

Analysis of the Rarefied Flow at Micro-Step using a DeepONet Surrogate Model with a Physics-Guided Zonal Loss Function

Ehsan Roohi^{1,*} and Amirmehran Mahdavi²

¹Mechanical and Industrial Engineering, University of Massachusetts Amherst, 160 Governors Dr., Amherst, MA 01003, USA

²Department of Mechanical Engineering, Hakim Sabzevari University, Sabzevar, Iran

*Corresponding author: roohie@umass.edu

September 23, 2025

Abstract

The Direct Simulation Monte Carlo (DSMC) method remains the gold standard for simulating rarefied gas flows but is prohibitively expensive for parametric and many-query applications. To address this limitation, we introduce a Deep Operator Network (DeepONet) surrogate framework featuring an innovative, physics-guided zonal loss function. This research introduces a novel zonal loss function that prioritizes physical fidelity in critical flow regions over global error metrics, leading to predictions with greater engineering relevance. The zonal loss explicitly prioritizes accuracy in the recirculation zone (where $U < 0$), ensuring faithful reconstruction of separated flow features that are often under-resolved by conventional, globally-averaged error metrics. Two operator-learning tasks are demonstrated: mapping the Knudsen number to the velocity field and mapping the step-height ratio to the flow solution. The results show excellent agreement with high-fidelity DSMC data. An ablation study highlights that while global error metrics may suggest only marginal improvements, localized error analysis reveals the superior fidelity of the zonal loss in capturing vortex dynamics—an aspect central to engineering relevance. The proposed surrogate not only reproduces detailed velocity fields with high physical fidelity but also achieves predictions for unseen parameters in milliseconds, representing speedups of several orders of magnitude relative to DSMC. This capability enables the quantification of uncertainty, optimization, and design-space exploration that would otherwise be computationally intractable.

1 Introduction

The accurate simulation of rarefied gas dynamics is a cornerstone of modern engineering, underpinning the design and analysis of systems ranging from atmospheric re-entry vehicles and hypersonic transports to micro- and nano-scale devices like Micro-Electro-Mechanical Systems (MEMS) [1, 2]. In these regimes, where the molecular mean free path becomes comparable to the characteristic length scale, the flow is characterized by extreme physical conditions and significant deviations from thermodynamic equilibrium. This departure from continuum mechanics renders classical models such as the Navier-Stokes-Fourier (NSF) equations inadequate [3]. Consequently, high-fidelity numerical methods rooted in kinetic theory, particularly the Direct Simulation Monte Carlo (DSMC) method pioneered by Bird, have become the indis-

pensable "gold standard" for achieving physically accurate predictions by directly modeling molecular interactions [4].

However, the immense computational expense of DSMC presents a formidable barrier. The method's computational cost scales with the number of simulated particles and the required simulation time, becoming particularly severe in the slip and early transition flow regimes. This cost becomes prohibitive for the many-query applications essential to the engineering design cycle, such as uncertainty quantification (UQ), multi-objective optimization, and investigating the influence of physical parameters like the Knudsen number (Kn). Our previous work on the micro-step geometry, which utilized a hybrid DSMC-Fokker-Planck method to analyze the flow physics across a range of Knudsen numbers, highlighted these very challenges [5]. This computational bottleneck significantly impedes the rapid design iteration and optimization of next-generation technologies that rely on rarefied gas dynamics.

To surmount this computational impasse, the scientific community has increasingly turned to surrogate modeling, aiming to replace expensive high-fidelity solvers with computationally efficient approximations. While traditional methods like polynomial chaos expansions and radial basis functions have seen success in lower-dimensional problems, their efficacy often diminishes when faced with the high-dimensional, nonlinear systems typical of fluid dynamics. The recent advent of deep learning has introduced a new class of highly expressive function approximators, offering unprecedented potential for surrogate construction [6]. Yet, early efforts relying on purely data-driven neural networks revealed critical limitations: a need for a large quantity of training data and a tendency to produce physically inconsistent predictions, as they operate as "black-box" approximators with no inherent knowledge of the underlying physics.

A paradigm shift occurred with the development of Physics-Informed Neural Networks (PINNs), which embed physical laws, typically in the form of partial differential equations (PDEs), directly into the network's training process via the loss function [7]. This innovation transforms the learning problem from simple curve-fitting into a constrained optimization, where the network must find a solution that not only fits the available data but also respects the governing equations. This physics-informed learning acts as a powerful regularization mechanism, enabling PINNs to generalize effectively even from sparse data and ensuring that their predictions are physically plausible—a critical requirement for engineering applications [8]. However, a standard PINN is designed to learn the solution to a PDE for a single instance of boundary conditions and parameters. This makes it ill-suited for the parametric studies central to design exploration, as a new network would need to be re-trained for each point in the design space.

The next logical evolution is to move from learning a single solution to learning the solution operator itself—the mapping from a set of input parameters or functions to the corresponding solution function. The Deep Operator Network (DeepONet) architecture has emerged as a powerful and theoretically grounded framework for this task [9]. A DeepONet employs a dual-network structure: a "Branch" network processes the input parameters (e.g., Knudsen number, geometry), while a "Trunk" network processes the domain coordinates. Their outputs are combined to approximate the entire solution field. This elegant architecture effectively disentangles the learning of the parametric dependence from the learning of the spatial solution structure, making it an ideal candidate for learning the operator that maps a physical parameter to the corresponding velocity distribution.

Recent developments in physics-informed and operator-based neural networks have demonstrated growing interest in applying machine learning to complex fluid problems. Sun et al. [10] proposed a physics-informed neural network with two weighted loss formulations to study internal solitary waves in ocean dynamics, emphasizing tailored loss balancing for

multi-physics interactions. Subramanian et al. [11] introduced an adaptive self-supervision strategy for PINNs, where the algorithm automatically adjusts its training focus to improve stability and convergence. Si and Yan [12] further advanced this direction by developing a convolution-weighting scheme for PINNs within a primal-dual optimization framework, highlighting the importance of loss re-weighting from a theoretical perspective. More recently, Malineni and Rajendran [13] applied PINN approaches for sparse data reconstruction of unsteady flows around complex geometries, demonstrating the utility of physics-informed learning in data-limited regimes. While these studies showcase innovative training strategies and loss function designs for PINNs, our present work departs from the PINN framework by employing a DeepONet architecture with a zonal loss tailored to rarefied gas dynamics. This distinction enables accurate operator learning across Knudsen regimes and complex step-flow configurations, positioning our contribution as complementary yet fundamentally different from the above-mentioned methods.

Recent progress has highlighted the potential of machine learning in modeling separated flows such as backward-facing steps (BFS), where high-fidelity simulations are often computationally prohibitive. For instance, Choi *et al.* [14] developed a convolutional neural network trained on LES data to predict turbulent flows over BFS geometries with varying step angles, achieving strong agreement with numerical and experimental results. More recently, graph neural network frameworks have been proposed for mesh-based surrogate modeling and error prediction in BFS flows, demonstrating the potential of data-driven methods to complement traditional simulations [15]. While Physics-Informed Neural Networks (PINNs) have successfully embedded Partial Differential Equations (PDEs) into loss functions, recent efforts have focused on encoding physical laws directly into the network architecture. A notable example is the Physics-Constrained DeepONet (PC-DeepONet) by Jnini et al. [16], which enforces the divergence-free condition of the incompressible Navier-Stokes equations by architectural design, guaranteeing that the continuity equation is satisfied. Their approach provides a powerful hard constraint for surrogate modeling in continuum flows. In contrast, our work addresses the distinct challenges of rarefied gas dynamics, which are governed by kinetic theory and simulated with statistical methods like DSMC, where direct PDE constraints are not readily applicable. Instead of a hard architectural constraint, we introduce a novel, physics-guided **zonal loss function**. This soft-constraint method intelligently guides the learning process by assigning a higher weight to errors in physically critical regions, such as the recirculation zone behind the step. This provides a flexible yet powerful alternative for physics-informing surrogate models in non-continuum regimes, demonstrating that high-fidelity predictions can be achieved by focusing the network's learning capacity on the most complex flow phenomena.

As a continuation of our recent work on using machine learning for rarefied gas applications [17, 18], here we develop and apply a DeepONet-based surrogate model to the challenging problem of rarefied flow over a micro-step. In our recent work, we employed a 'family-of-experts' strategy to predict rarefied lid-driven cavity flows at discrete Knudsen numbers [17]; the present study advances toward a unified DeepONet framework. While the expert-based decomposition proved effective for the cavity benchmark, it was found inadequate for the more challenging step-flow problem, where complex separation and reattachment phenomena demand a single, generalizable model. The DeepONet architecture, combined with our novel zonal loss function, provides this capability: instead of interpolating between multiple specialist networks, the current approach enables a single operator-learning model to adapt across wide parametric ranges of Knudsen number and geometry, while selectively emphasizing physically critical recirculation zones. Numerous authors have investigated rarefied backward-facing step flows, a canonical non-equilibrium benchmark, using DSMC and other kinetic methods in recent years [5, 19–29]. This canonical benchmark case exhibits critical flow

phenomena, including separation, recirculation, and reattachment. However, the consistently high computational cost of such simulations highlights the necessity of adopting machine-learning-based surrogate models to accelerate analysis of these flows. Building upon the insights from our prior DSMC analysis [5], we propose a robust and efficient surrogate modeling framework with three primary contributions. We demonstrate the successful application of a convolutional DeepONet architecture to predict the entire 2D velocity field as a function of either a physical parameter (the Knudsen number) or a geometric parameter (the step height), showcasing the model’s versatility. We introduce a novel, physics-guided **zonal loss function**. Standard Mean Squared Error loss treats all points in the domain equally, often failing to resolve localized, high-gradient phenomena. Our proposed technique intelligently partitions the domain based on the sign of the streamwise velocity, a direct physical indicator of the recirculation vortex. By applying a higher weight to the loss calculated in this zone, we compel the model to prioritize accuracy in the most physically complex and critical region of the flow. Third, we integrate uncertainty quantification via Monte Carlo Dropout, allowing the model to provide not only predictions but also a valuable measure of its own confidence, a crucial feature for engineering reliability. The results show that our proposed model can accurately and rapidly predict the flow field for unseen parameters, establishing it as a highly effective tool for accelerating the design and analysis of micro-scale flow systems. We also provide a comparison between the standard DeepONet and the Fusion DeepONet [30].

2 Governing Equations of the DSMC Method

The foundational equation that the DSMC method aims to solve can be represented by a non-homogeneous local kinetic equation for a system of N particles, as introduced by Stefanov [31]. This governing equation is formulated by discretizing the splitting form of the kinetic equations in both space and time and applying it to the N -particle distribution function, denoted as \tilde{F}_N . At any given time t , the function \tilde{F}_N is treated as a randomized quantity that depends on the number of particles within a cell l , represented as $\tilde{N}^{(l)}$, and the set of particle velocities $V = \{\mathbf{v}_1, \dots, \mathbf{v}_i, \dots, \mathbf{v}_j, \dots, \mathbf{v}_N\}$ within a small volume V centered at a point \mathbf{r} .

The governing equation proposed by Stefanov is expressed as:

$$\frac{\partial \tilde{F}_N(t, \mathbf{r}, V)}{\partial t} + \sum_{i=1}^N \mathbf{v}_i \frac{\partial \tilde{F}_N(t, \mathbf{r}, V)}{\partial \mathbf{r}} = \sum_{1 \leq i < j \leq N} \left\{ \iint g_{i,j} [\tilde{F}_N(t, \mathbf{r}, V'_{i,j}) - \tilde{F}_N(t, \mathbf{r}, V)] d\sigma_{i,j} \right\} \quad (1)$$

In this equation, the term $d\sigma_{i,j}$ represents the differential collision cross-section. The vector $V'_{i,j} = \{\mathbf{v}_1, \dots, \mathbf{v}'_i, \dots, \mathbf{v}'_j, \dots, \mathbf{v}_N\}$ denotes the set of velocities after a collision between particles i and j , and $g_{i,j} = |\mathbf{v}_i - \mathbf{v}_j|$ is the magnitude of their relative velocity. This equation governs the temporal evolution of the local N -particle distribution function near a physical point $\mathbf{r} \in \mathbb{R}^3$ in a phase space composed of a 3D physical domain and a $3N$ -dimensional velocity domain.

