energy consumption and emissions. In aerospace and marine applications, design optimisation increasingly relies on high-fidelity simulations capable of resolving complex flow phenomena. However, their computational cost often limits their use in early-stage design exploration, creating a trade-off between model accuracy and optimisation tractability. This frequently reduces early design accuracy, with performance limitations only emerging at later stages when high-fidelity solvers are typically introduced, leading to increased development time, higher costs, and inefficient use of resources. This work proposes a scientific machine learning framework designed to improve the efficiency of design optimisation by enabling the systematic use of information from multiple modelling sources [1-3]. The methodology combines multifidelity active learning and lookahead learning to guide the adaptive selection of new design evaluations across fluid dynamic models characterised by different levels of accuracy and computational cost. Information from heterogeneous sources is progressively integrated during optimisation, allowing the algorithm to identify the most informative regions of the design space and the appropriate level of model fidelity to employ at each stage. By accounting for the expected future benefit of newly acquired data, the framework promotes the strategic and selective use of high-fidelity simulations, leveraging their accuracy to improve design outcomes while using lower-fidelity evaluations to accelerate exploration and maintain computational tractability. The proposed methodology is demonstrated on aerodynamic and hydrodynamic design problems representative of transportation applications, where performance is governed by complex flow-geometry interactions and high-fidelity analyses are typically required, illustrating its ability to improve design efficiency while managing the computational demands associated with physics-based simulations.

References

[1] Di Fiore, F., and Mainini, L., ‘‘NM2-BO: Non-Myopic Multifidelity Bayesian Optimization,’’ Knowledge-Based Systems, Vol. 299, 2024, p. 111959. DOI: 10.1016/j.knosys.2024.111959.

[2] Di Fiore, F., and Mainini, L., "Nm-mf: Non-myopic multifidelity framework for constrained multi-regime aerodynamic optimization." AIAA Journal, Vol.61, 2023, p. 1270-1280. DOI: 10.2514/1.J062219.

[3] Pehlivan Solak, H., Wackers, J., Pellegrini, R., Serani, A., Diez, M., et al., “Optimising hydrofoils using automated multi-fidelity surrogate models.” Ships and Offshore Structures, 2024, pp.1-12. DOI: 10.1080/17445302.2024. 2422518

Keywords:

Efficient Design Optimisation; Scientific Machine Learning; Multifidelity Active Learning; Lookahead Optimisation; Aerodynamic and Hydrodynamic Design.

LBM-based Deep Learning Surrogate Modeling for Shape Optimization of Fans

[1]Izuru Kambayashi*; [1]Masaki Ihara; [1]Szu Yung Chen; [1]Byungjin An;

[1]*Ebara Corporation*

Abstract

With the increasing power output of electric motors, achieving both low aeroacoustic noise and sufficient cooling performance in motor cooling fans has become a critical challenge. In the design process, performance prediction using Computational Fluid Dynamics (CFD) is indispensable. Specifically, unsteady analysis using the Lattice Boltzmann Method (LBM) enables high-accuracy prediction of aeroacoustic noise. However, the cumulative computational cost remains a major obstacle for shape optimization, which requires evaluating a large number of design candidates.

In this study, we investigated a shape optimization method with reduced computational cost by constructing a deep learning-based surrogate model using LBM results as training data. LBM simulations were performed on various fan geometries. While conventional surrogate models based on discrete geometric parameters, such as blade dimensions, often have limitations in accuracy, we confirmed that a deep learning model utilizing 3D shape data as direct input could accurately predict noise distribution and wind velocity at specific observation points within a practical timeframe.

Based on this surrogate model, we performed shape optimization considering both aeroacoustic characteristics and aerodynamic performance. The results suggest that even in LBM-based designs, which typically require extensive computation time, an optimal fan shape meeting target performance can be identified in a short period. Future work will focus on extending this approach to machine-level optimization, considering interactions with surrounding structures such as motor housings.

Keywords: Lattice Boltzmann Method, Deep Learning, Surrogate Model, Aeroacoustics, Shape Optimization

A consistency-identifiable conditional variational autoencoder with stochastic and deterministic fusion for reliable generative shape design

[1,3]Ma Long; [2,3]Wu Xiaojing; [1,3]Zhang Weiwei*;

[1]*School of Aeronautics, Northwestern Polytechnical University, Xi'an, 710072, China*

[2]*Institute of AI for Industries, Chinese Academy of Sciences, Nanjing, 211135, China*

[3]*International Joint Institute of Artificial Intelligence on Fluid Mechanics, Northwestern Polytechnical University, Xi'an, 710072, China*

Abstract

Conditional generative models can directly generate solutions that satisfy requirements. They offer the potential to improve efficiency and reduce empirical dependence in engineering design. However, inconsistency between generated results and conditions, as well as reliance on extensive labeled data, limits their practical application. This work proposes a consistency-identifiable conditional variational autoencoder by deriving a joint variational lower bound. Conditional generation and quality screening are unified within a single variational inference framework. Through a dual utilization of limited sample information, the model preserves generative capability via probabilistic latent modeling, while a regressor incorporating latent distribution information enables deterministic self-assessment of generation quality. Furthermore, we establish a generate-and-select workflow driven by a consistency identification module. During conditional generation, the model exploits large-scale candidate generation followed by autonomous quality selection to obtain reliable results. This mechanism transforms the traditional blind sampling generation process into a reliability selection process. Numerical inverse problems and airfoil design task demonstrate that, compared with state-of-the-art models, the proposed approach reduces the error between the generated results and the target conditions by over 20% with more than a sixfold reduction in training data.

Keywords: Conditional variational autoencoder; generative learning; shape design; cycle consistency; inverse problems

Autonomous Aerodynamic Design and Stability Optimization of UAVs via a Skill-Based Multi-Agent LLM Framework

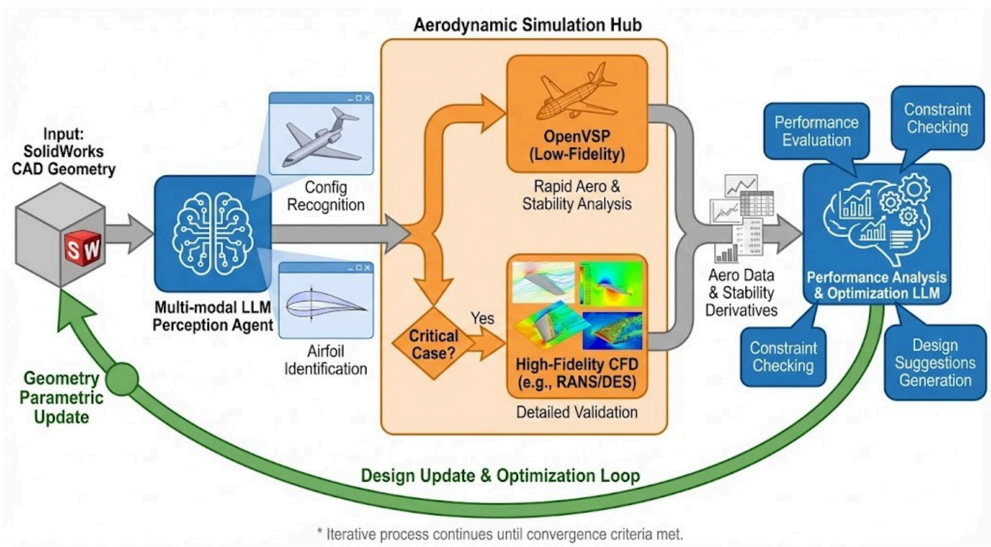
[1]Kang Hu; [1]Jiangyang Ge; [1]Haotian Jiang; [1,2] Tianhan Zhang*

[1]*School of Astronautics, Beihang University, Beijing 102206, China*

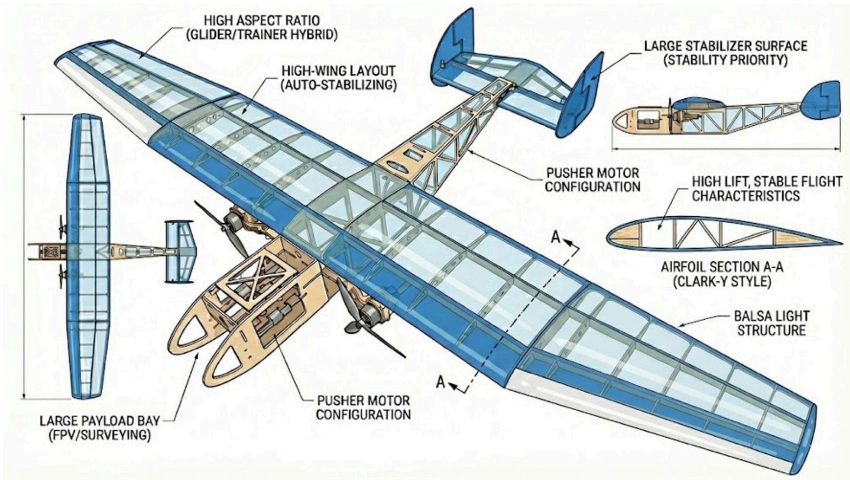
[2]*Key Laboratory of Spacecraft Design Optimization and Dynamic Simulation Technologies, 8 Ministry of Education, Beijing 102206, China*

Abstract

The integration of Large Language Models (LLMs) into complex engineering workflows represents a paradigm shift in fluid dynamics and aerospace design. Traditional aerodynamic optimization involves disjointed toolchains and relies heavily on manual expert intervention. We propose a novel Multi-Agent System (MAS) designed to autonomously orchestrate the complete aerodynamic design cycle of Unmanned Aerial Vehicles (UAVs).



Our methodology introduces a standardized "skill-definition protocol" (inspired by Claude Skill.md architectures) that encapsulates domain-specific knowledge into executable agent actions. This allows the system to seamlessly interface with heterogeneous tools: utilizing OpenVSP for parametric geometry modeling, and dispatching simulation tasks to OpenFOAM for high-fidelity flow physics and CFL3D for rapid aerodynamic assessment.The agents demonstrate the ability to generate scripts, debug errors, and synthesize multi-modal feedback from these solvers without human-in-the-loop.



We demonstrate the efficacy of this system by optimizing a fixed-wing UAV to mitigate lateral-directional instability (Dutch Roll). The design was validated through rigorous real-world flight experiments. Beyond visual evidence, we analyzed comprehensive flight logs to quantify performance. By isolating flight segments with constant airspeed and barometric altitude, we examined the time-domain response of the vehicle's attitude. The telemetry data revealed a significant reduction in the amplitude of coupled yaw-roll oscillations, confirming that the agent-optimized design successfully increased the damping ratio and eliminated the Dutch Roll mode. This work bridges the gap between generative AI and rigorous engineering physics, proving that LLM-based agents can effectively solve coupled non-linear problems in flight mechanics.

Keywords: Multi-Agent Systems, Aerodynamic Optimization, OpenFOAM, Flight Testing, Dutch Roll Control, LLM for Engineering.

Proceedings of the 2nd international Symposium on AI and Fluid Mechanics
Paper No [ ]

Adjoint-Based Training of Kolmogorov–Arnold Networks for Interpretable Turbulence Modeling

[1]K. Jigjid*; [1]A. A. Siddiqui; [2]N. A. K. Doan; [1]R. P. Dwight;

[1]*Department of Flow Physics & Technology, Faculty of Aerospace Engineering, TU Delft, 2629 HT Delft, The Netherlands;*

[2]*Department of Aeronautics, Imperial College London, SW7 2AZ London, United Kingdom*

Abstract

Improving the predictive capability of Reynolds-averaged Navier-Stokes (RANS) turbulence models remains an important challenge in computational fluid dynamics (CFD). Recently, data-driven approaches have been proposed to augment traditional turbulence closures using high-fidelity data. However, many machine learning models lack interpretability, which limits their usefulness for gaining physical insight and understanding model behavior. In this work, we develop an interpretable data-driven turbulence modeling framework in which Kolmogorov-Arnold Networks (KANs) [1] define the model structure, while model parameters are trained using an adjoint-based optimization approach implemented in DAFOAM [2].

KANs are based on the Kolmogorov-Arnold representation theorem [3, 4], which states that multivariate functions can be represented as compositions of univariate functions. In the KAN architecture, activation functions are parameterized using B-splines whose control points are optimized during training, rather than relying on fixed activation functions as in conventional neural networks. Training is performed using the adjoint-based optimization framework DAFOAM, built on OpenFOAM. Its discrete adjoint capabilities enable efficient computation of gradients of the flow solution with respect to model parameters, allowing the optimization to be performed directly within the CFD simulation.

The framework is demonstrated on the canonical periodic hill flow using two-dimensional direct numerical simulation (DNS) data for parameterized hill geometries [5]. Preliminary results show improved prediction of mean velocity profiles compared to the baseline k-omega SST RANS model. Ongoing work aims to interpret the learned KAN representation by approximating the univariate functions with symbolic expressions to understand how turbulence model corrections depend on local flow features.

Keywords: RANS; Data-driven turbulence modeling; Kolmogorov-Arnold Networks; Adjoint-optimization

References

1. Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark (2025). KAN: Kolmogorov-Arnold Networks. International Conference on Learning Representations (ICLR) 2025. arXiv:2404.19756.

2. P. He, C. A. Mader, J. R. R. A. Martins, and K. J. Maki (2020). DAFoam: An Open-Source Adjoint Framework for Multidisciplinary Design Optimization with OpenFOAM. AIAA Journal, 58:1304–1319.