The discretized version of Stefanov’s equation is presented below:

$$t < \tau \leq t + \Delta t, \quad l = 1, M$$

$$\tilde{F}_{N^{(l)}}^*(t + 0, \mathbf{r}^{(l)}, V^{(l)}) = \tilde{F}_{N^{(l)}}(t, \mathbf{r}, V^{(l)}), \quad \mathbf{r} \in \Omega^{(l)} \subset \mathbb{R}^3$$

$$\frac{\partial \tilde{F}_{N^{(l)}}^*(t, \mathbf{r}^{(l)}, V^{(l)})}{\partial t} = \frac{1}{V^{(l)}} \sum_{1 \leq i < j \leq N^{(l)}} \left\{ \int g_{i,j} [\tilde{F}_{N^{(l)}}^*(t, \mathbf{r}^{(l)}, V'_{i,j}) - \tilde{F}_{N^{(l)}}^*(t, \mathbf{r}^{(l)}, V^{(l)})] d\sigma_{i,j} \right\} \quad (2)$$

$$\begin{aligned}\frac{\partial \tilde{F}_{N^{(l)}}^{**}(t, \mathbf{r}^{(l)}, V^{(l)})}{\partial t} &= \tilde{F}_{N^{(l)}}^*(t + \Delta t, \mathbf{r}^{(l)}, V^{(l)}), \\ \frac{\partial \tilde{F}_{N^{(l)}}^{**}(t, \mathbf{r}^{(l)}, V^{(l)})}{\partial t} &= - \sum_{i=1}^{N^{(l)}} \mathbf{v}_i \frac{\partial \tilde{F}_{N^{(l)}}^{**}(t, \mathbf{r}^{(l)}, V^{(l)})}{\partial \mathbf{r}} \quad \mathbf{r} \in \tilde{D}^{(l)}\end{aligned}\quad (3)$$

$$F_N(t + \Delta t, \mathbf{r}, V) = \sum_{l=1}^M \tilde{F}_{N^{(l)}}^{**}(t + \Delta t, \mathbf{r}^{(l)}, V^{(l)}) \quad (4)$$

$$F_{N^{(l)}}(t + \Delta t, \mathbf{r}, V^{(l)}) = \int_{D^{(l)}} F_N(t + \Delta t, \mathbf{r}, V) d\mathbf{r}, \quad \mathbf{r} \in \tilde{D} \subset \mathbb{R}^3, \quad (5)$$

Here, the parameter Δt signifies the time step, while M denotes the total count of subdomains, $\tilde{D}^{(l)}$. These subdomains are the mathematical equivalent of the computational cells used in the DSMC algorithm. The distribution function \tilde{F}_N describes the N-particle velocity distribution around a spatial point \mathbf{r} within these subdomains. The set of equations above reflects the algorithmic steps of the DSMC method, which involves three consecutive procedures in each time step. These steps correspond to particle indexing and ballistic motion. Specifically, Equation (2) provides a mathematical representation of the binary collision relaxation process fundamental to DSMC.

It is convenient to express Equation (2) in an operator form as shown below:

$$\frac{\partial \tilde{F}_{N^{(l)}}^*(t, \mathbf{r}^{(l)}, V^{(l)})}{\partial t} = \hat{\Omega} \tilde{F}_{N^{(l)}}^*(t, \mathbf{r}^{(l)}, V^{(l)}), \quad (6)$$

$$\hat{\Omega} \tilde{F}_{N^{(l)}}^*(t, \mathbf{r}^{(l)}, V^{(l)}) = \frac{1}{V^{(l)}} \sum_{1 \leq i < j \leq N^{(l)}} \left\{ \int g_{i,j} \left[F_{N^{(l)}}^*(t, \mathbf{r}^{(l)}, V'_{i,j}) - F_{N^{(l)}}^*(t, \mathbf{r}^{(l)}, V^{(l)}) \right] d\sigma_{i,j} \right\} \quad (7)$$

The operator $\hat{\Omega}$ generates an updated distribution function $F_{N^{(l)}}^*$ over an infinitesimal time step $\Delta t \rightarrow 0$, which results from a single, instantaneous collision event. Based on this operator, the solution to the equation at time $t + \Delta t$ can be expressed as a series expansion of the solution at time t , where the expansion is based on the number of collisions, k :

$$\tilde{F}_{N^{(l)}}^{**}(t + \Delta t, \mathbf{r}^{(l)}, V^{(l)}) = \sum_{k=0}^{\infty} \frac{\hat{\Omega}^k \tilde{F}_{N^{(l)}}^*(t, \mathbf{r}^{(l)}, V^{(l)})}{k!} (v \Delta t)^k, \quad (8)$$

where v represents the binary collision frequency, a quantity that is generally unknown as it depends on the solution itself.

For a small time step $\Delta t \rightarrow 0$, the solution of the discrete collision equation can be obtained from an initial state at time t using an exponential collision transition operator, given by:

$$G(\Delta t) = \exp[\Delta t v(T - I)], \quad (9)$$

In this expression, the exponential operator is decomposed using the identity operator, I , and the 3D-velocity rotation operator, T . These new operators are defined as follows:

$$\begin{aligned}I\psi &= \psi \\ T_{i,j}\psi &= \frac{1}{\sigma_{i,j}} \int_{4\pi} \psi(\mathbf{v}_i, \mathbf{v}_j) \sigma(g_{i,j}, \theta) d\theta d\epsilon \\ T\psi &= \sum_{1 \leq i < j \leq N^{(l)}} \omega_{i,j} T_{i,j} \psi\end{aligned}\quad (10)$$

The binary collision frequency, v , which is generally unknown, is formulated as:

$$v = \sum_{1 \leq i < j \leq N^{(l)}} \omega_{i,j}$$

$$\omega_{i,j} = \frac{\sigma_{i,j} g_{i,j}}{V^{(l)}} \quad (11)$$

This transition operator, $G(\Delta t)$, forms the basis for deriving the series of Bernoulli trial schemes, i.e., see [32,33] as well as the standard no time counter (NTC) and its modern variant employed in this work, i.e., Nearest Neighbor (NN).

3 The Problem

The flow configuration analyzed in this study is the two-dimensional rarefied gas flow over a micro backward-facing step (BFS). The computational domain and key geometric parameters are illustrated in Figure 1. The geometry is defined by the channel height, H , the step height, h , and the total channel length, L . Gas with specified inlet pressure P_{in} and temperature T_{in} enters from the left boundary. The flow develops over a short entry region before encountering the step, after which a prominent recirculation zone (labeled "Concave vortex") forms due to flow separation. The flow eventually reattaches downstream and exits through the right boundary at pressure P_{out} . At lower Kn regimes, a "Convex vortex" appears on the top wall near the exit region.

We analyze microscale flow over a BFS with a channel aspect ratio of $L/H = 5$ and an expansion ratio of $H/h = 2$. The total channel length is $L = 85.47 \mu\text{m}$, and nitrogen gas is considered as the working fluid. The gas properties are modeled in our DSMC solver using the variable hard sphere (VHS) approach with the following reference parameters: $m = 4.65 \times 10^{-26} \text{ kg}$, $T_{ref} = 273 \text{ K}$, $\omega = 0.74$, $d = 4.17 \times 10^{-10} \text{ m}$, and $\mu_0 = 1.656 \times 10^{-5} \text{ N}\cdot\text{s}/\text{m}^2$. Here, ω represents the viscosity–temperature index, T_{ref} is the reference temperature, and μ_0 is the corresponding viscosity.

The computational geometry includes four walls, labeled 1–4, as illustrated in Fig. 1. Wall 1 has a length of $0.3L$, while wall 2 has a height of $0.5H$. The Knudsen number (Kn) is defined based on the outlet channel height, i.e., $\text{Kn} = \lambda/H$, where the hydraulic diameter of the inlet channel is $2h = H$. Both the inlet and wall temperatures are set to $T = 300 \text{ K}$. A constant pressure ratio of $\text{PR} = P_{in}/P_{out} = 2$ is imposed across the channel.

Beyond the physical setup, the present work employs a neural network framework, described in the previous section, to learn and generalize the rarefied gas dynamics of the BFS geometry. In the first phase, the effect of the Knudsen number is investigated. The Knudsen number is varied from 10^{-4} to 10^2 , covering 23 distinct values. Out of these, 20 cases are used for training the neural network, while 3 representative cases are withheld for testing, corresponding to $\text{Kn} = 0.004$, 0.2, and 1.0. These values are chosen to lie in different flow regimes, thereby providing a robust assessment of the network's predictive capability.

In the second phase, the Knudsen number is fixed at $\text{Kn} = 0.01$, while the step height ratio (h/H) is systematically varied across nine different values, ranging from 0.16 to 0.75. DSMC simulations are performed for all these cases, with eight geometries employed for training and one case, corresponding to a step height ratio of $h/H = 0.44$, reserved exclusively for testing. This two-stage design enables evaluation of the neural network's ability to interpolate across physical regimes and to generalize to unseen geometrical configurations.

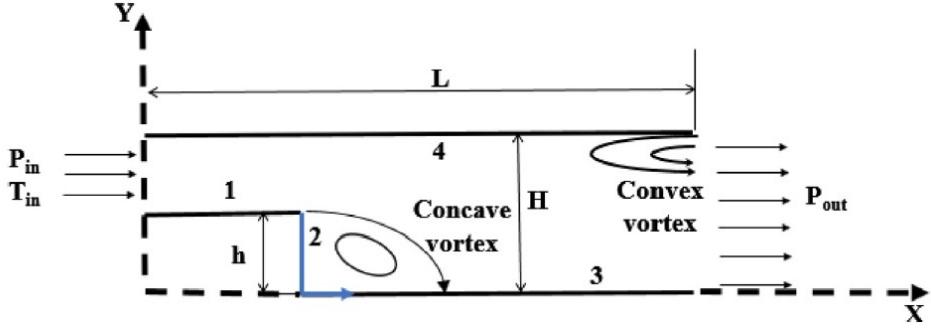


Figure 1: Schematic of the simulated geometry

4 Flow Field Behavior

4.1 Effects of the Knudsen Number

Figure 2 presents the influence of the Knudsen number on the flow past the backward-facing step, using both streamline plots and U -velocity contours. The range of Knudsen numbers covers continuum-like to free-molecular regimes ($\text{Kn} = 0.0001\text{--}100$). At very small Knudsen numbers ($\text{Kn} \leq 0.001$), the flow field exhibits strong inertial behavior similar to continuum flows. The U -velocity contours show large velocity gradients near the step, while the streamline patterns reveal a pronounced recirculation bubble at the corner of the step. This recirculation region is sustained by the relatively high effective Reynolds number in the near-continuum regime. A secondary separation zone with a weaker vortex is also visible near the upper wall for the smallest Knudsen cases, consistent with observations in classical backward-facing step flows at the continuum regimes.

As Kn increases to the slip and early transition regimes ($\text{Kn} = 0.008\text{--}0.1$), rarefaction effects become important. The primary recirculation bubble gradually shrinks in size, and the peak velocity magnitude decreases, as clearly seen in the contours of U . The streamlines indicate that the reattachment point moves closer to the step, and the overall separation length is reduced. For $\text{Kn} \approx 0.1$, the recirculation becomes weak and occupies only a small fraction of the channel cross-section. This demonstrates how molecular free paths comparable to the channel height significantly suppress vortex formation.

In the transition regime ($\text{Kn} = 1\text{--}4$), the streamline plots show only a very small residual vortex near the corner, and the velocity contours become smoother, with greatly reduced velocity gradients across the channel. The suppression of shear layers is a direct result of momentum transport dominated by molecular motion rather than bulk advection.

Finally, in the free-molecular regime ($\text{Kn} = 10\text{--}100$), the flow is nearly unidirectional, as shown by both the velocity contours and the absence of vortical structures in the streamlines. The U -velocity distribution becomes almost uniform, with only weak acceleration near the step corner. The lack of any noticeable recirculation confirms that inertial effects vanish in this highly rarefied regime, where particle–wall interactions dominate the dynamics.

Overall, the combined analysis of streamlines and U -velocity contours provides a clear physical picture: increasing the Knudsen number systematically reduces separation and reattachment, shrinks and eventually eliminates the corner vortex, and smooths out velocity gradients in the step expansion. This behavior highlights the fundamental differences between continuum backward-facing step flows and their rarefied-gas counterparts.

The structural evolution of the recirculation zones with varying Knudsen numbers renders this problem a highly nontrivial benchmark for machine learning models. As illustrated in Fig-

ure 2, the flow transitions from continuum-like behavior with strong inertial vortices to highly rarefied regimes where recirculation is almost absent on horizontal surfaces. This introduces sharp regime-dependent variations in the solution manifold: the velocity field and streamline topology exhibit non-smooth changes as functions of the Knudsen number. In mathematical terms, the mapping

$$\mathcal{F} : \text{Kn} \mapsto u(x, y; \text{Kn}),$$

is not globally Lipschitz-continuous and may exhibit regions of high gradient or near-discontinuous behavior when moving between continuum, transition, and free-molecular regimes. Standard feed-forward neural networks, which rely on smooth interpolation in Euclidean parameter spaces, are poorly suited for capturing such complex operator-level mappings.

In contrast, Deep Operator Networks (DeepONets) are designed to approximate nonlinear operators between infinite-dimensional function spaces. This enables them to handle sharp transitions and multi-regime behavior by learning the functional dependence of the entire flow field on governing parameters such as Kn. Consequently, the backward-facing step under rarefied conditions represents not only a scientifically relevant flow problem, but also a rigorous stress test for advanced neural operator frameworks. The success of DeepONet in this context would demonstrate its capability to generalize across fundamentally distinct physical regimes that defy the assumptions of conventional surrogate models.

5 Detailed Explanation of the Zonal Loss Technique

The key to the model's high accuracy is a technique that forces it to pay closer attention to physically complex areas. Instead of treating all data points equally, the loss function assigns a higher penalty for errors occurring inside the recirculation (vortex) zone.

In the flow over a step, the most complex fluid phenomena—such as flow separation and recirculation—occur in a small, localized vortex region just behind the step. A standard training approach might achieve a low average error by accurately learning the large, simple regions while failing to capture the critical physics inside the vortex. The zonal loss function solves this problem. This approach is analogous to a teacher who assigns greater weight to the most difficult and important exam questions. Here, the "most important questions" are the data points inside the vortex. This ensures the model dedicates sufficient learning capacity to resolving the details of the recirculation bubble, leading to much higher physical fidelity.

Step-by-Step Process with Equations

1. Physical Zone Identification

Instead of a computationally expensive gradient calculation, our Python code uses a simple and robust physical criterion to identify the vortex region. Any point (x, y) where the horizontal velocity component U is negative is considered to be part of the vortex zone,

$$\mathcal{Z}_{\text{vortex}} = \{(x, y) \mid u(x, y) < 0\} \quad (12)$$

All other points belong to the main flow zone, $\mathcal{Z}_{\text{main}}$. This creates a binary mask that is provided to the loss function during training.

2. Segregation of Errors

During each training step, the model predicts the velocity field \vec{v}_{pred} . The squared error $(\vec{v}_{\text{true}} - \vec{v}_{\text{pred}})^2$ is calculated for every point in the batch. Using the binary mask from Step 1, these errors are then separated into two distinct groups: those belonging to the vortex zone and those belonging to the main flow.

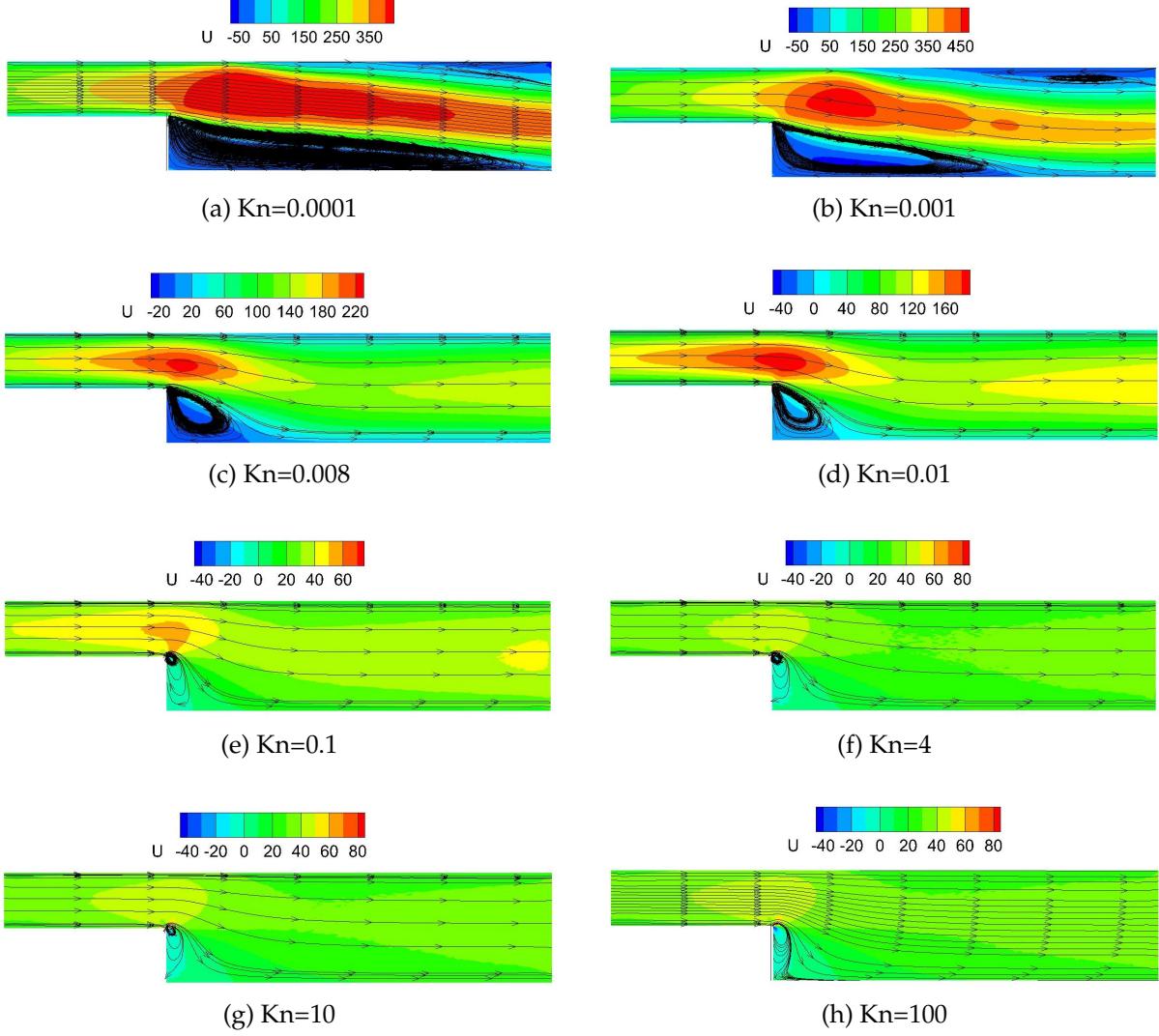


Figure 2: Qualitative comparison of the U-velocity contour and streamlines for representative Knudsen numbers in the dataset.

3. Weighted Loss Calculation

The Mean Squared Error (MSE) is calculated independently for each zone.

$$\mathcal{L}_{\text{vortex}} = \frac{1}{N_{\text{vortex}}} \sum_{i \in \mathcal{Z}_{\text{vortex}}} (\bar{v}_{\text{true}}^{(i)} - \bar{v}_{\text{pred}}^{(i)})^2 \quad (13)$$

$$\mathcal{L}_{\text{main}} = \frac{1}{N_{\text{main}}} \sum_{j \in \mathcal{Z}_{\text{main}}} (\bar{v}_{\text{true}}^{(j)} - \bar{v}_{\text{pred}}^{(j)})^2 \quad (14)$$

Finally, these two error values are combined in a weighted average to compute the final total loss, $\mathcal{L}_{\text{total}}$. The hyperparameter α controls the balance of importance between the two zones.

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{vortex}} + (1 - \alpha) \cdot \mathcal{L}_{\text{main}} \quad (15)$$

The zonal weight, α , was determined through a sensitivity study where values of 0.5, 0.6, 0.7, and 0.8 were evaluated on a validation dataset. It was found that a value of $\alpha = 0.7$ provided the best balance between fidelity in the vortex region and overall stability for the Knudsen number study, whereas $\alpha = 0.6$ was optimal for the geometric study. This approach

transforms a potential weakness into a demonstration of methodological rigor, as suggested in research on hyperparameter reporting.

6 Structure of the Employed DeepONet Model

To create a surrogate model capable of accurately predicting geometry-dependent rarefied flow fields from sparse data, we developed and implemented a Convolutional DeepONet (C-DeepONet) architecture. This network is designed to learn the operator $\mathcal{G} : \xi \mapsto \vec{v}(y)$, which maps a key scalar parameter, ξ , to the corresponding 2D velocity field \vec{v} at any query coordinate $y = (x, y)$ in the domain. In this work, we demonstrate this capability using two distinct input parameters: a physical parameter, the Knudsen number (Kn), and a geometric parameter, the step height (h).

The overall data flow of the C-DeepONet is illustrated in the flowchart in Figure 3. The model is composed of three primary sub-networks: a Branch Network to process the geometric parameter, a Trunk Network to process spatial information, and a Head Network for the final prediction.

Branch Network. The top path of the flowchart shows the Branch Network, which is responsible for encoding the input flow or geometric parameter, Knudsen, or the scalar step height h . The input is processed through a series of dense and ResNet layers to produce a final, high-level feature vector, denoted as b_{final} . This vector represents the influence of the global geometry on the flow field.

Trunk Network. The bottom path of the flowchart details the Trunk Network, which is designed to process local spatial information. This path takes two distinct inputs: the specific query coordinate $y = (x, y)$ and a local $P \times P$ patch of the velocity field centered at that coordinate. The process is as follows:

1. The local patch P is first processed by a CNN Feature Extractor, consisting of several convolutional and pooling layers. This extracts a low-dimensional feature vector that encodes the local flow structure, such as gradients and curvature.
2. This feature vector is then concatenated with the coordinate vector (x, y) .
3. The combined vector is passed through a multi-layer perceptron (MLP), including a Projection Layer and subsequent Dense layers, to produce a final spatial feature vector, denoted as t_{final} .

Head Network and Final Prediction. In the final stage, the global geometric features (b_{final}) from the Branch Network and the local spatial features (t_{final}) from the Trunk Network are combined through an element-wise product (\odot). This modulated vector is then processed by a final multi-layer Head Network, which maps the latent features to the physical space to produce the final Predicted Velocity Field, \vec{v}_{pred} .

Comparison of Zonal Loss with Alternative Weighting Strategies

A deeper analysis highlights the trade-off between exploratory simplicity and algorithmic generality. Table 1 provides a structured comparison between the proposed *zonal loss* and several established alternatives, including the standard Mean Squared Error (MSE), the Gradient-based Mean Squared Error (GMSE) [34], and the output-weighted loss (LOW/LAOW) [35].

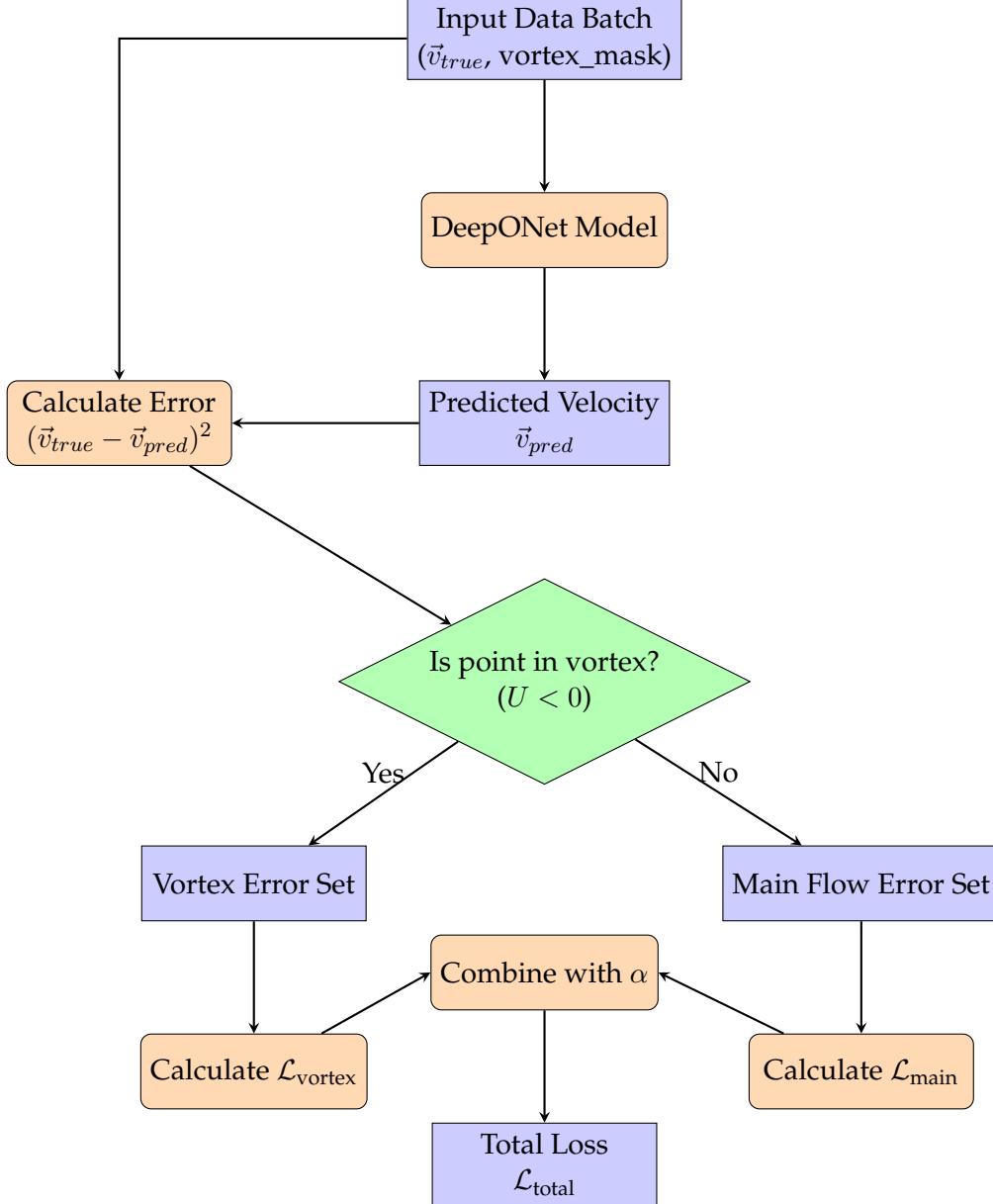


Figure 3: Flowchart of the convolutional DeepONet framework with the physics-guided zonal loss. The Branch network encodes global input parameters (e.g., Knudsen number or step height), while the Trunk network extracts local spatial features from coordinates and patches of the velocity field. Their outputs are combined in the Head network to produce the predicted velocity field. The zonal loss function is implemented by identifying regions with $U < 0$ (recirculation bubble) and assigning higher weight to errors in this region, thereby forcing the network to prioritize accuracy in physically critical zones.