3. A. N. Kolmogorov (1957). On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition. Doklady Akademii Nauk SSSR, 114:953–956.

4. V. I. Arnold (1957). On Functions of Three Variables. Doklady Akademii Nauk SSSR, 114:679–681.

5. H. Xiao, J.-L. Wu, S. Laizet, and L. Duan (2020). Flows over Periodic Hills of Parameterized Geometries: A Dataset for Data-Driven Turbulence Modeling from Direct Simulations. Computers and Fluids, 200:104431.

*Corresponding Author, K. Jigjid k.jigjid@tudelft.nl

AI-driven optimization of hull–propeller systems using CFD and propeller design codes

[1]Vladimir Krasilnikov*; [2]Benedetto Di Paolo; [2]Emiliano Costa

[1]*SINTEF Ocean AS, Trondheim, Norway;*

[2]*ENGYS Srl, Trieste, Italy*

Keywords: Ship hull, Propeller, Design, Open-source CFD, Optimization

Abstract. An automated framework for hull and propeller design optimization is presented, combining the CFD and dedicated propeller design codes with derivative-free evolutionary search (Nevergrad) and Bayesian optimization (BoTorch). The framework supports wake-adapted propeller blade design, parametric circulation shape control, and constraint enforcement on both geometric and hydrodynamic quantities, and is operated through a purpose-built graphical user interface. The methodology is demonstrated on the case of underwater hull and propeller design for a SWATH vessel. Both Nevergrad and BoTorch consistently locate the same optimal regions, confirming robustness of the results and providing substantial design improvement in terms of required propulsion power. The findings demonstrate that modern derivative-free and Bayesian optimizers, when tightly coupled to physics-based solvers through a structured evaluation pipeline, can efficiently explore mixed design spaces subject to engineering constraints, thus providing a viable route to AI-assisted optimization in industrial design workflows.

1. Introduction

Hydrodynamic optimization of ship hull and propulsor designs is one of the key enablers for reducing the environmental impact of shipping. The large number of design variables and constraints, together with multiple conflicting objectives (e.g., efficiency, stability, manoeuvrability, structural strength, acoustic performance, and cost), makes traditional design strategies costly. Manual design iterations and even automated design sweeps do not guarantee a thorough exploration of the design space. Furthermore, hull and propeller designs are often performed by independent actors with only limited information exchange (e.g., nominal wake field at the propeller, estimated propulsion factors, propeller open-water characteristics). Modern approaches therefore rely on reduced-order models (ROMs) and machine-learning (ML) surrogates for rapid early-stage design exploration [1], [2], followed by higher-fidelity calculations where the ship hull and propulsion arrangement are addressed as a system [3], [4].

This paper focuses on the design phase, where the main particulars of the ship hull and propulsion system are already decided, but the shapes of hull and propeller need to be optimized for maximum propulsive efficiency under prescribed constraints. The solution of this complex problem is driven by generic optimization algorithms which process the results of physics-based solvers. The use of analytical optimum condition for wake-adapted propellers [5], [6] ensures that for each hull variant, the propeller design meets the overall design objective. Typical propeller design constraints are given by the structural strength requirements, cavitation margins, and levels of propeller-induced pressure pulses and radiated noise, which are set by the class rules. To implement a cost-efficient optimization procedure, an automated multi-fidelity workflow is developed, combining CFD and dedicated propeller design tools. Parametric geometry engines are used to vary hull and propeller shapes. Optimization is performed using a two-stage strategy that combines derivative-free evolutionary search (Nevergrad [8]) for global exploration with Bayesian optimization (BoTorch [9]) for sample-efficient refinement. The following sections provide details about the software involved, their connections in the design workflow, and the design optimization strategies. As an application example, we consider the design of an underwater hull and propellers for a medium-size offshore Small Waterplane Area Twin Hull (SWATH) vessel [12].

2. Software and design workflow

The workflow implemented in the present study employs the open-source CFD package HELYX [10], developed by ENGYS, and the propulsor modelling suite AKPA [4], [11], developed at SINTEF Ocean, building on the experience with coupling these software packages as described in [7]. In HELYX, the solution of the viscous, incompressible, turbulent flow past the ship hull — without or with a working propeller — is obtained using the Reynolds-Averaged Navier–Stokes (RANS) equations method. For turbulence closure, the k–ω SST model is used. Propeller rotation is modelled either by the Moving Reference Frame (MRF) method or by the Sliding Mesh (SM) method with Arbitrary Mesh Interfaces (AMI). Alternatively, the effect of the working propeller can be represented by an Actuator Disk (AD) model that applies body force distributions provided directly by the AKPA propeller codes. The CFD simulation setup is automated using a Python journaling system and macro library in HELYX. The wake field at the propulsor location — either nominal or total, including propeller-induced velocities — is written in the file format supported by AKPA. To produce hull geometry, a suitable parametric model can be employed. In the present work, a simplified model capable of generating base shapes (bodies of revolution, foils, ducts) and their geometric transformations was developed.

The core modules of AKPA include the parametric propeller model AKPA_par, the non-linear lifting-line/lifting-surface blade design code AKPD, and the panel method propeller analysis code AKPA. In a typical propeller design workflow, given the design point (e.g., ship speed and resistance) and the wake field at the propeller location, one begins by generating a geometric propeller model based on preliminary estimates of the main particulars (e.g., diameter, hub ratio, number of blades, blade area ratio, total skew angle), which is then used in a wake-adapted blade design calculation. Both the design and analysis codes include an algorithm to predict the effective wake field on the propeller from the nominal wake computed by the CFD solver. At the first iteration, a CFD-based ROM can be employed to estimate the thrust deduction factor. The design code AKPD implements the Generalized Optimum Condition (GOC) [5] to derive the optimum circulation distribution along the blade radius, accounting for the radially varying axial and tangential components of the wake field as well as the thrust deduction. Alternatively, the following parametric form of the circulation distribution can be used:

, (1)

where is the radial coordinate, and , and are the parameters controlling blade loading distribution along the radius. This parametric form can be conveniently adopted to design propellers with an unloaded blade tip when improved acoustic performance is required, while retaining full adaptation to the wake field for maximum achievable efficiency. Optimization of propeller RPM within the bounds set by the engine specifications can also be performed at this stage. Likewise, the designer can explore the influence of other design variables such as propeller diameter, number of blades, blade area ratio, etc. Furthermore, AKPD implements a procedure to derive the maximum blade section thickness that meets class society requirements for blade strength. These features of the design code are essential for the robustness of the overall hull–propeller optimization process, as they eliminate the need for blade design iterations for each hull candidate by readily delivering the best propeller design for the given combination of design variables and constraints.

The propeller geometry resulting from the design calculation can be further evaluated by an unsteady panel method analysis using AKPA, for example to verify propeller cavitation performance and unsteady blade loads in a circumferentially non-uniform wake field. The propeller geometry is then passed to the HELYX solver to generate standard propeller open-water characteristics (OWC) and to verify propeller performance behind the ship hull by finding the actual RPM at which propeller thrust balances hull resistance. In the present workflow, it is possible to: (i) perform a complete CFD self-propulsion calculation with iterative RPM adjustment; or (ii) derive RPM from the computed OWC. If the difference between the design target and the actual propeller operating point behind the hull is larger than the required tolerance, the wake field and thrust deduction are updated from the latest CFD results and the propeller design calculation is repeated. When propeller performance requirements for the given hull are satisfied, it is possible to proceed to the next hull design candidate.

3. Optimization strategies

The optimizer is the mastermind behind the design process, and this is exactly where the Artificial Intelligence (AI) component enters the present development. The constrained design problem is formulated as the minimisation of the delivered shaft power subject to box constraints on the design vector and one or more inequality constraints enforced through three coexisting mechanisms (derived, penalty, and outcome). Two complementary search strategies are used and may be chained.

Strategy I — Derivative-free evolutionary exploration (Nevergrad). This strategy [8] targets global exploration of the design space and tolerates the noisy or discontinuous objectives that arise when integer variables (e.g. blade number) interact with the geometric and hydrodynamic constraints. We use the Nevergrad meta-optimiser *NGOpt*, which inspects problem dimensionality, budget, parallelism and variable types and dispatches to a problem-appropriate evolutionary or quasi-Newton sub-algorithm. The optimiser runs in an asynchronous “ask–tell” loop with thread-pool parallelism, batch size equal to the number of CFD/AKPA workers. The results are returned to the optimiser in submission order to keep the search trajectory deterministic across worker schedules. Constraint violations are handled by the penalty rule, , so the search treats the feasible region as the only admissible domain.

Strategy II — Bayesian optimisation (BoTorch). The second strategy [9] is intended for sample-efficient refinement once the feasible basin has been localised. However, it can also be applied directly and unconditionally to the initial design space. A Gaussian-process surrogate of the (sanitised) objective is built from the current evaluation history using BoTorch’s *SingleTaskGP* with input normalisation in the unit hypercube and output standardisation. A small, fixed observation noise stabilises the posterior in the presence of near-duplicate evaluations. This is an important feature, since in propeller optimization problems the sought optima are often flat. The candidate that maximises the acquisition function over the box is found by the dedicated procedure *optimize_acqf* with ten restarts and 128 raw Sobol-type samples. The two following acquisition options are currently implemented: (i) the analytic log-Expected-Improvement (*LogEI*); and (ii) the Monte-Carlo q-log-Noisy-Expected-Improvement (*qLogNEI*), which is used whenever any constraint is declared in *outcome* mode. When constraints are handled natively (*outcome* mode), the acquisition conditions expected improvement on the joint feasibility probability, , evaluated by Monte-Carlo sampling. This is preferred to the penalty path of the Strategy I (Nevergrad) in problems with a long, thin feasible boundary, because the surrogate learns where infeasibility begins and the acquisition is encouraged to probe the boundary rather than to recoil from it.

The two strategies share the same configuration, evaluation backend, history schema and constraint declarations, which makes their results directly comparable and enables a chain “Strategy I $\rightarrow$ Strategy II” pipeline where the Bayesian surrogate is initialised from the Nevergrad evaluation history, so refinement begins from an already informative posterior and avoids re-spending budget on redundant exploration. In practical design tasks, one can also apply an early-stopping rule to both stages, so the run is terminated when the best-so-far improvement over the last evaluations falls below a relative tolerance , with exposed in the run configuration as a user-defined setting.

4. Optimization management

The optimization framework is made accessible to designers through a dedicated cross-platform Graphical User Interface (GUI), *AKPA Propeller Optimization Studio,* implemented in Python using the *CustomTkinter* toolkit. It is organized as a single-window application with a navigation panel giving access to dedicated dialogs – Workflow, Operating Point, Design Variables, Design constraints and Strategy – where the user configures the entire exploration problem. Once a run is launched, the Monitor panel displays a live convergence plot, and the Runs panel presents a colour-coded ranked list of all evaluation folders. Finally, the Results and Compare panels enable analysis of results and side-by-side comparison of runs from different optimization cases or optimizer strategies, producing overlay convergence plots and summary tables.

5. Design examples

As an application example, we consider design of an underwater hull and propellers for a medium-size offshore Small Waterplane Area Twin Hull (SWATH) vessel [12] with the following main particulars:

length overall 49.35 (m), breadth under water 18.5 (m), draught 4.55 (m), displacement 900 (t), design speed 20 (knots). SWATH ships offer exceptional seakeeping performance in rough seas, as their submerged torpedo-shaped hulls – connected to the platform by slender struts – minimize the waterplane area and thus dramatically reduce wave-induced motions compared to conventional monohull or catamaran designs. However, the increased power demand due to larger wetted surface area, combined with the need to balance efficiency across a range of operating speeds, makes propulsor and hull design a considerably more demanding task than for conventional vessels, providing a compelling motivation for systematic optimization.

The baseline design, carried out without using optimization tools, established the main dimensions of the underwater hull and propeller: hull volume 382.75 (m^3) (two hulls accounting for 87.2 % of the ship's displacement volume), hull length 45.9 (m), propeller diameter 2.645 (m), and relative shaft submergence 1.038. It also provided an estimate of the required engine brake power per propulsor of 1700 (kW). This estimation was based on a statistical propulsive efficiency of =0.65 and accounts for additional resistance due to appendages (rudders, tunnel thrusters, stabilization fins), aerodynamic resistance and a 15% sea margin. Based on the estimated resistance and thrust deduction factor 0.18, a single 4-bladed propeller was designed to deliver thrust of 124.8 (kN) at the design speed.

Formal optimization exercises were then conducted. In the first exercise, the effect of underwater hull shape on propulsive efficiency was investigated using a simple design grid sweep. The hull volume was kept constant and equal to that of the baseline design, while the longitudinal position of the center of buoyancy (LCB) was varied according to Lackenby-style axial shift. The baseline propeller design was retained unchanged across all hull variants. Self-propulsion calculations were performed using the MRF method with a fully resolved propeller. Propulsion factors were derived from the computed propeller OWC. In Figure 1, five representative hull designs (des #1…5) are shown and compared in terms of their propulsive efficiency . The towing resistance is found to remain nearly identical for all studied hull variants. However, propulsive performance varies considerably due to differences in the ratio of thrust deduction () to axial wake fraction (). The best hull candidates achieve propulsive efficiency 0.70, compared to the baseline value of 0.65, corresponding to 3-3.5 % reduction in engine power. This result highlights the importance of optimizing propeller operating conditions behind the hull, rather than focusing solely on the reduction of hull resistance.