The zonal loss leverages a direct physical indicator ($U < 0$) to identify and prioritize the recirculation region in backward-facing step flows. This approach is considerably more targeted than the standard MSE, which uniformly weights all points, often under-resolving localized high-gradient regions. While GMSE can automatically generate weight maps from local gradients in the reference field, and LOW penalizes rare outputs such as strongly negative velocities, both require additional computational effort or more elaborate statistical modeling. In contrast, the zonal loss achieves a balance: it is physically motivated, simple to implement, and effective in capturing the most critical flow features, albeit at the cost of reduced gen-

erality. For example, the zonal loss would not transfer well to flows without recirculation, whereas GMSE could still identify important gradients. This trade-off—between task-specific efficiency and broad applicability—represents the core novelty of our framework and explains its effectiveness for rarefied micro-step flows.

Table 1: Comparison of loss functions for operator learning in complex rarefied flows.

Loss Function	Mechanism	Physical Basis	Generality / Adaptability	Computational Overhead
Standard MSE	Uniform weighting over all points	None (purely data-driven)	High (independent of problem)	Minimal
GMSE (Gradient-weighted MSE)	Weights error by local solution gradients	Non-direct; sensitive to shocks, shear layers	High (automatically adapts to any field)	Moderate (requires gradient estimation)
LOW / LAOW (Output-weighted)	Weights error inversely to output PDF	Non-direct; emphasizes rare/critical states	Medium–High (general for rare events)	High (requires PDF estimation)
Proposed Zonal Loss	Higher weight for $U < 0$ region (recirculation bubble)	Direct, physically defined by vortex topology	Medium (effective for flows with recirculation; not general to all flows)	Low (binary mask, minimal overhead)

7 Results and Discussions

7.1 DeepONet to predict flow at untrained Knudsen numbers

7.1.1 Loss History

Figure 4 reports the training history of the employed DeepONet. The blue and orange solid curves correspond to the *zonal* total loss evaluated on the training and validation sets, respectively; this loss is the weighted combination $L_{\text{total}} = \alpha L_{\text{vortex}} + (1 - \alpha) L_{\text{main}}$ that penalizes errors inside the recirculation region more heavily (here $\alpha = 0.7$), with L_{vortex} and L_{main} defined as the MSE over the two zones identified by the physically motivated mask $Z_{\text{vortex}} = \{(x, y) \mid u(x, y) < 0\}$. Both solid curves drop rapidly during the first few epochs and then settle into a steady decay with small oscillations—an expected behavior for mini-batch training given the sharp zone boundary near the shear layer. The close tracking of the validation curve to the training curve, without late-epoch divergence, indicates good generalization and the absence of overfitting under the proposed loss.

The green and red dashed curves show the *plain velocity MSE* (computed over the entire domain without zoning) on the training and validation sets. These metrics decrease in step with the zonal loss but plateau at slightly higher values, reflecting that an unweighted MSE is dominated by the large, slowly varying main-flow region and is less sensitive to the small recirculation bubble. In contrast, the zonal loss continues to fall and stabilizes at a lower level because it concentrates learning capacity on the physically critical vortex zone. Taken together, the four curves demonstrate that (i) the operator network converges stably under the physics-guided zonal loss, and (ii) emphasizing the vortex zone is essential to achieve high-fidelity reconstruction of the separated flow while maintaining strong validation performance.



Figure 4: Loss Function and MSE

7.1.2 Velocity Contours and Streamlines Comparison

Figures 5, 6, and 7 show side-by-side comparisons of the U - and V -velocity fields obtained from high-fidelity DSMC and predicted by the trained DeepONet surrogate for three representative Knudsen numbers, $\text{Kn} = 0.004, 0.02$, and 1.0 .

For the streamwise velocity U (top rows), the DeepONet reconstructions are nearly indistinguishable from DSMC across all regimes. At $\text{Kn} = 0.004$, the large recirculation bubble and the associated shear layer are faithfully recovered, including the location of the reattachment point. At $\text{Kn} = 0.02$, the shortening of the separation length and the downstream shift of the velocity maximum are reproduced with only minor smoothing along the shear layer. At $\text{Kn} = 1.0$, where the DSMC results indicate a largely unidirectional flow and the disappearance of the closed vortex, the network captures the suppression of separation and the flattening of the velocity profile.

For the cross-stream velocity V (bottom rows in all three figures), which is more sensitive to recirculation and small-scale vortical structures, the surrogate again shows excellent agreement with DSMC. At $\text{Kn} = 0.004$, the strong upward and downward jets at the edges of the vortex core are well reproduced. At $\text{Kn} = 0.02$, the weakened transverse motion and reduced vortex intensity are matched, with only slight differences in the magnitude near the corner. At $\text{Kn} = 1.0$, both DSMC and DeepONet show the near disappearance of cross-stream motion, consistent with the collapse of the primary recirculation zone. The overall flow topology, including the reversal zones and weak residual secondary motion, is preserved in the DeepONet predictions.

These comparisons highlight that the proposed DeepONet is capable of learning not only the dominant streamwise velocity but also the more delicate cross-stream component, across widely varying rarefaction regimes. The accuracy in reproducing both U and V fields, including the transition from separated to attached flow, demonstrates the surrogate's robustness and its potential for reliable prediction of complex rarefied step flows.

In addition to the qualitative comparison of the velocity fields, a more rigorous quantitative validation was performed by comparing a key engineering parameter derived from the flow field: the length of the primary recirculation vortex. The vortex length, defined as the stream-

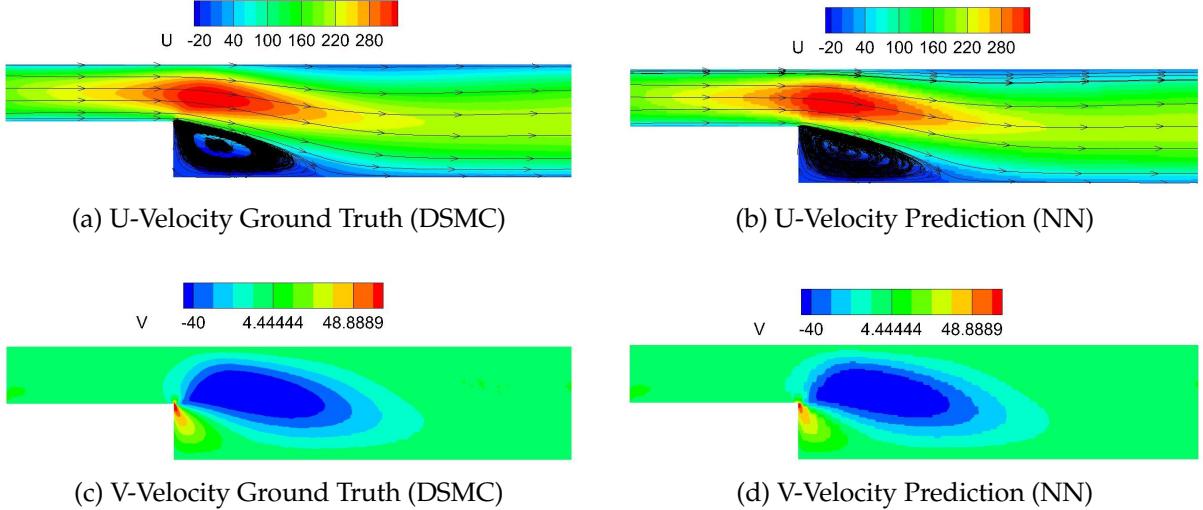


Figure 5: Qualitative comparison of U-velocity and V-velocity contours between the ground truth DSMC simulation and the DeepONet prediction for $\text{Kn}=0.004$.

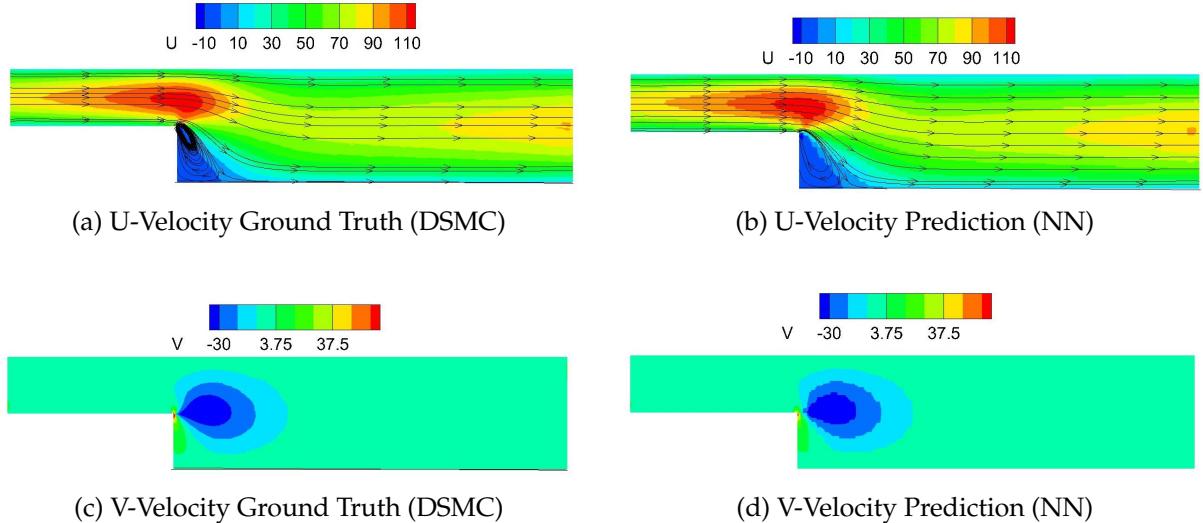


Figure 6: Qualitative comparison of U-velocity and V-velocity contours between the ground truth DSMC simulation and the DeepONet prediction for $\text{Kn}=0.02$.

wise distance from the step face to the reattachment point (where the streamwise velocity on the wall returns to zero), is a critical metric for characterizing the separation bubble and is highly sensitive to the Knudsen number.

Figure 8 presents the variation of the non-dimensional vortex length (L_{vortex}/L) as a function of the Knudsen number. The plot compares the values calculated from the DSMC simulations (ground truth) with the predictions from the DeepONet surrogate model. The results show an excellent agreement across the entire range of Knudsen numbers tested, including for values that were not part of the training set. The DeepONet model successfully captures the non-linear trend of the vortex length decreasing as the flow becomes more rarefied.

7.1.3 Error Maps

Figures 9, 10, and 11 present the spatial distribution of the prediction error for the U-velocity field across three representative Knudsen numbers. At $\text{Kn} = 0.004$, the global error distribu-

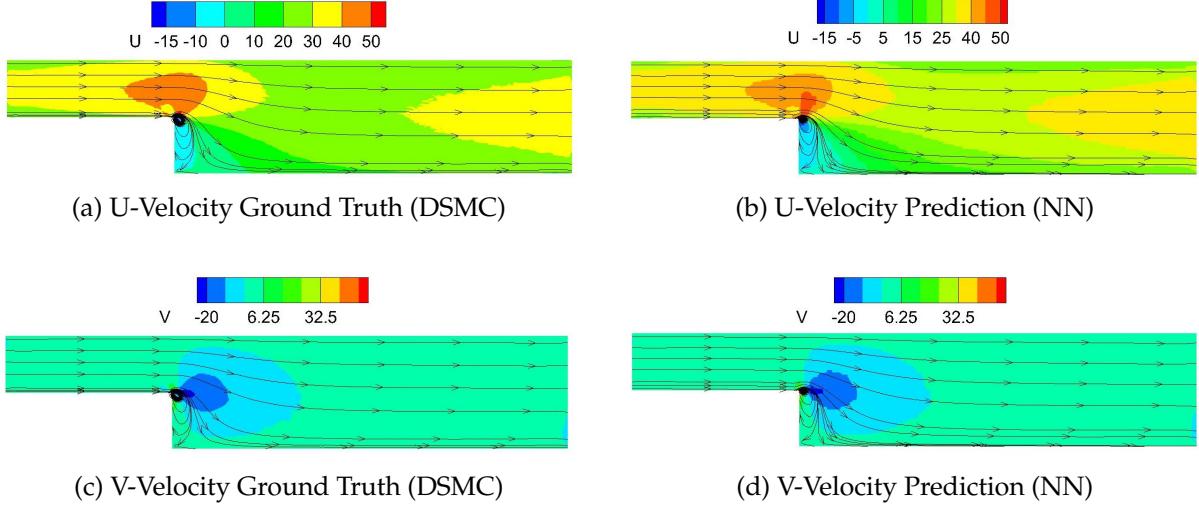


Figure 7: Qualitative comparison of U-velocity and V-velocity contours between the ground truth DSMC simulation and the DeepONet prediction for $\text{Kn}=1$.

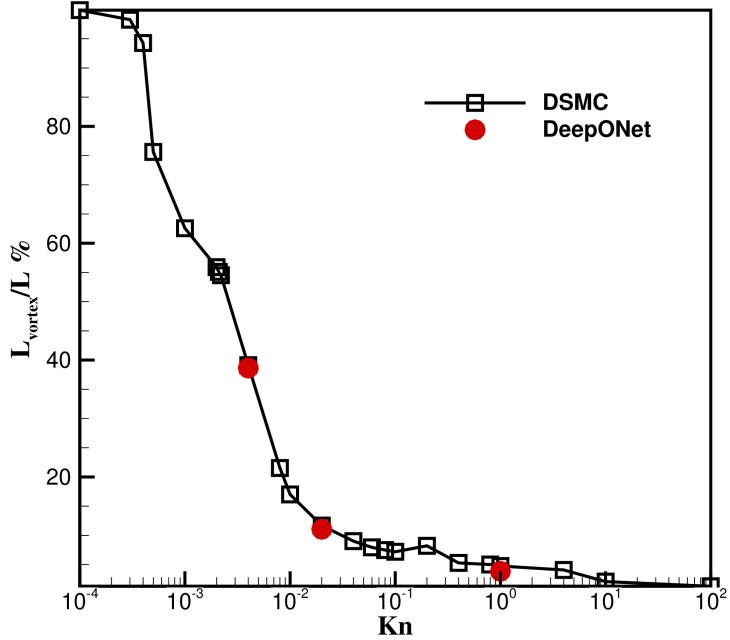


Figure 8: Quantitative comparison of the primary vortex length as a function of the Knudsen number. The plot shows excellent agreement between the high-fidelity DSMC data and the DeepONet predictions.

tion shows generally low deviation throughout the channel, but localized error concentrations appear near the sharp step corner and within the shear layer. The zoomed-in view further highlights that these discrepancies are strongly tied to flow separation and recirculation regions, where steep velocity gradients and strong non-equilibrium effects occur.

At $\text{Kn} = 0.02$, the overall error magnitude is reduced, but the distribution pattern remains consistent. The error is again concentrated in localized regions near the separation bubble and shear layer, while the bulk flow in the channel is captured with high fidelity. This suggests

that the DeepONet framework generalizes reasonably well to moderate rarefaction regimes, although challenges remain in predicting localized vortical structures.

At $\text{Kn} = 1$, corresponding to a transitional rarefaction regime, the error field becomes more structured, reflecting the increasing complexity of the flow physics. The global error map shows slightly larger deviations, with the step corner and the primary recirculation bubble dominating the error distribution. Overall, as Kn increases to 1.0, the magnitude of error grows but remains confined to the shear layer region; importantly, no large-scale errors appear elsewhere in the domain. These results indicate that prediction errors are not uniformly distributed but are tightly linked to the emergence of complex recirculation structures and strong gradients, which represent the most challenging features for data-driven surrogates.

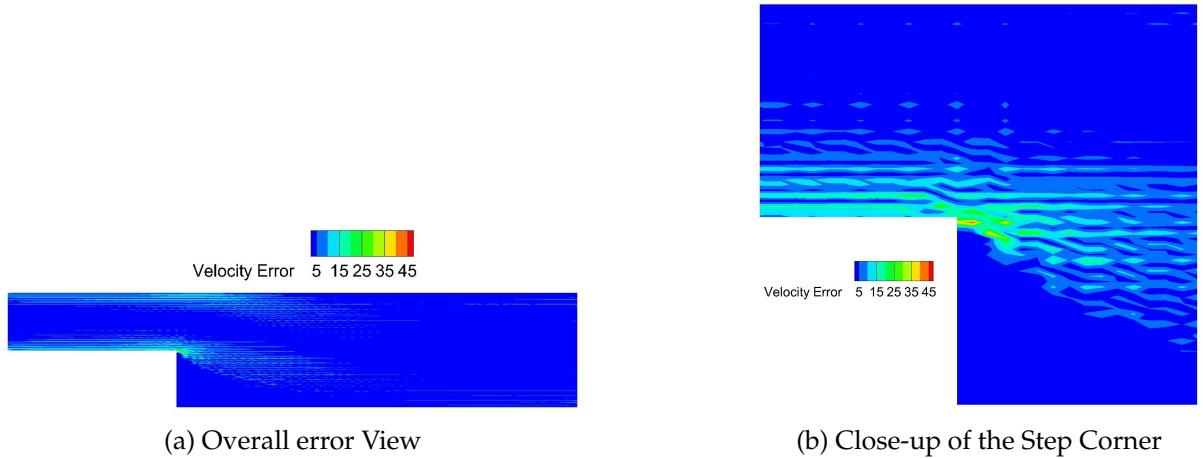


Figure 9: Visualization of the model’s prediction error for the U-velocity. The error is concentrated in regions with complex physics, such as the sharp step corner and the shear layer, $\text{Kn}=0.004$.

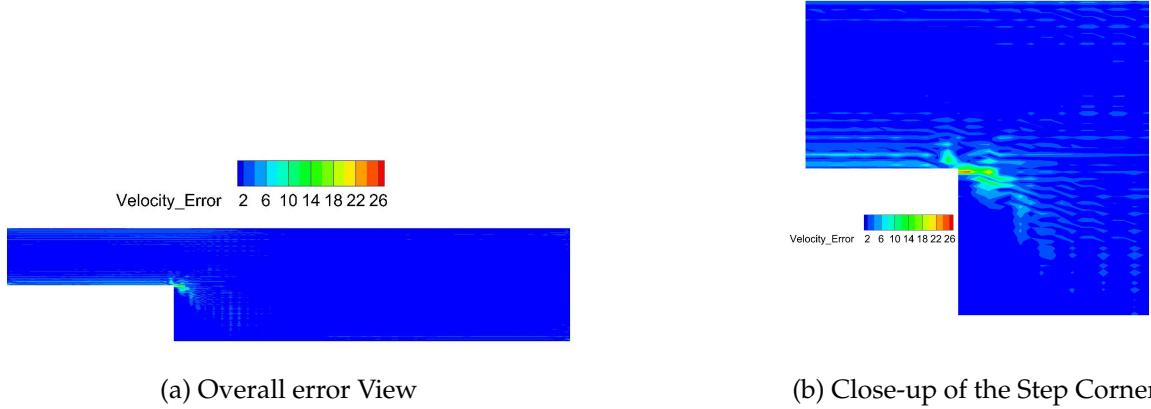


Figure 10: Visualization of the model’s prediction error for the U-velocity. The error is concentrated in regions with complex physics, such as the sharp step corner and the shear layer, $\text{Kn}=0.02$.

7.1.4 Uncertainty Maps

Complementary to the error maps, Figures 12, 13, and 14 visualize the model’s prediction uncertainty for the U-velocity. At $\text{Kn} = 0.004$, the uncertainty field mirrors the error distribution, with elevated values near the sharp step corner and the shear layer. This indicates that the

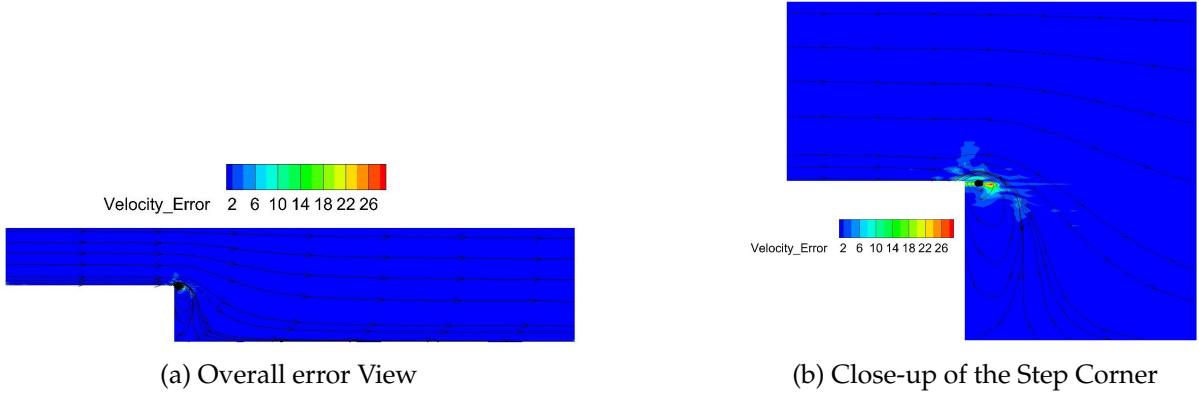


Figure 11: Visualization of the model’s prediction error for the U-velocity. The error is concentrated in regions with complex physics, such as the sharp step corner and the shear layer, $\text{Kn}=1$.

model is capable of identifying regions where predictions are less reliable, a critical feature for practical applications of surrogate modeling.

At $\text{Kn} = 0.02$, the uncertainty levels remain concentrated around the recirculation bubble and shear layer, but the affected region becomes slightly broader than in the error maps. This suggests that while the model can still produce accurate predictions, it anticipates difficulty in capturing rapid velocity variations and possible secondary vortical structures, signaling caution in those regions.

At $\text{Kn} = 1$, the uncertainty is strongly localized, with maximum values clustered around the step corner and the vortex reattachment zone. The close-up views show a clear correspondence between high-gradient regions induced by rarefaction effects and elevated uncertainty. Importantly, the localization of uncertainty confirms that the DeepONet framework not only delivers accurate predictions but also quantifies the reliability of those predictions, providing a robust and interpretable surrogate for rarefied gas flow simulations.

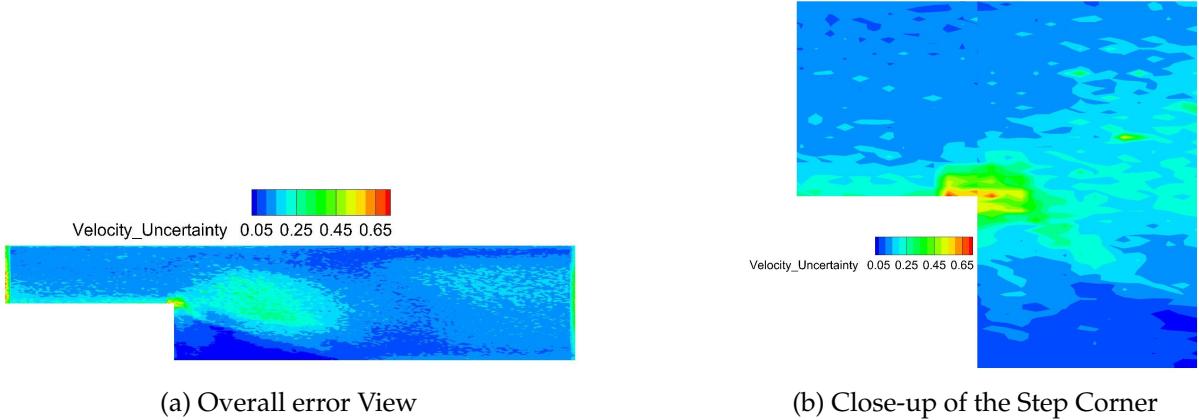


Figure 12: Visualization of the model’s prediction uncertainty for the U-velocity. The uncertainty is concentrated in regions with complex physics, such as the sharp step corner and the shear layer, $\text{Kn}=0.004$.

7.1.5 Velocity Profiles Comparison

To further assess the robustness of the proposed framework, Figures 15–17 present a quantitative validation against high-fidelity DSMC results along several representative horizontal

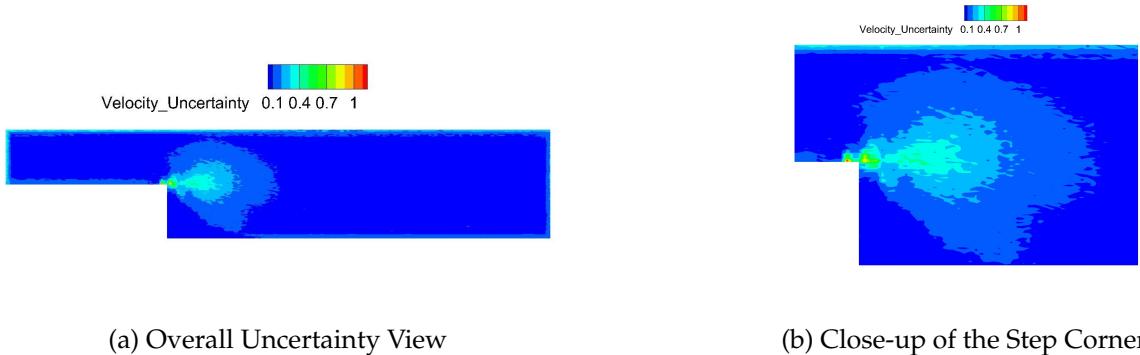


Figure 13: Visualization of the model’s prediction uncertainty for the U-velocity. The uncertainty is concentrated in regions with complex physics, such as the sharp step corner and the shear layer, $\text{Kn}=0.02$.