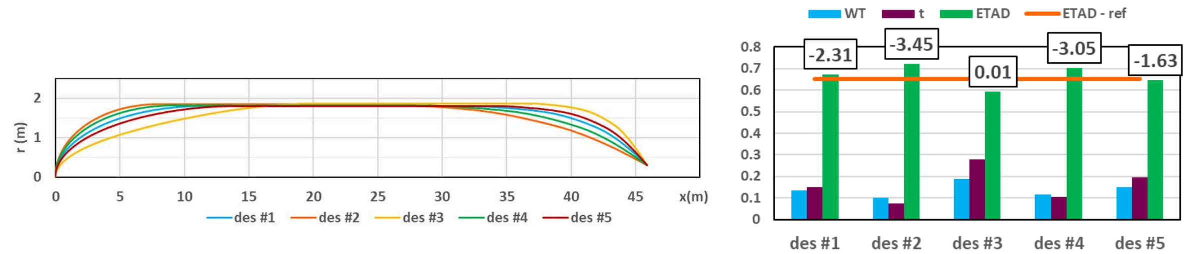


Figure 1 Longitudinal profiles of underwater hull (left) and corresponding propulsion factors (right). Labels show % power savings compared to the baseline design

Further improvements can be achieved by performing a wake-adapted propeller design. For this exercise, hull design #4 from the previous study was selected, owning to a more favourable LCB position that places it closer to the ship's center of gravity (LCG) – an important criterion for longitudinal stability of SWATH ships. Minimization of shaft delivered power was set as optimization objective. For the first propeller design, the design variables were propeller RPM and number of blades (), while radial circulation distribution was derived analytically from the Generalized Optimum Condition (GOC). In addition to the built-in blade strength constraint according to DNV class rules [13], an explicit derived constraint was applied to blade area ratio based on the Keller's criterion to prevent developed cavitation. Both the Nevergrad and BoTorch strategies were run over the same design space. The convergence of the two optimization runs is illustrated in Figure 2 (left). Both methods converge to the same minimum power within 0.03%, confirming a robust global optimum. The difference of 400 (W) is well within the numerical resolution of the design code and is physically insignificant. BoTorch converges faster and more decisively. It identifies the near-optimal region by evaluation #6, commits to Z = 6 / ~196 RPM by evaluation #17, and exploits that region for the remainder of the budget. Nevergrad explores more broadly throughout — including excursions to high-RPM candidates well past evaluation #30 — and requires roughly twice as many evaluations to reach its best. The objective landscape reveals a near-degeneracy between Z = 5 and Z = 6. Both solutions are equally valid from a practical standpoint, and the choice between them would be governed by secondary criteria such as vibration modes and manufacturing cost. Based on these considerations, the 5-bladed propeller design is recommended for the present case. The cumulative power savings resulting from the combined hull and propeller design optimization amount to 6.58% relative to the baseline.

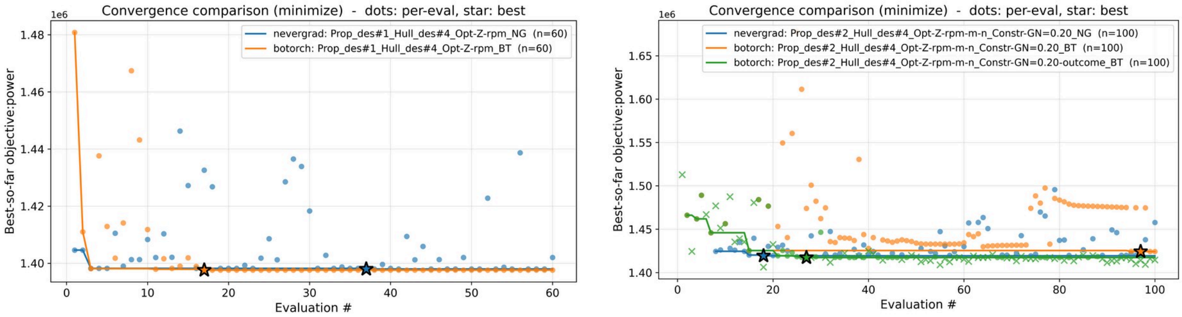


Figure 2 Convergence of design exploration processes for optimum propeller design (left) and low-noise propeller design (right). Hull des #4

While a propeller designed using GOC ensures highest propulsive efficiency, additional requirements may arise in some applications – for example, regarding the level of radiated noise. One relevant measure is the reduction of tip vortex strength which is achieved by unloading the blade tip. The parametric form of circulation distribution implemented in AKPD provides a convenient tool for such modification. Using the equation (1) and imposing an additional constraint on the normalized circulation at the reference blade radius (e.g., 0.95R), one can produce blade designs with improved cavitation/acoustic performance while still targeting minimization of power demand. Before launching the constrained optimization, unconstrained tests were conducted with the parametric loading shape in which the coefficients and were added to the design variables (four variables together with RPM and Z) to verify that the method recovers the circulation distribution corresponding to the GOC optimum. Both Nevergrad and BoTorch again converged to nearly identical results, yielding the shape of circulation very close to that of GOC at the radial stations $\geq 0.5R$, as illustrated in Figure 3.

With the performance of the parametric circulation form confirmed, an additional constraint was imposed on the normalized circulation at the radius 0.95R, GN0.95=0.20. This constraint renders infeasible – and thus excluded from the optimum search – all solutions with tip loading exceeding that threshold. The convergence histories shown in Figure 2 (right) reveal several noteworthy trends. All three optimizers converge to within 0.5% of the same minimum. However, the choice of constraint-handling mode has a decisive effect on BoTorch performance. With penalty mode, the Gaussian process surrogate is corrupted by the clipped $+\infty$ objective values assigned to violators, causing large exploratory excursions and delaying the best solution until evaluation #97. With outcome mode, BoTorch maintains a separate constraint GP, learns the feasibility boundary explicitly, and reaches its best solution at evaluation #27. Nevergrad reaches a competitive result (within 0.14% of the global best) already at evaluation #18, faster than either BoTorch variant in terms of first-arrival-at-optimum, but thereafter makes no further improvement – the penalty rule provides no gradient information about proximity to the constraint boundary, so the search stagnates once a feasible basin is found. In both cases, the optimization delivers a practically reliable design; notably, even with the tip-unloading constraint active, the required power remains below that of the baseline propeller, confirming that the efficiency penalty associated with the acoustic constraint is modest.

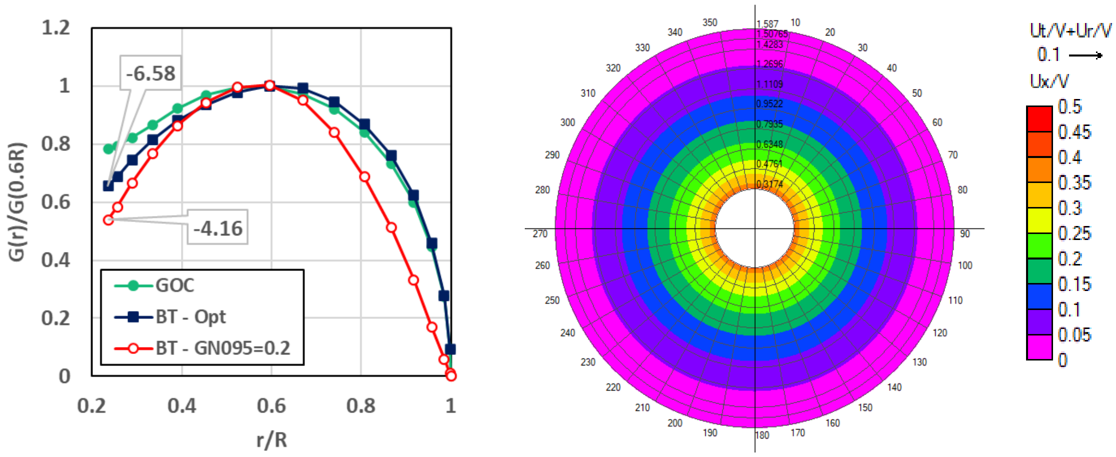


Figure 3 Radial circulation distributions for optimum and low-noise propeller designs (left) and effective wake field (right). Labels show % power savings compared to the baseline design

The effect of tip unloading on the flow field around blade tip is illustrated in Figure 4. Compared to the unconstrained optimum design, the strength of the tip vortex core is visibly reduced, the vortex itself appears more diffuse and deformed, and the suction-side pressure in the tip region is maintained at a higher level — all of which increase the margin against cavitation inception and contribute to a reduction in broadband noise.

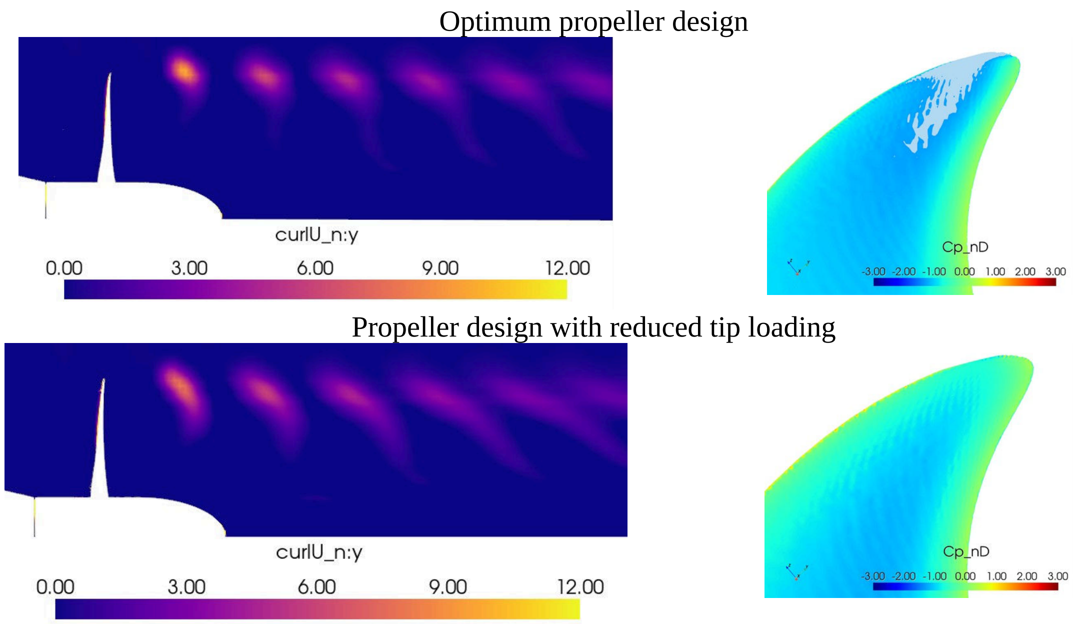


Figure 4 Visualization of tip vortex and pressure distribution in the blade tip area for the optimum and low-noise propeller designs

6. Conclusions

A software framework for automated design optimization of the system ship hull and propeller has been developed, coupling the open-source CFD solver HELYX and the propulsor modelling suite AKPA. The two complementary optimization strategies are presently implemented: derivative-free evolutionary search (Nevergrad) and Bayesian optimization (BoTorch). The framework is made accessible through a dedicated graphical user interface that covers the entire workflow. Case studies were conducted regarding the design of underwater hull and propeller for medium-size offshore SWATH vessel, demonstrating the capability of the framework on problems of increasing complexity. The findings highlight the importance of integrated approach to the hull and propeller design. For standard wake-adapted propeller optimization for minimum power in the space of RPM and number of blades (under the built-in constraint of blade are ratio to prevent development of cavitation), both the Nevergrad and BoTorch converged to the same optimum within 0.03%, showing 6.58% power savings compared to the baseline design. For problems involving more design variables and constraints such as low-noise propeller design, the choice of constraint treatment has a decisive effect on search efficiency. BoTorch with outcome mode reached the best feasible solution 3.6 times faster than BoTorch with penalty, while Nevergrad with penalty provided the fastest initial screening, finding a near-optimal feasible design within 18% of the evaluation budget. Taken together, these results support a hybrid two-stage workflow: Nevergrad for rapid global feasibility screening, followed by BoTorch with outcome-mode constraints for focused refinement.