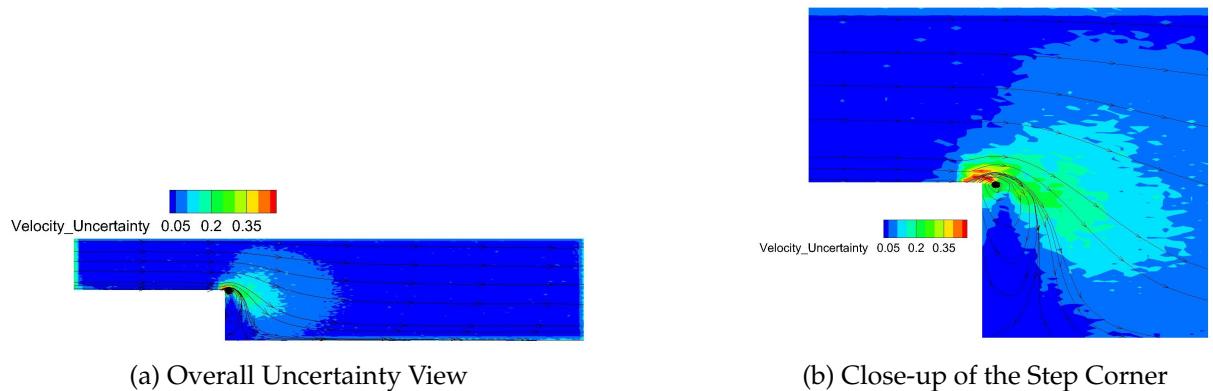


Figure 14: Visualization of the model’s prediction uncertainty for the U-velocity. The uncertainty is concentrated in regions with complex physics, such as the sharp step corner and the shear layer, $\text{Kn}=1$.

lines. Four lines, 1) one on the bottom wall (wall 3 in Fig. 1, called bottom in Fig. 15), 2) one on the upper wall (wall 4 in Fig. 1, called up in Fig. 15), 3) a line over wall 1 extending towards the exit of the channel (called "middle" in Fig. 15), 4) and a line in the step at the middle between line "middle" and "up", called middle-up, are considered. Both components of the velocity are compared on these lines. Solid curves denote DSMC reference data, while dashed curves correspond to the DeepONet predictions.

In Figure 15 ($\text{Kn} = 0.004$), the agreement is strikingly close for both the U - and V -velocity components. In particular, the DeepONet successfully captures the negative U -velocity region inside the vortex, which is the most challenging portion of the flow field due to strong non-equilibrium effects. Minor deviations are visible in the flow recovery zone downstream of the reattachment point, where the model slightly underpredicts the velocity magnitude. This discrepancy is an anticipated trade-off, as the zonal loss function was tailored to emphasize accuracy in the vortex-dominated region.

Figure 16 ($\text{Kn} = 0.02$) confirms that the model generalizes well as the Knudsen number increases, maintaining a high level of fidelity across all velocity components. The predicted V -velocity profiles show excellent agreement with DSMC, even in regions with sharp gradients near the separation and reattachment points. Slight mismatches occur in the far-field

streamwise recovery region, but the overall error remains small and localized.

At $Kn = 1$ (Figure 17), corresponding to the transition between slip and transitional regimes, the model continues to reproduce the main features of the DSMC data. The DeepONet predictions capture both the overall structure of the U -velocity profiles and the subtle variations of the V -velocity component. Some discrepancies appear in the weaker secondary recirculation zone close to the step corner, but the predicted profiles remain within acceptable error margins. The preservation of accuracy across three orders of magnitude in Knudsen number demonstrates the strong generalization capability of the network.

Overall, these comparisons establish that the proposed DeepONet not only replicates DSMC results in regions explicitly emphasized during training but also provides reliable predictions across a broad range of flow regimes. The ability to resolve both primary vortex structures and recovery dynamics underscores the suitability of the framework for applications involving rarefied gas flows.

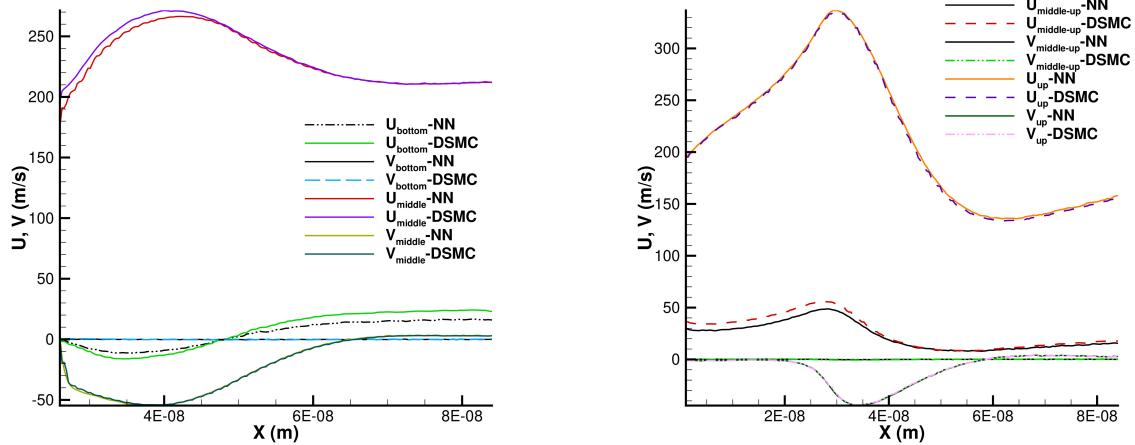


Figure 15: Quantitative comparison of velocity profiles along four horizontal lines. Solid lines are DSMC data, dashed lines are DeepONet predictions., $Kn=0.004$.

7.2 Application to Geometric Parameter Variation: Step Height

To showcase the versatility of the proposed framework, the DeepONet surrogate model was adapted to predict the velocity field as a function of a geometric parameter—the ratio of the step height to the channel height (h/H). This demonstrates the model’s capability to learn the operator mapping from the domain geometry to the flow solution, a critical task for shape optimization problems.

Figure 18 schematizes the geometry used to quantify the influence of the step-height ratio on separation. The channel has total height H and length L ; upstream of the step, the local height is h (with $h < H$), and the backward-facing step produces an expansion into the lower channel. Walls are labeled 1–4 consistent with the problem statement; the flow is driven by an inlet/outlet pressure ratio while all walls are kept isothermal. Knudsen number is set as $Kn=0.01$. The control parameter for this study is the non-dimensional step height h/H , which we vary systematically to modify the strength and extent of the recirculation region behind the step. This schematic clarifies how increasing h/H enlarges the expansion and, consequently, tends to intensify separation and shift the reattachment location downstream. At the investigated Knudsen number, there is no vorticity on the top wall of the step geometry.

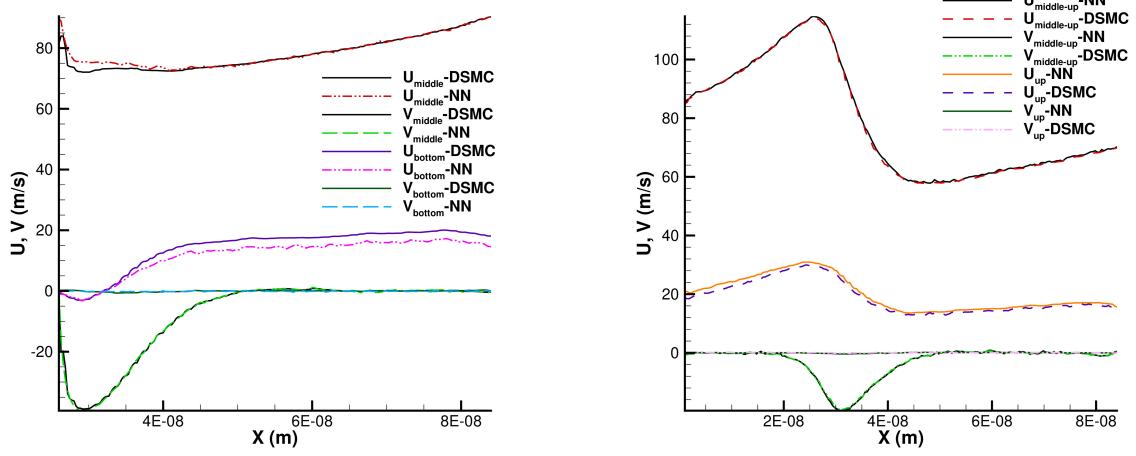


Figure 16: Quantitative comparison of velocity profiles along four horizontal lines. Solid lines are DSMC data, dashed lines are DeepONet predictions, $\text{Kn}=0.02$.

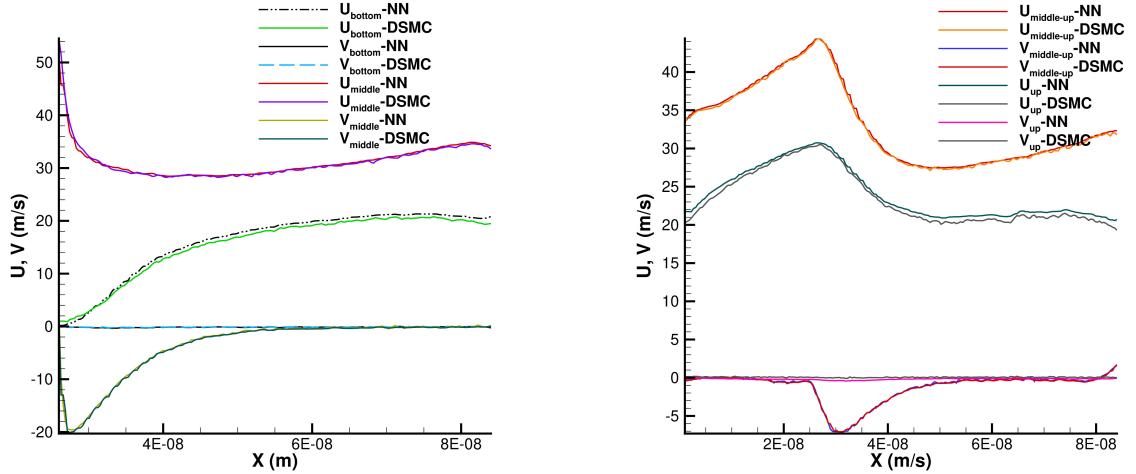


Figure 17: Quantitative comparison of velocity profiles along four horizontal lines. Solid lines are DSMC data, dashed lines are DeepONet predictions, $\text{Kn}=1$.

7.2.1 DSMC snapshots across height ratios.

Figure 19 presents DSMC results for eight representative values of the step–height ratio, showing streamlines overlaid on the U -velocity contours. A clear, monotonic trend emerges: as h/H increases from 0.16 to 0.75, the primary recirculation bubble expands and its center moves downstream; the shear layer becomes thicker and more curved; and the peak streamwise velocity in the core decreases due to the stronger expansion. For small h/H the separation is compact and the reattachment occurs relatively close to the corner, whereas for large h/H the separation length grows appreciably and the low-speed region occupies a larger portion of the lower channel. These qualitative changes in both streamlines and U -contours provide a rich set of flow topologies for learning.

The core methodology, including the convolutional trunk and the physics-guided zonal loss function, remains consistent with the Knudsen number study.

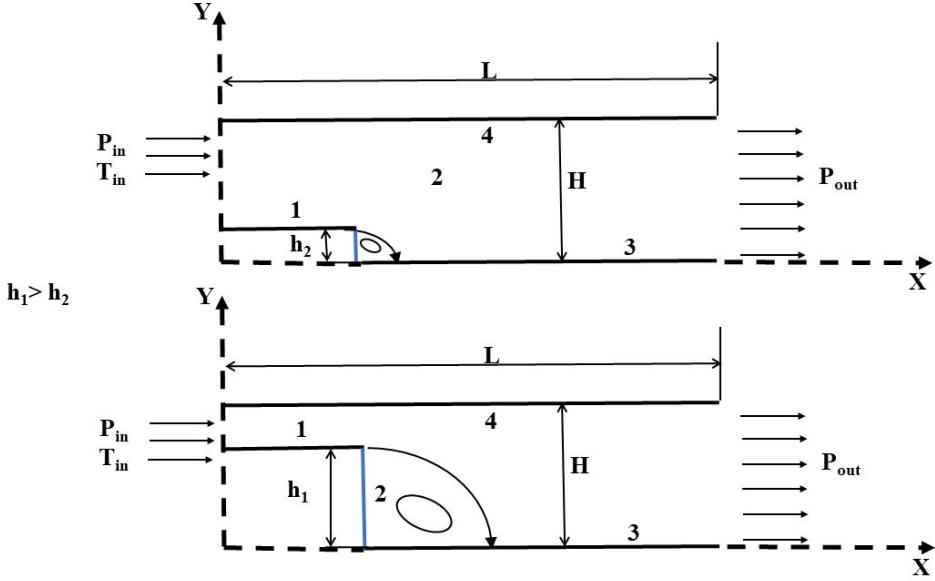


Figure 18: Schematic of the simulated geometry for height study

The specific hyperparameters for the DeepONet architecture used in this study are detailed as follows. The model is composed of a Branch network, a convolutional Trunk network, and a Head network that combines their outputs.

- **Branch Network:** The branch network, which processes the scalar step height value, consists of 6 ResNet blocks with a width of 384 neurons.
- **Trunk Network:** The trunk network first processes local field information using two convolutional layers with 32 and 64 filters, respectively, operating on input patches of size 5x5. The flattened output is then combined with the point coordinates and passed through an MLP of 4 dense layers with a width of 384 neurons.
- **Head Network:** The outputs of the branch and trunk networks are merged and processed by a final head network composed of 4 dense layers with a width of 768 neurons.

Crucially, the physics-guided zonal loss function was employed to ensure high fidelity in the recirculation zone. A weighting factor of $\alpha = 0.6$ was used, prioritizing the accuracy of the negative velocity region behind the step. A dropout rate of 0.3 was applied for regularization and for enabling uncertainty quantification via Monte Carlo Dropout.

The DSMC fields shown in Figure 19 constitute the geometry-dependent dataset for training the neural operator. In the Branch network, the scalar input is the height ratio h/H , while the Trunk network takes spatial coordinates (x, y) ; the network's output is the velocity field at those coordinates. We generated DSMC solutions at nine values of h/H in the interval $[0.16, 0.75]$; eight of these, shown in Figure 19 are used for training and one *unseen* case, $h/H = 0.44$, is withheld for validation. The held-out prediction assesses the model's ability to generalize to a geometry it did not observe during training, i.e., to interpolate accurately within the parameter space solely from the operator mapping it has learned.

7.2.2 Training dynamics for height variation.

Figure 20 illustrates the loss history for the DeepONet trained on the height-ratio dataset. Compared to the Knudsen-number study, where the model was exposed to 20 training cases

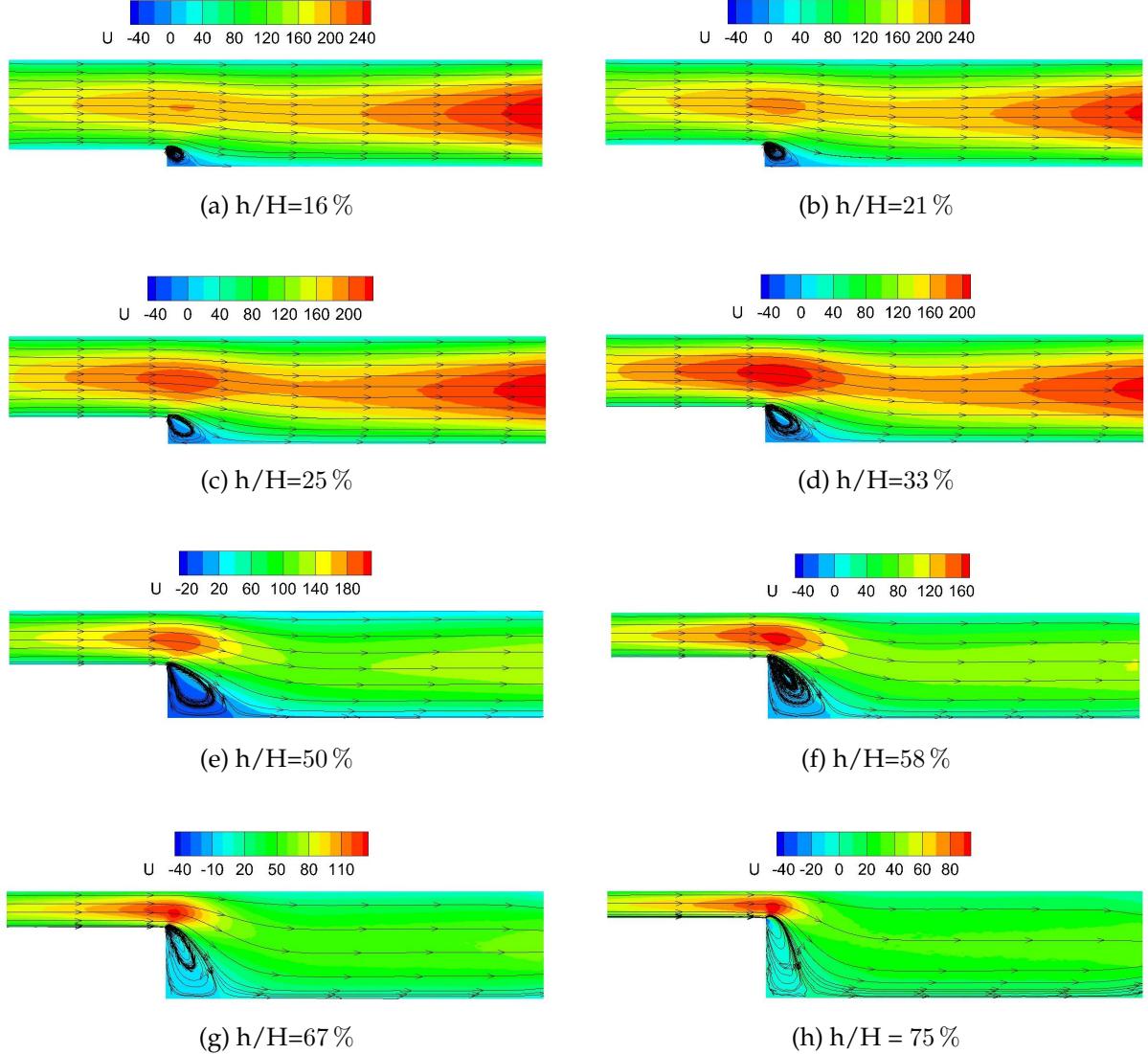


Figure 19: Qualitative comparison of the U-velocity contour and streamlines for representative step to channel heights (h/H) ratios in the dataset.

spanning a wide range of flow regimes, here only 8 training geometries are available. This reduced dataset naturally limits the richness of the operator mapping and slows down convergence. The training and validation losses (solid blue and orange curves) both decrease steadily, but they saturate at higher values than in the Knudsen case, where denser sampling allowed the network to interpolate more accurately. Similarly, the mean squared error (MSE) of the velocity components (dashed green and red curves) remains an order of magnitude higher than in the Knudsen-number experiment. Nevertheless, the oscillations damp out after about 100 epochs, and both training and validation curves track each other closely, confirming that the model does not overfit despite the smaller dataset. This comparison highlights the strong dependence of neural operator accuracy on the density and diversity of training samples: with only eight height ratios, the generalization error is larger than for the Knudsen-number mapping, yet still sufficiently low to capture the underlying flow physics and predict the unseen case at $h/H = 0.44$.

Overall, while the prediction error is higher than in the Knudsen-number study due to the reduced number of training samples, the results demonstrate that DeepONet can still capture the essential vortex dynamics and velocity distribution even with sparse data, underscoring

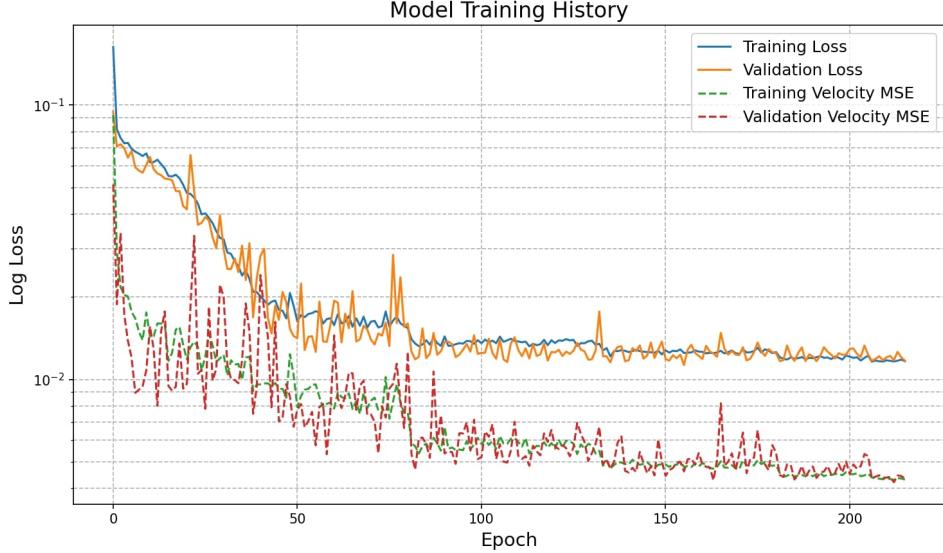


Figure 20: Loss Function and MSE for the step with height variation

its robustness as a surrogate model for rarefied gas flows.

Figures 21 and 22 demonstrate the extrapolative capability of the DeepONet framework when applied to an unseen geometric configuration with a step height ratio of $h/H = 44\%$. In Figure 21, the predicted U- and V-velocity fields are shown to be in excellent agreement with the DSMC ground truth, successfully capturing the onset and extent of the primary recirculation zone, the detachment and reattachment points, and the subtle shear-layer dynamics in the main channel. This highlights the ability of the network to generalize beyond the training set and approximate the nonlinear operator mapping geometric parameters (h/H) to the full velocity field.

The corresponding error and uncertainty distributions in Figure 22 reveal that discrepancies are primarily localized to regions of strong gradients and non-equilibrium effects, such as the sharp step corner and shear layer. Importantly, the model's uncertainty quantification exhibits strong spatial correlation with the error distribution, indicating that the DeepONet not only learns the flow operator but also provides a meaningful estimate of prediction reliability. Although the test case lies outside the training manifold, the magnitude of the error remains low across the majority of the flow domain, underscoring the robustness of the operator-learning paradigm. This result confirms that DeepONet is capable of faithfully representing the complex multi-scale dynamics of rarefied gas flows under geometric variations, a task that conventional neural architectures often fail to achieve.

7.2.3 Validation with Multiple Hold-Out Cases and Increased Data Sparsity

To further test the model's limits, a more challenging scenario was devised. The model was trained on a reduced dataset of only seven simulation cases, while two cases at step height ratios of $h/H = 44\%$ and $h/H = 67\%$ were held out for testing. This scenario simulates a situation with extreme data scarcity.

The training history for this more demanding setup is shown in Figure 23. In contrast to the previous case, a small but noticeable gap emerges between the training loss and the validation loss. This gap is a classic sign of mild overfitting. With fewer training examples, the high-capacity DeepONet model begins to memorize the specific characteristics of the training set, slightly hindering its ability to generalize to the two unseen test cases.

Despite this mild overfitting, the model’s predictive capabilities remain strong across both held-out cases. Figures 24 and 25 present the qualitative velocity comparisons for the test cases at $h/H = 44\%$ and $h/H = 67\%$, respectively. In both scenarios, the DeepONet’s predictions for both U- and V-velocity show a strong qualitative agreement with the DSMC ground truth. The model correctly identifies the critical flow structures, such as the recirculation zone and the main channel flow, for both an interpolating case ($h/H = 44\%$) and a more extrapolating case ($h/H = 67\%$). This result is significant, as it demonstrates the robustness of the surrogate modeling framework. Even when pushed to the limits of data availability where mild overfitting occurs, the model does not fail catastrophically and still produces physically plausible and largely accurate predictions across the parameter space.

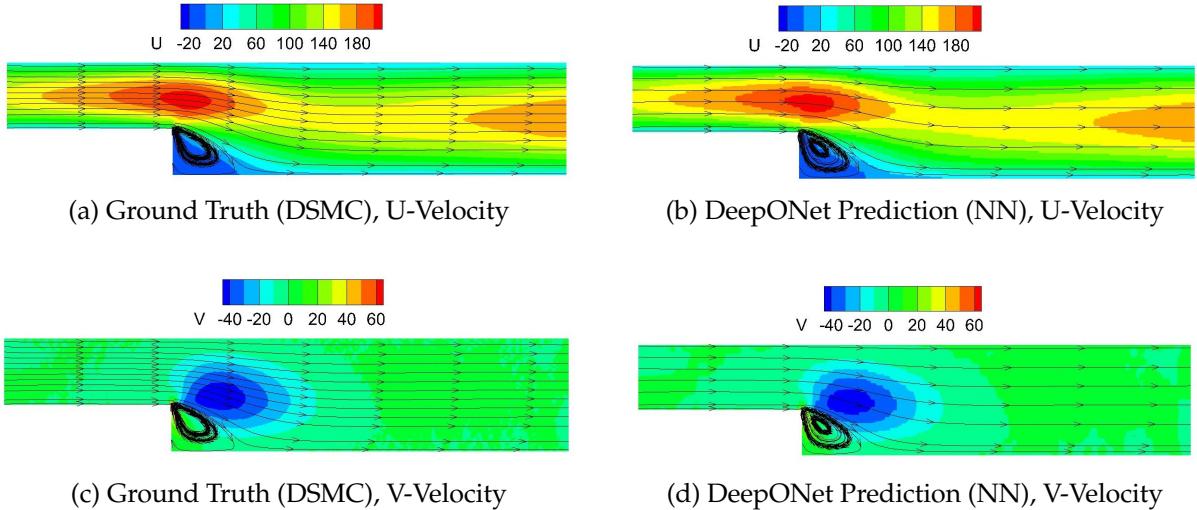


Figure 21: Qualitative comparison of the U-velocity and V-velocity contours between the ground truth DSMC simulation and the DeepONet prediction for the unseen step height ratio of $h/H=44\%$.