7. References

[1] E. Costa, B. Di Paolo, A. Krassas, P. Geremia, C. Groth, M. E. Biancolini, M. Camponeschi, U. Cella and E. Di Meo ‘Interactive Reduced Order Models for Ship Hull Design and Optimization Coupling an Open-Source CFD Tool with Advanced RBF Mesh Morphing’, Computer and IT Applications in the Maritime Industries COMPIT’25, Pontignano, Italy, 7-8 October 2025

[2] M. Wheeler, D. Ponkratov, E. Arens, J. Hodges, S. Khan and J. Awan ‘Integrating AI into CFD simulations for Next Generation Ship Design’, 27th Numerical Towing Tank Symposium NuTTS 2027, Zagreb, Croatia, 14-16 September 2025

[3] V. Krasilnikov, K. Koushan, M. Nataletti, L. Sileo and S. Spence ‘Design and Numerical and Experimental Investigation of Pre-Swirl Stators PSS’, SMP’19 6th International Symposium on Marine Propulsors, Rome, Italy, 26-30 May, 2019

[4] V. Krasilnikov, ‘Design Exploration Study with Counter-Rotating Propellers for a Zero-Emission Coaster Vessel’, SMP’24 8th International Symposium on Marine Propulsors, Berlin, Germany, 17-20 March, 2024

[5] A. Achkinadze and V. Krasilnikov ‘A Generalized Optimum Condition for Wake Adapted Screw Propeller’, Propellers/Shafting’97 SNAME Symposium, Virginia Beach, VA, USA, 23-24 September 1997

[6] A. Achkinadze, V. Krasilnikov and I. Stepanov ‘A Hydrodynamic Design Procedure for Multi-Stage Blade-Row Propulsors Using Generalized Linear Model of the Vortex Wake’, Propellers/Shafting’2000 SNAME Symposium, Virginia Beach, VA, USA, 20-21 September 2000

[7] V. Krasilnikov and B. Di Paolo ‘Automated CFD tools for the design of marine propulsion systems’, eBook chapter pp 3-15, “Innovations in Sustainable Maritime Technology—IMAM 2025”, vol. 1, Springer, 2025; ISBN: 978-3-032-01432-0, DOI: 10.1007/978-3-032-01432-0

[8] ‘Nevergrad - A gradient-free optimization platform’, https://facebookresearch.github.io/nevergrad/

[9] ‘BoTorch - Bayesian Optimization in PyTorch’, https://botorch.org/

[10] ENGYS, HELYX CFD Package, https://engys.com/helyx/

[11] ‘AKPD/AKPA – Propeller design and analysis program’, https://www.scribd.com/document/253428658/AKPD-AKPA-ProgramPackage-pdf

[12] G. Schellenberger ‘SWATH Technology. Advanced SWATH Design Methods’, ThyssenKrupp Marine Systems, Nordseewerke, https://www.scribd.com/document/208739650/03-Swath-Technology-Small-Waterplane-Area-Twin-Hull

[13] DNV, Rules for Classification: Ships — Part 4 Systems and Components, Chapter 5 Rotating Machinery – Driven Units, Section 1 Propellers, DNV AS, Høvik, Norway.

Proceedings of the 2nd international Symposium on AI and Fluid Mechanics
Paper No [leave blank]

A generative adversarial network based method for aerodynamic configuration design of general aviation aircraft

[1,2,3]Yu Youzheng; [1,2,3]Kou Jiaqing*; [2,2,3]Zhang Weiwei;

[1]*School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China;*

[2]*International Joint Institute of Artificial Intelligence on Fluid Mechanics, Northwestern Polytechnical University, Xi'an 710072, China;*

[3]*National Key Laboratory of Aircraft Configuration Design, Xi'an 710072, China*

Abstract

Aerodynamic configuration design plays a critical role in the conceptual phase of aircraft development, where diverse design options must be efficiently generated and evaluated to identify optimal configurations. However, conventional aerodynamic design methodologies heavily rely on expert experience, which limits the exploration of innovative configurations and restricts the diversity of feasible solutions. Meanwhile, mainstream aerodynamic optimization approaches typically commence with the refinement of initial design stages, concentrating on local optimization within predefined configurations, rather than exploring the diversity of aerodynamic layout alternatives during the conceptual design phase. Recently, generative artificial intelligence has offered a novel solution paradigm for aircraft configuration design. This study presents a generative aerodynamic configuration design methodology based on Generative Adversarial Network (GAN), with application to a small general aviation aircraft. First, the GAN and its variants are evaluated for their capability in representing parametric space and generating aerodynamic configurations. Based on this analysis, a Conditional Wasserstein GAN with Gradient Penalty (CWGAN-GP) is established for efficient generation of aerodynamic configurations. Second, for wing design under low-speed cruise conditions (Mach number 0.2, angle of attack 2 degrees), the representation capability of different generative models in the parameter space and their ability to generate configurations under given conditions are analyzed, demonstrating the advantages of CWGAN-GP in generative design. Finally, for the conceptual aerodynamic configuration design of a general aviation aircraft under typical cruise conditions (Mach number 0.6, angle of attack 2 degrees), the CWGAN-GP model successfully generates a variety of aerodynamic configurations using the lift coefficient and lift-to-drag ratio as conditional parameters. By extending beyond the training parameter ranges, the proposed method exhibits strong generalization and extrapolation capabilities, capable of producing different aerodynamic shapes outside the original condition space. These results indicate that the generative design methodology presented offers a promising new paradigm for next-generation aircraft aerodynamic configuration exploration and innovation.

Keywords: generative models; aircraft design; aerodynamic configuration design; GAN; aerodynamics

Introduction

General aviation aircraft are widely used for civil transportation, regional mobility, emergency response, and agricultural operations. Their aerodynamic efficiency, fuel economy, stability, and safety are strongly influenced by the selected configuration. In a conventional aircraft development process, conceptual design establishes the baseline layout and therefore constrains most subsequent performance improvement[1]. Although this phase occupies only a limited portion of the development cycle, it determines the feasible design space for later preliminary and detailed design. A central requirement is therefore to generate multiple feasible aerodynamic layouts rapidly and to identify promising candidates before costly high-fidelity analyses are introduced.

Classical aerodynamic design methods can be broadly divided into direct optimization and inverse design[2,3]. Gradient-based methods such as adjoint or finite-difference sensitivity approaches are efficient for smooth objectives but are often sensitive to the initial design and may converge to local optima[4,5]. Gradient-free methods such as genetic algorithms, particle swarm optimization, and Bayesian optimization provide broader search ability, but their computational cost becomes significant in high-dimensional geometry spaces[6-8]. More importantly, many of these approaches start from a predefined configuration and focus on local refinement rather than layout-level exploration.

Machine-learning-based aerodynamic design has developed rapidly in recent years[9-11]. Surrogate modeling, deep-learning-assisted parameterization, and inverse design methods have improved the efficiency of airfoil and wing optimization. Generative models provide a different design paradigm: instead of iteratively searching for a single optimum, they learn a direct mapping between design requirements and candidate geometries. GANs, variational

*Corresponding Author, Kou Jiaqing jqkou@nwpu.edu.cn

autoencoders, and diffusion models have been applied to airfoil and reduced-dimensional aerodynamic shape design, but three-dimensional aircraft layout generation for conceptual design is still less mature. This gap motivates the present study.

The contribution of this work is a CWGAN-GP-based aerodynamic configuration design framework for general aviation aircraft. The method is first evaluated using a six-parameter wing benchmark to compare different GAN variants. It is then applied to Cessna Citation X, a 28-parameter-type aircraft configuration, where the lift coefficient and lift-to-drag ratio are imposed as conditional variables. The focus is not only on matching target performance, but also on retaining geometric diversity and physically meaningful aerodynamic trends.

Generative design framework

The proposed framework is shown in Figure 1. A parameterized geometry model is first constructed. Aerodynamic data are then generated by OpenVSP and VSPAERO, which are suitable for rapid conceptual design and early screening. The resulting database contains geometry variables and the corresponding aerodynamic coefficients. For conditional generation, aerodynamic targets are embedded into the generator and discriminator together with a random noise vector. Once training is complete, the generator can directly produce multiple geometry-parameter sets for specified aerodynamic requirements.

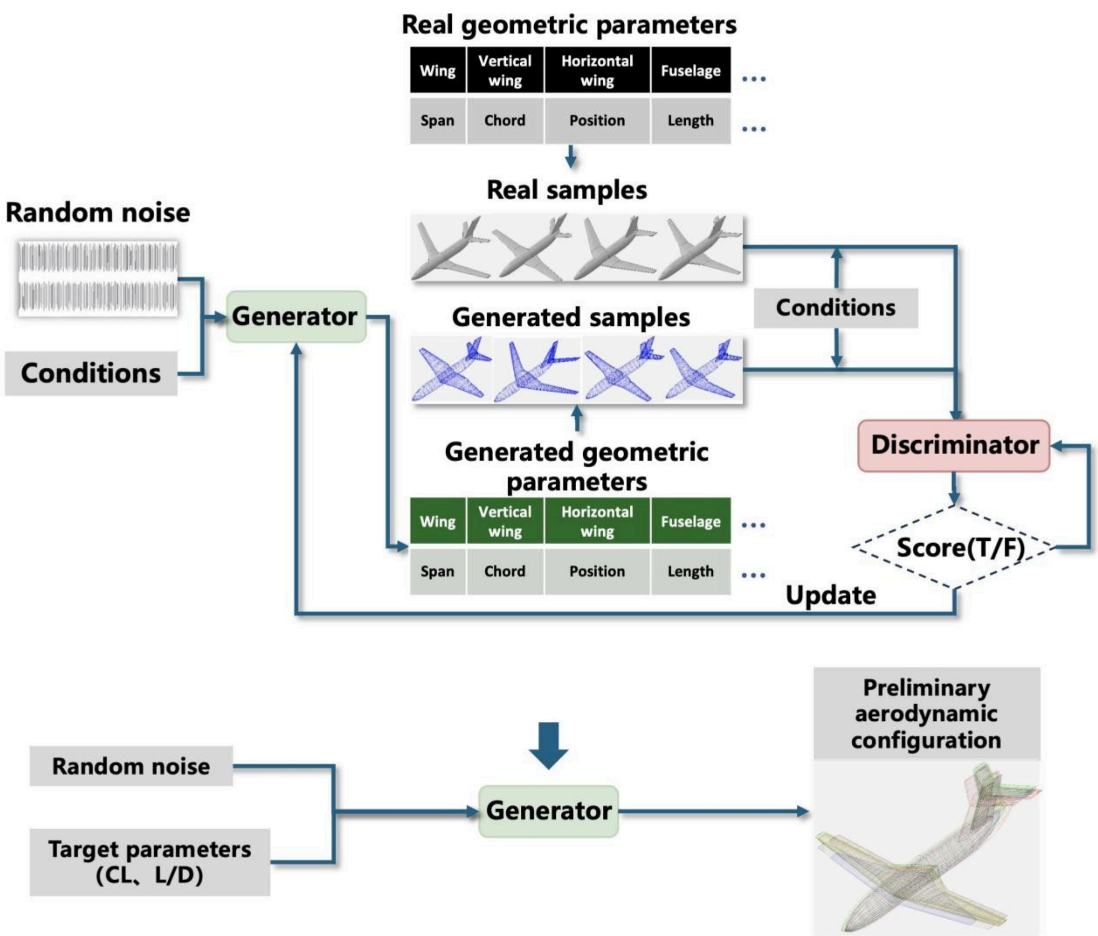


Figure 1 Generative aerodynamic configuration design framework based on CWGAN-GP.

*Corresponding Author, Kou Jiaqing jqkou@nwpu.edu.cn

CWGAN-GP combines conditional GAN, Wasserstein distance, and gradient penalty. The conditional variables guide the generated samples toward target aerodynamic properties, the Wasserstein formulation improves distribution learning and the gradient penalty avoids the instability caused by weight clipping. In this paper, unconditional GAN, WGAN, and WGAN-GP are used to evaluate design-space fitting, while CGAN, CWGAN, and CWGAN-GP are used to evaluate conditional design. Maximum mean discrepancy (MMD) is used to measure the similarity between generated and training distributions. Mean squared error (MSE) measures the deviation between the generated aerodynamic response and the target condition. The mean nearest-neighbor distance (MNND) in a t-SNE-reduced space is used as a diversity metric.

Datasets and model settings

Two datasets are used. The first is a wing dataset with six geometry variables: span, tip chord, root chord, dihedral angle, sweep angle, and NACA four-digit airfoil maximum thickness ratio. Latin hypercube sampling is used to generate 800 samples. Aerodynamic coefficients are computed at Mach 0.2 and 2° angle of attack. A preliminary longitudinal static-stability screening is applied so that the retained wing samples are physically meaningful at the design condition.

The second dataset is a small general aviation aircraft dataset derived from a Cessna Citation X-type configuration. The aircraft configuration includes main wing, horizontal tail, vertical tail, and fuselage parameters. Thirty-one design variables are initially sampled, with area constraints applied to the main wing and tail surfaces. After geometric dependency processing, 28 feature variables are used by the generative models. A total of 2500 samples are generated by Latin hypercube sampling. The aerodynamic database is computed at Mach 0.6, Reynolds number 2.0×10^6, and 2°angle of attack; lift coefficient and lift-to-drag ratio are used as the two performance conditions for generation.

Case	Geometry features	Samples	Flow condition	Conditional variable
Wing benchmark	6	800	Ma = 0.2, α = 2°	C_L
Aircraft configuration	28	2500	Ma = 0.6, Re = 2.0×10^6, α = 2°	C_L, *L/D*

Table 1 Summary of the two generative design datasets.

For the wing benchmark, all models use the same generator and discriminator structures to ensure a fair comparison. The generator contains three hidden layers with 512, 128, and 16 neurons and outputs six geometry variables. The discriminator contains hidden layers with 128, 64, and 16 neurons. GAN and CGAN use a sigmoid output in the discriminator, where as the Wasserstein-based models use an unrestricted scalar output. For the aircraft case, CWGAN and CWGAN-GP are retained according to the wing benchmark results. The generator output dimension is changed to 28, and the hidden layers are 512, 128, and 32 neurons. The conditional input dimension is two, corresponding to C_L and *L/D*.

Wing benchmark results

The wing benchmark is used to determine which generative formulation is most appropriate for aerodynamic configuration design. For unconditional generation, WGAN and WGAN-GP fit the original geometry distribution more accurately than the standard GAN, as indicated by smaller MMD values. This shows that replacing the Jensen-Shannon divergence by the Wasserstein distance improves the learned design-space representation. The standard GAN can reproduce some boundary values but exhibits stronger distribution mismatch in the interior of the parameter space.

For conditional generation with a target lift coefficient C_L= 0.18, the standard CGAN achieves a small conditional error but suffers from limited diversity. Its MNND is less than half of the values obtained by CWGAN and CWGAN-GP, indicating mode collapse in several geometry variables. CWGAN-GP provides the best overall compromise, with the lowest MSE and a high MNND. When multiple target lift coefficients are tested, CWGAN-GP gives the smallest mean bias and MSE, demonstrating that the target condition has a stronger and more stable guiding effect.

Model	Conditional	MMD	MSE at CL=0.18	MNND	MSE over targets

*Corresponding Author, Kou Jiaqing   jqkou@nwpu.edu.cn

GAN	No	0.1931	-	-	-
WGAN	No	0.0415	-	-	-
WGAN-GP	No	0.0960	-	-	-
CGAN	Yes	-	2.0×10-5	0.3721	3.61×10-5
CWGAN	Yes	-	5.3×10-5	0.8048	5.53×10-5
CWGAN-GP	Yes	-	6.0×10-6	0.8017	1.43×10-6

Table 2 Wing-case performance of different GAN variants.

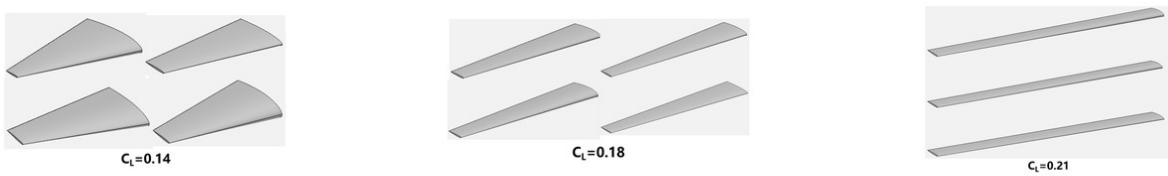


Figure 2 Typical wing samples generated by CWGAN-GP for different target lift coefficients.

The generated wing shapes further support this interpretation. As the target lift coefficient increases from 0.14 to 0.21, the generated wings tend to have larger aspect ratio, which is consistent with conventional aerodynamic design logic under the same low-speed condition. This indicates that CWGAN-GP does not merely interpolate parameter values, but captures a meaningful relationship between geometry and the imposed aerodynamic target.

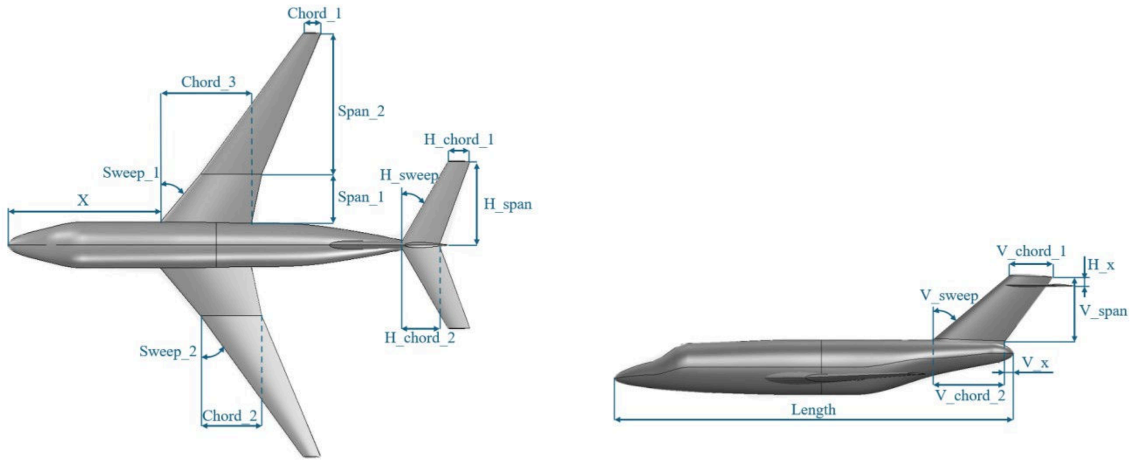
General aviation aircraft configuration design

The aircraft case evaluates whether the method can be scaled from a low-dimensional wing benchmark to a more practical three-dimensional layout problem. The Cessna Citation X-type model is parameterized by variables describing the main wing, horizontal tail, vertical tail, and fuselage, as illustrated in Figure 3. VSPAERO is used for rapid database generation. To assess the suitability of this low-order aerodynamic tool, one representative configuration is also computed by Ansys Fluent using a compressible Spalart-Allmaras turbulence model. The OpenVSP lift prediction agrees well with CFD, with an average relative error within 10%, while drag shows larger absolute discrepancies but similar variation trends. Therefore, VSPAERO is adequate for trend-oriented conceptual design and large-sample database construction in this study.

For the aircraft case, the target condition is set to C_L = 0.23 and L/D = 15. The performance comparison in Table 3 shows that CWGAN-GP substantially outperforms the other conditional models. Its MSE for lift is 1.0×10^{-5}, and its MSE for lift-to-drag ratio is 0.0409. Compared with CWGAN, the accuracy improvement is more than threefold for both target quantities. The MNND also increases from 0.6150 to 0.7983, showing that the improved accuracy is not obtained by collapsing all generated samples into a narrow geometry cluster.

The probability density distributions in Figure 4 confirm the quantitative results. The CWGAN-generated configurations do not concentrate sufficiently around the target values when the number of geometry features and conditional variables increases. In contrast, CWGAN-GP produces responses centered closer to both target quantities. This result suggests that the gradient-penalty formulation improves the conditional mapping in high-dimensional aerodynamic configuration spaces.

*Corresponding Author, Kou Jiaqing jqkou@nwpu.edu.cn



(a)Top-view *(b)Side view*

Figure 3 Parameterized Cessna Citation X-type aircraft model used for generative design

Model	Conditional	MMD	MSE for C_L	MSE for L/D	MNND
GAN	No	0.0859	-	-	-
WGAN	No	0.0413	-	-	-
WGAN-GP	No	0.0313	-	-	-
CGAN	Yes	-	9.1 x 10^-5	0.1417	0.4267
CWGAN	Yes	-	6.9 x 10^-5	0.3395	0.6150
CWGAN-GP	Yes	-	1.0 x 10^-5	0.0409	0.7983

Table 3 Aircraft-case performance metrics at C_L= 0.23 and L/D = 15.

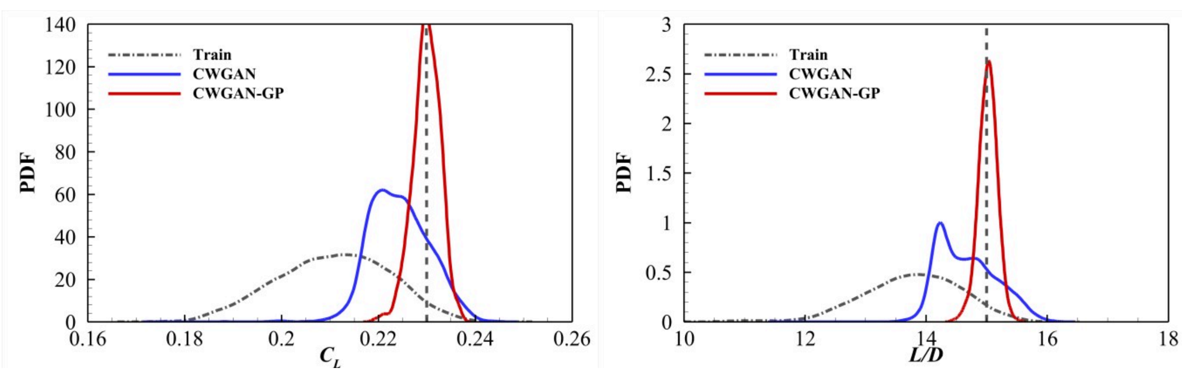


Figure 4 Probability density distributions of aerodynamic responses generated for C_L= 0.23 and L/D = 15.

The representative aircraft configurations in Figure 5 show diverse but physically interpretable configurations. At the cruise condition of Mach 0.6, the model tends to generate layouts with relatively large aspect ratio and moderate sweep, which is consistent with reducing induced drag and improving lift-to-drag ratio. The method also shows useful behavior outside the training condition range. When the target is extended to C_L = 0.25 and *L/D* = 17, CWGAN-GP still produces samples whose computed aerodynamic responses are closer to the target values than CWGAN. Off-design evaluation from 1°to 4°angle of attack further indicates that the generated designs retain favorable lift trends beyond the single training angle. In a static-stability check using samples with negative pitching-moment derivative, the generated layouts also retain negative $dC_M / d\alpha$, although this stability derivative is not explicitly introduced as a condition. This suggests that the model can inherit implicit physical constraints present in the training data.

*Corresponding Author, Kou Jiaqing jqkou@nwpu.edu.cn

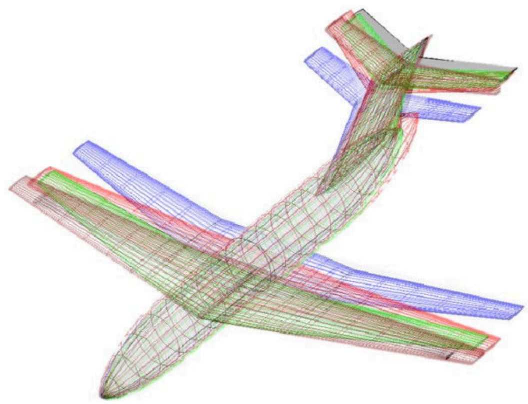


Figure 5 Representative aircraft layouts generated by CWGAN-GP under the target condition C_L= 0.23 and L/D = 15.

Conclusion

A CWGAN-GP-based generative design framework has been developed for aerodynamic configuration design of general aviation aircraft. The method learns an end-to-end mapping from target aerodynamic performance to geometry parameters and can generate multiple candidate configurations without iterative trial-and-error search. The main conclusions are as follows. First, the wing benchmark demonstrates that Wasserstein-based models describe the geometry distribution more accurately than the standard GAN, and CWGAN-GP provides the best balance between conditional accuracy and diversity. Second, when extended to a 28-feature aircraft configuration problem, CWGAN-GP improves the conditional accuracy by more than a factor of three compared with the other conditional GAN variants and increases the diversity metric by nearly 30%. Third, the generated layouts show meaningful aerodynamic trends, including aspect-ratio adjustment for lift and lift-to-drag requirements, and retain reasonable behavior under off-design angles of attack. Future work will incorporate larger datasets, richer geometry parameterizations such as cambered airfoils and wing twist, and explicit stability constraints to construct a more complete generative-design and forward-optimization framework.

References

1. Chen X Q, Yao W, Zhao Y. Multidisciplinary design optimization of flight vehicles theory and applications[M]. 2nd ed. Beijing: National Defense Industry Press, 2023: 34 (in Chinese).

2. Chen S S, Jia M L, Lin J H, et al. Empowering aircraft technology applications with generative models: Research progress and prospects[J]. Acta Aeronautica et Astronautica Sinica, 2025, 46(10): 631194 (in Chinese).

3. Du X S, Ren J, Leifsson L. Aerodynamic inverse design using multifidelity models and manifold mapping[J]. Aerospace Science and Technology, 2019, 85: 371-385.

4. Peter J E V, Dwight R P. Numerical sensitivity analysis for aerodynamic optimization: a survey of approaches[J]. Computers & Fluids, 2010, 39(3): 373-391.

5. Luo J Q, Xiong J T, Liu F. Aerodynamic design optimization by using a continuous adjoint method[J]. Science China Physics, Mechanics & Astronomy, 2014, 57(7): 1363-1375.

6. Holland J H. Adaptation in natural and artificial systems [M]. Cambridge: The MIT Press; 1992.

7. Kennedy J, Eberhart R. Particle swarm optimization[C]// Proceedings of ICNN'95 - International Conference on Neural Networks. Piscataway: IEEE Press, 1942-1948.

8. Rasmussen C E. Gaussian processes in machine learning[M]// Advanced Lectures on Machine Learning. Berlin, Heidelberg: Springer, 2004: 63-71.

*Corresponding Author, Kou Jiaqing jqkou@nwpu.edu.cn

A physics informed reinforcement learning framework for efficient airflow diffuser design

Alfredo López[1], Florian Sobieczky[1], Christopher Lackner[2], Matthias Hochsteger[2]
Bernhard Scheichl[3], Helmut Sobieczky[3], Christoph Feichtinger[4]
[1]SCCH Software Competence Center Hagenberg, Austria
[2]CERBSim GmbH, Austria
[3]Institute of Fluid Mechanics and Heat Transfer, TU Wien, Austria
[3]Windpuls Gmb, Austria

Abstract

This paper presents a physics-informed reinforcement learning framework for aerodynamic shape optimization applied to the design of a two-dimensional airflow diffuser. The objective is to optimize the diffuser geometry to maximize the homogeneity of the airflow at the outlet, guiding air from a ventilator towards a heat exchanger. This approach integrates computational fluid dynamics simulations with a physics-informed black-box optimization method that leverages low-cost physics-based features alongside expensive objective evaluations to efficiently explore the design space. The experimental results demonstrate that the proposed method effectively reconstructs the objective landscape and identifies near-optimal designs while reducing the computational cost.

1 Introduction

Aerodynamic shape optimization (ASO) plays a central role in the design of high-performance products and systems across various industrial sectors, such as aerospace, automotive, and renewable energy. ASO integrates computational fluid dynamics (CFD) simulations with optimization algorithms to iteratively assess and refine the design performance [9]. Physics-informed reinforcement learning (PIRL) has gained particular attention for shape optimization problems, where, in addition to the high dimensionality induced by the shape parameterization, the high computational expense of numerical simulations adds an additional layer of difficulty to these already challenging problems [4, 5]. Several PIRL approaches have been recently proposed for airfoil shape design of flying objects, where aerodynamic constraints are incorporated into the learning process using deep learning models [5, 7, 12, 16]. The present study deviates from these works by addressing a different application: the optimization of an airflow diffuser channel guiding air inside a device, where the objective is to design an optimal channel geometry that directs the (incompressible) airflow produced by a ventilator towards a heat exchanger, with the goal of maximizing the homogeneity of the flow before it reaches the exchanger under physics-based constraints.

2 Aerodynamic shape optimization

Our goal is to formulate and solve an ASO problem for the optimal design of a two-dimensional airflow diffuser that guides air driven by a ventilator toward a heat exchanger. We used the open-source software environment Netgen/NGSolve [13] to perform the CFD computations based on the finite-element method. The physical properties of the air were assumed uniform throughout the computational domain. We considered a constant outlet pressure boundary conditions, jointly with the typical 'do-nothing' outlet conditions, and a uniform, steady, and horizontal inlet flow. The diffuser shape is symmetric with respect to the horizontal aligned x-axis and has vertical inlet and outlet faces. For the actual diffuser considered, the relevant Reynolds number obtained from the inflow speed and reference diffuser characteristic length was approximately 2×10^5. The diffuser geometry and flow quantities were made non-dimensional according to the usual manner. Time-dependent implicit large-eddy simulations of the flow field were computed using the mass-conserving mixed-stress formulation method [8, 6]. Turbulence is triggered by numerical perturbations due to the spatial discretization (unstructured grid).

We denote by L the diffuser length, by h_{in} the inlet semi-height and by h_{out} the outlet semi-height, with $h_{\text{in}} < h_{\text{out}}$; see Figure 1. The diffuser shape is parameterized by $x \in [0, 1]^2$, which determines the position of the Bézier control point that defines the diffuser contour. Let $u_{t,x} : [0, L] \times [-h_{\text{out}}, h_{\text{out}}] \to \mathbb{R}^2$ and $p_{t,x} : [0, L] \times [-h_{\text{out}}, h_{\text{out}}] \to \mathbb{R}$ denote the CFD velocity field and pressure function at a given simulation time step t, associated with the geometry x. Let consider

$$\max_{x \in X} f_{\text{obj}}(x), \tag{1}$$

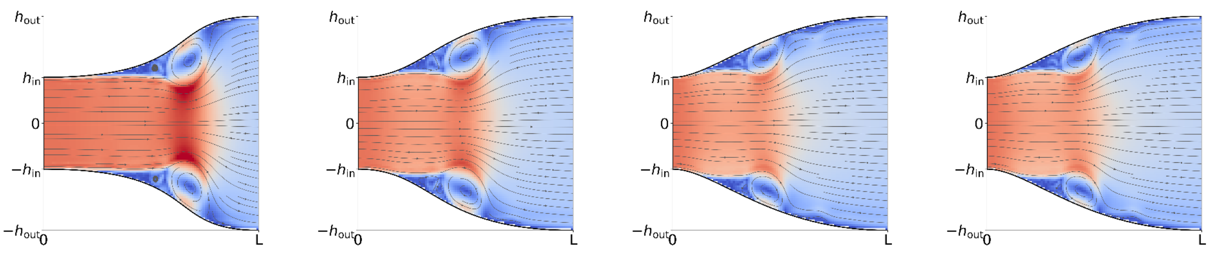
where $X \subset [0, 1]^2$ and $f_{\text{obj}} : [0, 1]^2 \to \mathbb{R}$ measures the time-averaged longitudinal homogeneity of the outlet velocity field obtained from CFD simulations of a channel shaped according to x. More precisely,

$$f_{\text{obj}}(x) = -\frac{1}{T} \sum_{t=1}^{T} \sigma(t, x), \tag{2}$$

where T is the number of simulation time steps,

$$\sigma(t, x) := \frac{\max_{s \in [-h_{\text{out}}, h_{\text{out}}]} \|u_{t,x}(L, s)\|^2 - \min_{s \in [-h_{\text{out}}, h_{\text{out}}]} \|u_{t,x}(L, s)\|^2}{\max_{s \in [-h_{\text{out}}, h_{\text{out}}]} |\pi u_{t,x}(L, s)|} \tag{3}$$

and π denotes the projection onto the first coordinate. It should be noted that small values of σ in (3) correspond to a horizontal and constant velocity field at the outlet. Consequently, the negative sign on the right-hand side of (2) is employed to frame this as a maximization problem. Figure 1 illustrates the diffuser shape for different values of x, along with a snapshot of the corresponding velocity field.



(a) (b) (c) (d)

Figure 1: Diffuser channel geometry and snapshots of the instantaneous velocity field for different shapes.

An important step in physics-informed machine learning is to extract low-dimensional physics-based features from CFD data guided by domain expertise [14, 11]. In our case of airflow channels, we assessed the pressure drop of the ideal pressure recovery at the outlet (estimated for a virtually steady inviscid and irrotational flow using Bernoulli's equation) and the actual pressure drop found by CFD (finite element) simulations, including RANS/temporal averaging. We define the pressure loss as

$$f_{\text{feat}}(x) = \frac{1}{T_0} \sum_{t=1}^{T_0} \lambda(t, x), \tag{4}$$

where $T_0 < T$ is a low-fidelity number of simulation time steps and $\lambda(t,x) = |\lambda_{\text{ber}}(t,x) - \lambda_{\text{num}}(t,x)|$ is the difference between the Bernoulli pressure drop $\lambda_{\text{ber}}(t,x) = \frac{1}{2}(h_{\text{in}}/h_{\text{out}})^2 - 1) \int_{-h_{\text{out}}}^{h_{\text{out}}} u_{t,x}^2(0,s)ds$ and the numerical pressure drop $\lambda_{\text{num}}(t,x) = \int_{-h_{\text{in}}}^{h_{\text{in}}} p_{t,x}(0,s)ds - \int_{-h_{\text{out}}}^{h_{\text{out}}} p_{t,x}(L,s)ds$. A smaller pressure loss indicates a more developed turbulent regime, which enhances momentum redistribution and results in a more homogeneous RANS-averaged flow at the outlet.

3 Physics-informed black-box optimization

Black-box optimization [3, 1] addresses objective functions whose gradients or internal structures are unavailable or non existing, as in the ASO problem of Section 2. Let $x \in X$ denote a candidate design and $f_{\text{obj}}(x)$ its performance metric to be maximized. Since $f_{\text{obj}}(x)$ is treated as a black box, analytical gradients and explicit expressions are unavailable. The proposed physics-informed black-box optimization (PIBBO) framework further incorporates inexpensive physics-informed descriptors $f_{\text{feat}}(x)$ capturing relevant physical characteristics of the design. These descriptors guide the optimization toward promising regions of the search space while reducing unnecessary high-cost evaluations. The relationship between objective and feature functions is modeled through a joint multi-output Gaussian process distribution,

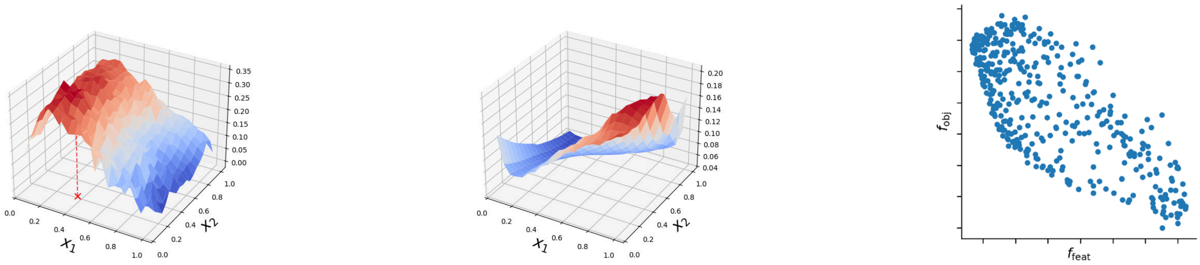
$$(f_{\text{obj}}, f_{\text{feat}}) \sim \mathcal{GP}(0, K). \tag{5}$$

The optimization procedure is formulated as a sequential decision-making problem. At iteration k, the algorithm selects a new design candidate x_k by maximizing the expected improvement, $x_k = \arg\max_{x \in X} \mathbb{E}\left[\max\left(0, f_{\text{obj}}(x) - f_{\text{obj}}(x_{k-1}^*)\right) \middle| \{(x_j, y_j, z_j)\}_{j=0}^{k-1}\right]$, where the expectation is computed with respect to the surrogate model (5) conditional on the history $\{(x_j, y_j, z_j)\}_{j=0}^{k-1}$, and x_{k-1}^* is the current best design. The quantities $y_k = f_{\text{obj}}(x_k)$ and $z_k = f_{\text{feat}}(x_k)$ are then evaluated (through CFD simulations in the ASO application) and added to the history. The overall strategy is formulated within a partially observable Markov decision process (POMDP) framework, where the environment state is only indirectly observed [15]. For a complete mathematical treatment, see [10].

4 Experiment and analysis

This section describes the application of the proposed PIBBO framework described in Section 3 to the diffuser shape optimization problem in (1). The search space X is defined as a 20×20 grid with coordinate values in $\{0.05, 0.1, \ldots, 1\}$. To establish a ground truth for this illustrative case study, we conducted the 400 simulations and computed the corresponding f_{obj} and f_{feat} from (2) and (4). We considered a diffuser with $L = 3$, inlet semi-height $h_{in} = 0.3$, and outlet semi-height $h_{out} = 0.7$. Each simulation used a 100×100 spatial grid over $[0, L] \times [-h_{\text{out}}, h_{\text{out}}]$, with $T = 724$ and $T_0 = 40$ time steps of size 0.025. The full set of simulations required approximately 15 h 34 min on a standard workstation, averaging 2.34 min per run. Figures 2a and 2b show the resulting objective f_{obj} and feature f_{feat} over X. An increasing trend in f_{obj} is observed along the first coordinate, whereas the second coordinate exhibits a more irregular behavior. Figure 2c shows the expected inverse relationship between objective and feature values. To model the joint distribution between the objective and feature functions, we employ the intrinsic coregionalization kernel K [2] in (5). The PIBBO method was run for 20 iterations without a stopping criterion. Figure 3 illustrates the resulting design x_k and surrogate model for iterations $k = 1, 8, 11, 13$. At iteration $k = 11$, the surrogate model closely captures the main objective pattern, and the design x_k lies near the true maximizer. The geometries of the associated airflow diffusers correspond to those depicted in Figure 1.

To evaluate the potential advantages of incorporating additional physics features, we conducted a version of the method that utilizes only the objective function information for the same problem. Our observations indicate that the inclusion of additional features results in only a slight improvement in the number of iterations and proximity to the true optimizer. We believe that the true benefit of incorporating additional features will become evident when employing a heterotopic sampling approach, wherein low-fidelity samples are densely collected compared to costly objective functions. This topic is further discussed in the conclusion section.



(a) (b) (c)

Figure 2: Two-dimensional surfaces of the objective function f_{obj} and feature f_{feat} over the search space X, together with the corresponding objective–feature scatter plot. The objective exhibits a clear increasing trend along the first coordinate and a more irregular dependence on the second coordinate, while the scatter plot reveals an inverse relationship between f_{obj} and f_{feat}.

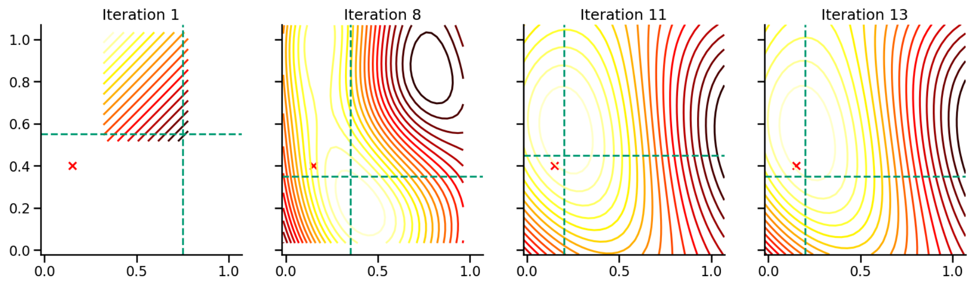


Figure 3: Selected designs x_k (dotted lines intersection) and corresponding surrogate probabilistic models (contour lines) at iterations $k = 1, 8, 11$, and 13. At iteration $k = 11$, the surrogate model closely reproduces the main pattern of the true objective function, and the selected design lies near the true maximizer (red cross).

5 Conclusion

We applied a physics-informed black-box optimization framework to a two-dimensional airflow-channel shape optimization problem. The results suggest that the proposed method effectively selects sampling points according to an exploration–exploitation strategy, thereby reconstructing the main patterns of the true objective function and attaining a near-optimal solution. The main technical contribution of the proposed framework is the use of additional low-cost feature evaluations to guide the sequential sampling process. To further investigate this idea, we are currently extending the action space to pairs of the form $a_t = (x_t, d_t)$, where d_t is a binary decision variable that determines the evaluation fidelity. If $d_t = 0$, only the inexpensive feature map $f_{\text{feat}}(x_t)$ is evaluated, whereas if $d_t = 1$, both $f_{\text{feat}}(x_t)$ and $f_{\text{obj}}(x_t)$ are evaluated. Our ultimate goal is to apply the proposed methodology to a real industrial airfoil shape optimization problem, in which geometries are described by more than 30 parameters, and each CFD evaluation requires several hours of computation on high-performance computing systems.

Acknowledgments

This research was primarily supported by the FFG project "AIrFoil" (FFG No. 915010). Additional funding was provided by BMIMI, BMWET, and the State of Upper Austria through the SCCH competence center INTEGRATE (FFG No. 892418) within the COMET program managed by the Austrian Research Promotion Agency (FFG).

References

[1] Stéphane Alarie et al. "Two decades of blackbox optimization applications". In: *EURO Journal on Computational Optimization* 9 (2021), p. 100011.

[2] Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. "Kernels for Vector-Valued Functions: A Review". In: *Foundations and Trends in Machine Learning* 4.3 (2012), pp. 195–266.

[3] Charles Audet and Warren Hare. *Derivative-Free and Blackbox Optimization.* eng. Springer Series in Operations Research and Financial Engineering. Cham: Springer International Publishing, 2017.

[4] Sahil Bhola et al. "Multi-fidelity reinforcement learning framework for shape optimization". en. In: *Journal of Computational Physics* 482 (June 2023), p. 112018.

[5] Thomas P. Dussauge et al. "A reinforcement learning approach to airfoil shape optimization". en. In: *Scientific Reports* 13.1 (June 2023), p. 9753.

[6] Jay Gopalakrishnan, Philip L. Lederer, and Joachim Schöberl. "A mass conserving mixed stress formulation for the Stokes equations". In: *IMA Journal of Numerical Analysis* 40.3 (July 2020), pp. 1838–1874.

[7] Amanda Lampton, Adam Niksch, and John Valasek. "Reinforcement Learning of a Morphing Airfoil-Policy and Discrete Learning Analysis". en. In: *Journal of Aerospace Computing, Information, and Communication* 7.8 (Aug. 2010), pp. 241–260.

[8] Philip L. Lederer, Xaver Mooslechner, and Joachim Schöberl. "High-order projection-based upwind method for implicit large eddy simulation". In: *Journal of Computational Physics* 493 (2023), p. 112492.

[9] Jichao Li, Xiaosong Du, and Joaquim R.R.A. Martins. "Machine learning in aerodynamic shape optimization". In: *Progress in Aerospace Sciences* 134 (2022), p. 100849.

[10] Alfredo Lopez et al. "Towards a PIRL Framework for Efficient Airflow Diffuser Design". In: *Proceedings of the Austrian Symposium on AI, Robotics, and Vision*. Vol. 3. 1. 2026, pp. 146–151.

[11] Riccardo Margheritti et al. "Feature Extraction from Flow Fields: Physics-Based Clustering and Morphing with Applications". In: *Applied Sciences* 15.23 (Nov. 2025), p. 12421.

[12] P. Scavella, G. Paolillo, and C.S. Greco. "Deep reinforcement learning-based airfoil design and optimization: An aerodynamic analysis". en. In: *Aerospace Science and Technology* 167 (Dec. 2025), p. 110638.

[13] Joachim Schöberl. "NETGEN An advancing front 2D/3D-mesh generator based on abstract rules". In: *Computing and Visualization in Science* 1.1 (July 1997), pp. 41–52.

[14] Pushan Sharma et al. "A Review of Physics-Informed Machine Learning in Fluid Mechanics". In: *Energies* 16.5 (2023).

[15] Matthijs T. J. Spaan. "Partially Observable Markov Decision Processes". In: *Reinforcement Learning: State-of-the-Art*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 387–414.

[16] Jonathan Viquerat et al. *Direct shape optimization through deep reinforcement learning*. en. arXiv:1908.09885 [cs]. Dec. 2020.

Proceedings of the 1st International Symposium on AI and Fluid Mechanics
Paper No []

Optimization with Differentiable Flow Solvers: A Comparative Study of Bayesian and Gradient-Based Methods

Larissa Jeindl[1,2], **Marian Staggl**[1,2], **Stefan Posch**[1,2]
[1] *CD Laboratory for Physics-driven Machine Learning in lndustrial Applications, Graz, Austria*
[2] *Institute of Thermodynamics and Sustainable Propulsion Systems, Graz University of Technology, Austria*
Corresponding author: Larissa Jeindl (larissa.jeindl@tugraz.at)

Keywords: Differentiable simulation, Fluid dynamics optimization, Bayesian optimization, Gradient-based optimization, Simulation-driven optimization, Quasi-Newton methods, L-BFGS

Abstract. Gradient-free and gradient-based optimization methods are widely used in engineering design; however, their practical performance strongly depends on the availability and accuracy of derivative information. While such methods have been extensively compared on analytical benchmark functions, their implications for optimization workflows based on fluid flow problems remain largely unexplored. Therefore, this paper investigates three optimization strategies, namely Bayesian optimization without gradients, gradient-enhanced Bayesian optimization, and the classical quasi-Newton method L-BFGS applied to optimization problems arising from fluid dynamics, where objective evaluations are computationally expensive and highly non-linear. In order to enable the determination of consistent objective function values and exact gradients, a fully differentiable flow solver is employed allowing controlled and fair comparison under realistic numerical conditions. Convergence behavior, sample efficiency, sensitivity to initialization, and robustness under simulation-driven conditions are analyzed across several two flow benchmark problems. A discussion on when differentiable flow solvers improve optimization performance and how they shape modern engineering optimization workflows is provided.

1. Introduction

The optimization of complex fluid dynamics problems often involves evaluating expensive loss functions. To minimize computational costs while identifying optimal design parameters, surrogate-based global optimization strategies are frequently employed. This paper systematically compares three distinct optimization methodologies: standard gradient-free Bayesian Optimization (BO), Gradient-Enhanced Bayesian Optimization via automatic differentiation (BO-AD), and the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm. The performance of these algorithms is evaluated across two canonical computational fluid dynamics (CFD) setups: a lid-driven cavity and a vortex-street setup. Morita et al. investigated a gradient-free Bayesian shape optimization for the lid-driven cavity case [1] and explicitly mentions gradient enhancement as a future research topic. Most applied research combining CFD with BO-AD relies on the adjoint method to compute gradients, rather than using a fully differentiable CFD solver. Furthermore, these studies tend to focus narrowly on shape optimization, leaving inverse problems, where the goal is to recover unknown inputs or parameters from observed output, largely unexplored [2, 3, 4]. More broadly, the utility of gradients from fully differentiable CFD solvers has been demonstrated across a range of applications: from learning to control and correct fluid simulations [5], to turbulence model learning with explicit analysis of gradient accuracy and computational cost [6], to recovering unknown viscosity fields from observed velocity data in an inverse problem setting [7]. However, none of these works investigates how autograd-derived gradients can enhance surrogate-based global optimization strategies such as BO, nor do they assess the trade-off between gradient accuracy and storage cost in that context, which are the central contributions of the present work.

2. Methodology

2.1. Bayesian Optimization and Gradient-Enhanced Bayesian Optimization

BO is a sequential, model-based strategy for the global optimization of objective functions that are expensive to evaluate [8, 9]. The goal is to find $\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$ with as few evaluations of the black-box objective f as possible [10]. The framework alternates between two steps: fitting a Gaussian Process (GP) surrogate to all observations collected so far, and maximizing an acquisition function $\alpha(\mathbf{x})$ to select the next evaluation point. A GP is fully specified by its mean $m(\mathbf{x})$ and kernel $k(\mathbf{x}, \mathbf{x}')$, i.e. $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ [11]. Common acquisition functions include Expected Improvement, Upper Confidence Bound, and Probability of Improvement, all of which balance exploration and exploitation [12, 13].

Gradient-Enhanced BO When gradient information $\nabla f(\mathbf{x})$ is available, the GP surrogate can incorporate derivative observations. Since differentiation is a linear operator, the joint distribution of function values and derivatives remains Gaussian [14], with cross-covariances obtained by differentiating the base kernel:

$$\mathrm{Cov}\left[f(\mathbf{x}),\, \frac{\partial f}{\partial x'_j}(\mathbf{x}')\right] = \frac{\partial\, k(\mathbf{x}, \mathbf{x}')}{\partial x'_j}, \qquad \mathrm{Cov}\left[\frac{\partial f}{\partial x_i}(\mathbf{x}),\, \frac{\partial f}{\partial x'_j}(\mathbf{x}')\right] = \frac{\partial^2 k(\mathbf{x}, \mathbf{x}')}{\partial x_i\, \partial x'_j}. \tag{1}$$

Each evaluation thereby contributes $d + 1$ pieces of information to the surrogate rather than one, where d is the number of optimization parameters, substantially improving the posterior between sparse observations [14]. A Matérn-$\frac{5}{2}$ kernel with automatic relevance determination (ARD), provided by GPyTorch [15], is used as the base kernel, as it provides the regularity required for the mixed partial derivatives [11]. Gradients $\nabla_{\boldsymbol{\theta}}\mathcal{L}$ are obtained via automatic differentiation (AD) through the differentiable CFD solver PICT [6], at a computational cost comparable to that of a single forward simulation pass. A single backward pass yields gradients with respect to all parameters simultaneously. The relevant hyperparameters are listed in Table 1 in the Appendix.

2.2. Limited-memory BFGS (L-BFGS)

L-BFGS is a widely used quasi-Newton optimization algorithm well-suited for parameter spaces where derivative information is available. Instead of computing the full, computationally expensive Hessian matrix, L-BFGS approximates it using a limited history of past gradient updates. Given an objective function $f(\mathbf{x})$, L-BFGS updates the current state $\mathbf{x}_k$ using the search direction $\mathbf{p}_k = -H_k \nabla f(\mathbf{x}_k)$, where H_k is the inverse Hessian approximation. While L-BFGS is highly efficient and exhibits rapid local convergence, it is inherently a local optimizer and is highly susceptible to getting trapped in local optima depending on the initialization [16].

3. Applications

3.1. Test Case 1: Lid-Driven Cavity

The first test case is the two-dimensional lid-driven cavity, a classical CFD benchmark [17]. The square domain of side length $L = 1.0$ is discretized on a 32×32 structured grid with wall-normal growth ratio of 1.1 at all boundaries. The lid imposes a prescribed horizontal velocity while all other walls enforce no-slip conditions, and the incompressible Navier–Stokes equations are solved using the differentiable PISO-based solver PICT [6].

The two free optimization parameters are the viscosity $\nu \in [2 \times 10^{-3},\, 2 \times 10^{-2}]$ and the lid velocity $u_{\mathrm{lid}} \in [-5.0,\ 5.0]$, giving $\mathbf{x} = (\nu,\, u_{\mathrm{lid}})^\top$. Each evaluation consists of 100 pseudo-time steps ($\Delta t = 0.25$), with the final differentiable iterations retaining the computational graph to enable gradient propagation. The objective is the Mean-Squared-Error between the simulated and a reference velocity field $\mathcal{L}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{u}_i(\mathbf{x}) - \mathbf{u}_i^{\mathrm{ref}}\right)^2$, where $N = 2 \times 32 \times 32 = 2048$. The problem thus constitutes an inverse problem, where the optimizer must recover the parameter combination that best reproduces the target flow state.

3.2. *Test Case 2: Vortex Street*

The second test case is a two-dimensional channel flow past a square obstacle, inducing a von Kármán vortex street. The domain is discretized on an eight-block structured grid with resolution scale $r_{\text{scale}} = 32$ and a fixed time step of $\Delta t = 0.1\,\text{s}$. The single design parameter is the kinematic viscosity $\nu \in [2 \times 10^{-5}, 7 \times 10^{-5}]$, and the objective is to match the amplitude of the downstream velocity oscillations to a prescribed target $A_{\text{ref}} = 1.0$, where the amplitude is defined as half the peak-to-peak range of the signal, $A = \frac{u_{\max} - u_{\min}}{2}$:

$$\mathcal{L}(\nu) = \frac{1}{N} \sum_{i=1}^{N} \left(A_i(\nu) - A_{\text{ref}}\right)^2 . \tag{2}$$

A parameter-dependent initial state is precomputed for each viscosity, followed by $N_{\text{warm}} = 100$ steps without gradient tracking to reach a statistically stationary state. AD is then enabled for the final $N_{\text{diff}} = 35$ steps (or $N_{\text{diff}} = 25$), differentiating through the resolved periodic regime only. Here, steps correspond to physical time iterations rather than solver iterations. This truncated backpropagation-through-time strategy limits peak memory usage while preserving a meaningful gradient signal. Although this truncated strategy is an approximation, it proves sufficiently accurate after only a few back-propagated time steps. Being one-dimensional, this case allows exact visualization of the GP surrogate and acquisition landscape, providing a controlled benchmark for assessing the benefit of gradient-enhanced BO.

4. Results

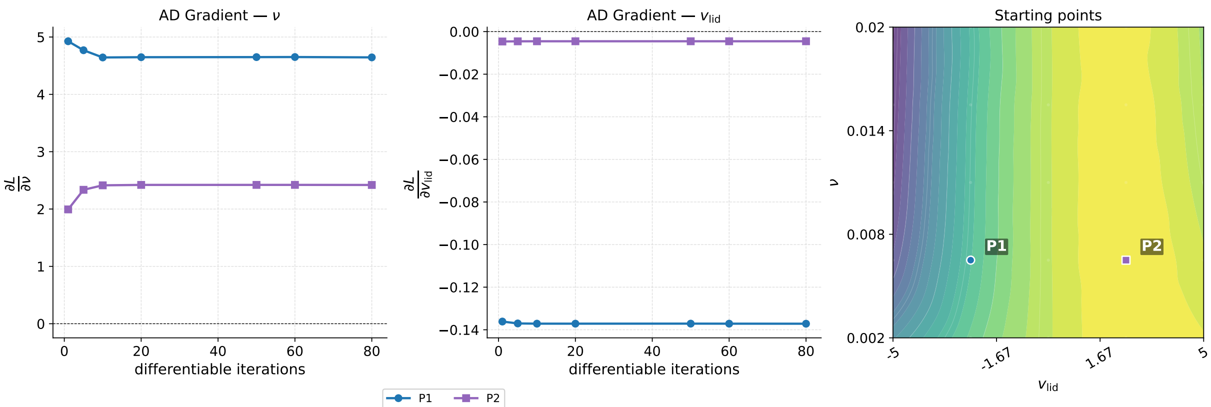
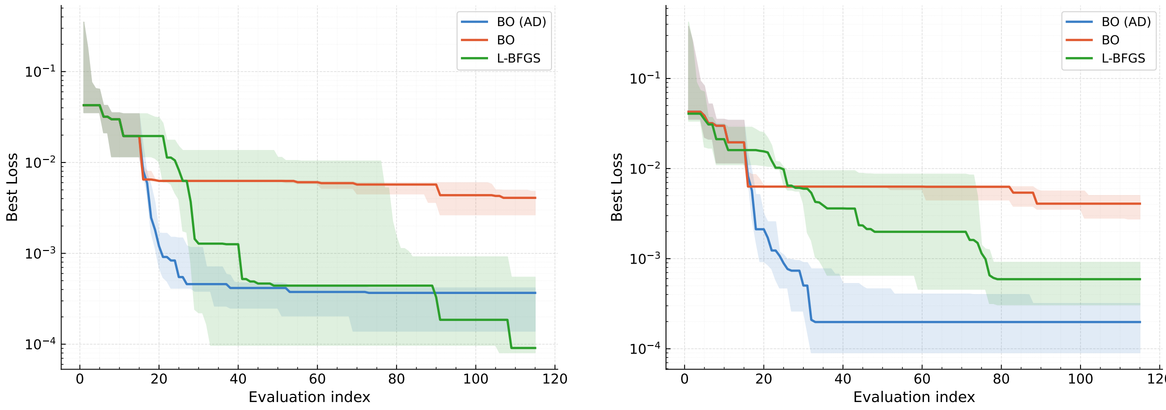


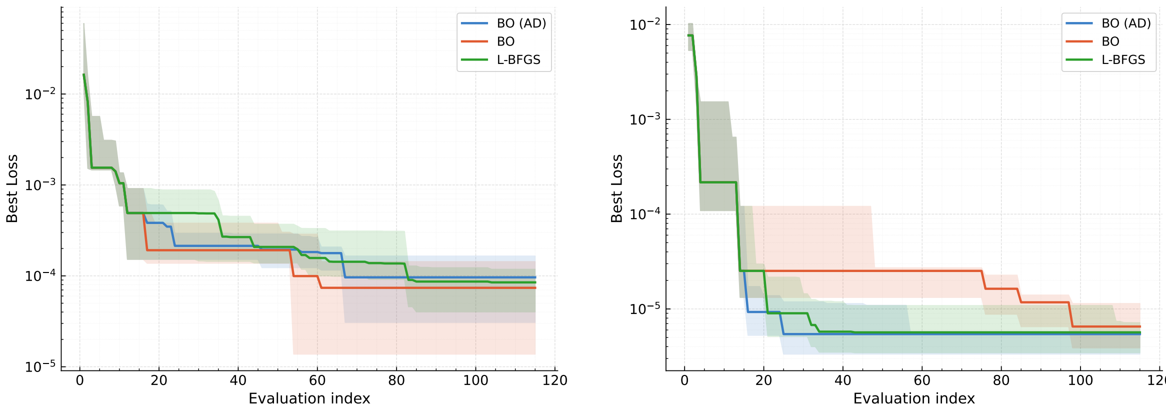
Figure 1: Gradient computed via AD as a function of differentiable iterations at two probe points in the lid-driven cavity domain.

A critical parameter in the optimization setup is the number of differentiable iterations, which must be carefully determined. Since the computational graph grows with each additional iteration, it is not feasible to differentiate through the entire simulation; although alternatives such as gradient checkpointing exist, they are not employed here. A suitable truncation point must therefore be established to reduce GPU memory exhaustion. To the best of the authors' knowledge, no prior work has systematically addressed this question. To investigate the sensitivity of the gradient to this parameter, two probe points within the lid-driven cavity domain were monitored as the number of differentiable iterations was varied. As shown in Figure 1, the gradient changes substantially when increasing from 1 to 10 differentiable iterations, beyond which it remains effectively constant. Based on this observation, values of 10 and 20 differentiable iterations were selected for further investigation in the lid-driven cavity case. BO-AD outperformed both L-BFGS and BO without gradient information, with BO-AD exhibiting significantly faster convergence than L-BFGS, underscoring the advantage of incorporating gradients into the surrogate-based optimization framework. Figure 2 presents the lid-driven cavity case evaluated across seven different seeds for both 10 and 20 differentiable iterations. As shown in Figure 2a, the 10-iteration setting yields slightly less accurate results overall compared to its 20-iteration counterpart (Figure 2b). Notably, L-BFGS achieves comparatively better results at 10 differentiable iterations; however, its performance exhibits a considerably stronger dependence on the choice of seed, indicating lower robustness. A possible explanation is that truncated backpropagation reliably captures the gradient direction even with few iterations, while the gradient amplitude may be mis-scaled. Since L-BFGS determines its step size via a line search, it is inherently robust to such amplitude errors and benefits primarily from an accurate search direction.



(a) (b)

Figure 2: Different convergence results in the lid-driven cavity case for 7 different seeds: BO using gradients, no gradient and L-BFGS. (a) shows the results with 10 differentiable iterations, (b) shows 20 differentiable iterations using AD.



(a) (b)

Figure 3: Different convergence results in the Vortex street case for 3 different seeds: BO using Gradients, no gradient and L-BFGS. (a) shows the results with 25 differentiable iterations, (b) shows 35 differentiable iterations using Automatic Differentiation

The vortex street case is examined in Figure 3 for 25 and 35 differentiable iterations. The 35-iteration setting was selected to ensure that the periodic flow propagates through at least one full period within the tracked cell, while the 25-iteration setting was included to assess the effect of an incomplete period on optimizer performance. As shown in Figure 3a, the reduced number of differentiable iterations yields considerably less accurate results across all optimizers, and the resulting gradient inaccuracies are found to mislead the gradient-based methods. In contrast, for 35 differentiable iterations (Figure 3b), BO-AD and L-BFGS exhibit similar convergence behavior, though BO-AD reaches a converged state in slightly fewer iterations. Both outperform the gradient-free solution.

5. Discussion and conclusion

This work systematically compared three optimization strategies: gradient-free BO, BO-AD, and L-BFGS, applied to two canonical fluid dynamics inverse problems using a fully differentiable flow solver.

A key finding concerns the number of differentiable iterations required to obtain meaningful gradient information. While one might expect that differentiating through the entire simulation is necessary for accurate gradients, the results demonstrate that a substantially smaller number of differentiable iterations is sufficient, with gradients stabilizing well before the simulation fully converges. This has direct practical implications: accurate gradient information can be obtained at a fraction of the memory cost of full differentiation. However, the number of differentiable iterations must be chosen carefully: if too few iterations are used, the gradient signal becomes inaccurate and can actively mislead gradient-based optimizers, as observed in the vortex street case with 25 differentiable iterations.

Regarding optimizer performance, BO-AD outperformed gradient-free BO across both test cases, confirming that AD-derived gradients substantially enrich the surrogate model and accelerate optimization. Compared to L-BFGS, BO-AD demonstrated superior robustness across different initializations, while L-BFGS showed stronger sensitivity to the choice of seed. Nevertheless, when gradients are inaccurate, both gradient-based methods are adversely affected, highlighting that gradient quality is a prerequisite for reliable performance.

In conclusion, this paper demonstrates that gradient-enhanced BO, enabled by fully differentiable flow solvers, represents a compelling strategy for simulation-driven optimization in CFD. The method combines the global exploration behavior of BO with the efficiency of gradient information, offering advantages in convergence speed and sample efficiency over its gradient-free counterpart. The trade-off between gradient accuracy and memory cost is identified as a critical design consideration in such workflows. Future work should investigate the scalability of the proposed approach to higher-dimensional parameter spaces. As each simulation evaluation contributes $d + 1$ informative observations to the surrogate rather than one, the relative advantage of gradient-enhanced methods is expected to become more significant with increasing dimensionality, a behavior further supported by Horvath et al. [18].

Acknowledgments

The financial support by the Austrian Federal Ministry of Economy, Energy and Tourism, and the Christian Doppler Research Association is gratefully acknowledged. Furthermore, the authors want to thank Prof. Manfred Ulz from the Institute of Strength of Materials at Graz University of Technology for his valuable input.

References

[1] Y. Morita, S. Rezaeiravesh, N. Tabatabaei, R. Vinuesa, K. Fukagata, and P. Schlatter, “Applying Bayesian optimization with Gaussian process regression to computational fluid dynamics problems,” *Journal of Computational Physics*, vol. 449, p. 110788, Jan. 2022.

[2] J. Müller, J. M. Reumschüssel, T. Kaiser, S. Knechtel, and K. Oberleithner, “Combining Bayesian optimization with adjoint-based gradients for efficient control of flow instabilities,” in *Proceedings*

of the Summer Program 2024, Center for Turbulence Research, 12 2024.

[3] C. Sabater and S. Görtz, *Gradient-Based Aerodynamic Robust Optimization Using the Adjoint Method and Gaussian Processes*, pp. 211–226. Cham: Springer International Publishing, 2021.

[4] C. Talnikar and Q. Wang, "Adjoint-based trailing edge shape optimization of a transonic turbine vane using large eddy simulations." arXiv preprint arXiv:2011.06744, 2020.

[5] P. Holl, V. Koltun, K. Um, and N. Thuerey, "phiflow: A differentiable PDE solving framework for deep learning via physical simulations," *Workshop on Differentiable Vision, Graphics, and Physics in Machine Learning at NeurIPS*, 2020.

[6] A. Franz, H. Wei, L. Guastoni, and N. Thuerey, "PICT–a differentiable, GPU-accelerated multi-block piso solver for simulation-coupled learning tasks in fluid dynamics," *Journal of Computational Physics*, vol. 544, p. 114433, Jan. 2026.

[7] S. Posch, M. Staggl, W. Sanz, "Differentiable simulation for inverse-like fluid flow problems," in *Proceedings of the 1st International Symposium on AI and Fluid Mechanics*, 2025.

[8] E. Brochu, V. M. Cora, and N. de Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning." arXiv preprint arXiv:1012.2599, 2010.

[9] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. d. Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[10] P. I. Frazier, "A tutorial on Bayesian optimization." arXiv preprint arXiv:1807.02811, 2018.

[11] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[12] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[13] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proceedings of the International Conference on Machine Learning 2010*, 2010.

[14] J. Wu, M. Poloczek, A. G. Wilson, and P. I. Frazier, "Bayesian optimization with gradients," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[15] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[16] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[17] U. Ghia, K. Ghia, and C. Shin, "High-re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method," *Journal of Computational Physics*, vol. 48, no. 3, pp. 387–411, 1982.

[18] P. Horvath, M. Staggl, and S. Posch, "Derivative-enhanced training for data-efficient surrogate modeling," in *Proceedings of the Austrian Symposium on AI, Robotics, and Vision*, vol. 3, 2026.

A. Hyperparameters

Table 1: Gaussian process and acquisition function hyperparameters for both test cases.

Parameter	**Lid-Driven Cavity**	**Vortex Street**
CFD Simulation		
Total Iterations n_{tot}	100	pre-calculated state + 135
Optimizer		
Initial samples n_{init}	15	15
Iterations n_{iter}	100	100
Gaussian Process kernel		
Kernel type	Matérn	Matérn
Smoothness ν	2.5	2.5
Length-scale bounds	$[0.01,\ 5.0]$	$[0.02,\ 1.0]$
Initial length-scale	0.2	0.2
Training iterations	150	150
Training learning rate	0.1	0.05
Random candidates n_{rand}	4096	4096
Noise init (normalised)	0.05	0.1
Noise lower bound	10^{-3}	10^{-4}
Acquisition function		
Type	EI	EI
ξ (exploration factor)	0.05	0.05
Exploration decay	0.9	0.9
Decay delay (iterations)	5	5
Stagnation boost patience	5	5
Stagnation boost multiplier	1.5	1.5
Max boost factor	1.0	1.0