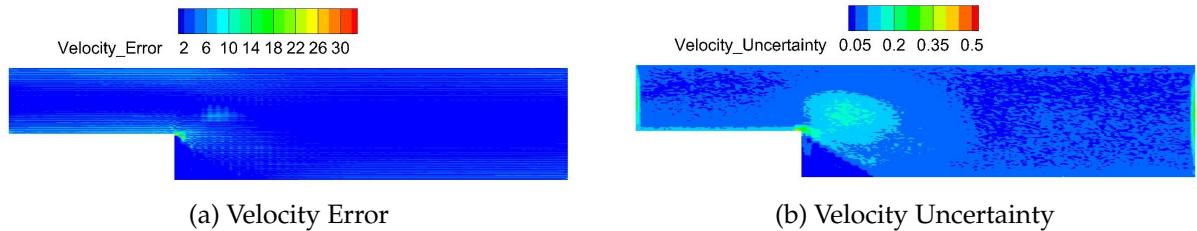


Figure 22: Velocity error and uncertainty for the case of $h/H=44\%$.

7.3 Comparison of Various Loss Function Formulations

In this section, we compare three distinct loss functions to train the DeepONet model. The choice of loss function directly influences the model’s learning process and its ability to handle localized physical phenomena. We compare our Zonal Loss Function with the Mean Squared Error (MSE) Loss and Gradient-Weighted Mean Squared Error (GMSE) Loss suggested in [34].

7.3.1 Mean Squared Error (MSE) Loss

The Mean Squared Error is a standard, widely used loss function that treats all data points with equal importance. It computes the average squared difference between the predicted values (\hat{y}_i) and the true values (y_i) over all N points in the training batch. The formulation is given

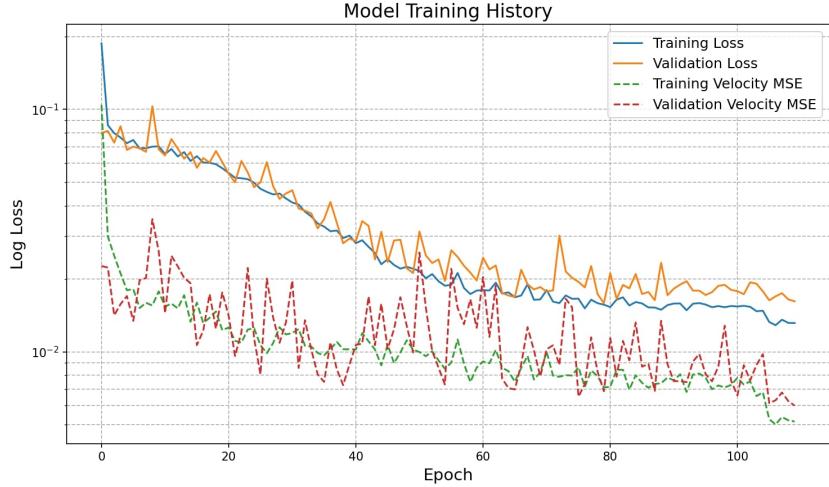


Figure 23: Training and validation loss history for the simplified DeepONet model trained on cases with varying step heights.

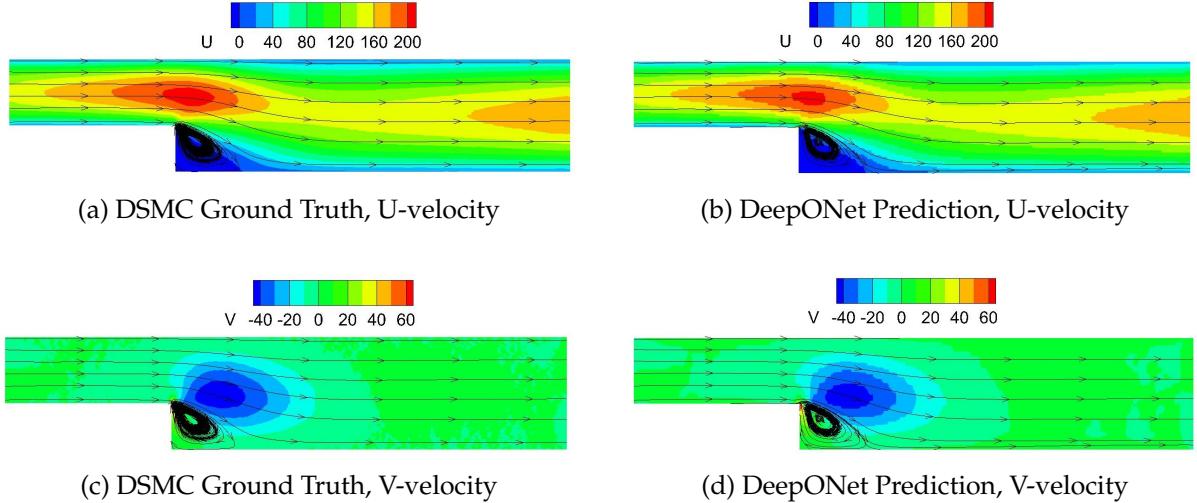


Figure 24: Qualitative comparison of U-velocity (top row) and V-velocity (bottom row) contours between the ground truth DSMC simulation and the DeepONet prediction for the unseen step height ratio of $h/H = 44\%$.

by:

$$L_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N \|y_i - \hat{y}_i\|^2 \quad (16)$$

While robust and straightforward, the MSE loss function does not account for the underlying physical structure of the problem. For flows with localized, high-gradient regions, its global averaging nature can lead to the "smearing" of errors from critical zones across the entire computational domain.

7.3.2 Gradient-Weighted Mean Squared Error (GMSE) Loss

The Gradient-Weighted Mean Squared Error (GMSE) is an advanced loss function proposed by Cooper-Baldock et al. designed to enhance model training for datasets like those in Computational Fluid Dynamics (CFD), where small, high-variance regions contain the most critical information [34]. It modifies the standard Mean Squared Error by assigning a spatially-varying

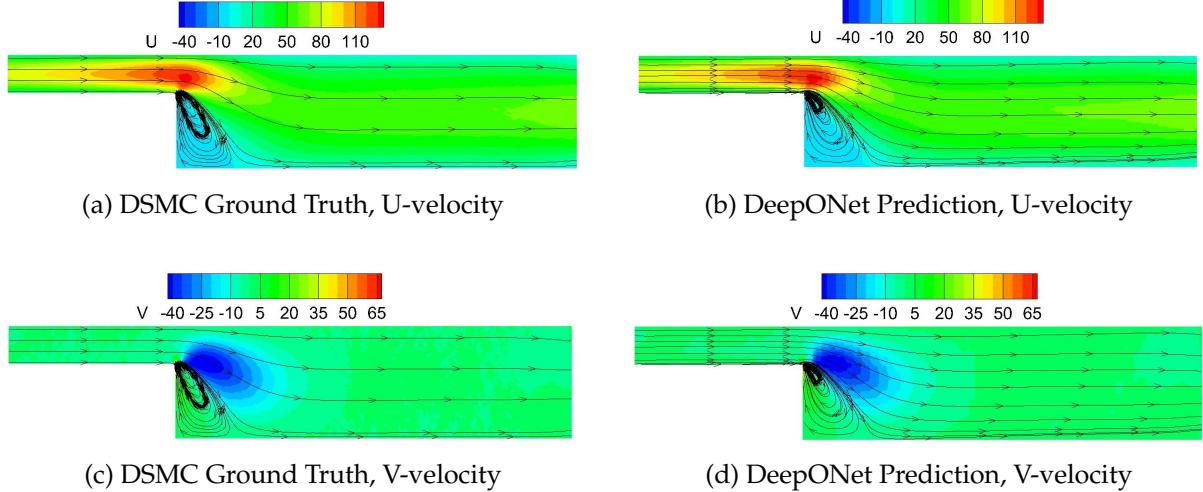


Figure 25: Qualitative comparison of U-velocity (top row) and V-velocity (bottom row) contours between the ground truth DSMC simulation and the DeepONet prediction for the unseen step height ratio of $h/H = 67\%$.

weight, $w(\mathbf{x}_i)$, to the error at each point \mathbf{x}_i . This compels the model to focus more on regions with complex features and strong gradients [34].

The general form of the GMSE loss is given by:

$$L_{\text{GMSE}} = \frac{1}{N} \sum_{i=1}^N w(\mathbf{x}_i) \|y_i - \hat{y}_i\|^2 \quad (17)$$

The dynamic weight matrix $w(\mathbf{x}_i)$ (denoted as W_i in the paper) is calculated for each ground truth field through a multi-step process:

1. Disparity Calculation: First, the gradient field is computed to identify regions of high pixel disparity. This is achieved by calculating the element-wise difference along the x and y axes and then determining the magnitude of the resulting vector[cite: 199, 205]. This non-linear operation effectively captures high-frequency details[cite: 214].

$$W_{d,x} = W_x - W_{x-1} \quad (18)$$

$$W_{d,y} = W_y - W_{y-1} \quad (19)$$

$$W_d = \sqrt{W_{d,x}^2 + W_{d,y}^2} \quad (20)$$

2. Gaussian Blur: A Gaussian blur is applied to the disparity array W_d to smooth and slightly enlarge the identified areas of importance[cite: 217]. The Gaussian function is defined as:

$$W_{\text{blur}}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (21)$$

where σ controls the standard deviation of the blur.

3. Gamma Correction: The blurred array, W_{blur} , is then raised to the power of a fixed value γ to enhance the contrast, which strengthens or weakens the gradient weights[cite: 222].

$$W_\gamma = W_{\text{blur}}^\gamma \quad (22)$$

4. Normalization and Offset: The resulting array W_γ is normalized to a range of $[0, 1]$. To ensure that low-gradient regions (e.g., the freestream) are not entirely ignored, an offset C_o is applied to establish a non-zero lower bound for the weights[cite: 252]. The final weight matrix W_i is calculated as:

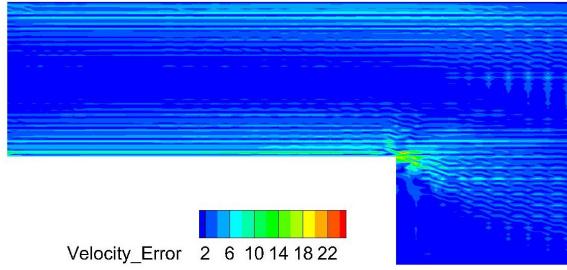
$$W_{\text{norm}} = \frac{W_\gamma - \min(W_\gamma)}{\max(W_\gamma) - \min(W_\gamma)} \quad (23)$$

$$W_i = (W_{\text{norm}} \cdot [1 - C_o]) + C_o \quad (24)$$

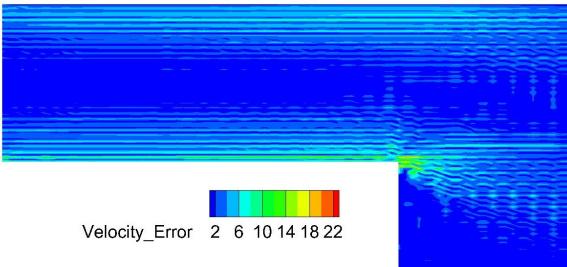
This comprehensive weighting strategy provides a "soft" but highly effective focus on physically significant areas like shear layers and wakes, leading to faster convergence and more accurate reconstructions[cite: 38, 595].

7.4 Results of Ablation Study: Comparison of Prediction Errors

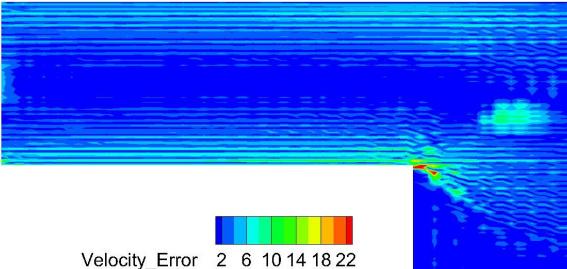
The efficacy of each loss function is visually assessed by examining the spatial distribution of the prediction error for the U-velocity component. Figure 26 presents a zoomed-in view of the error fields in the region immediately downstream of the backward-facing step.



(a) Error distribution using the MSE loss function.



(b) Error distribution using the GMSE loss function.



(c) Error distribution using the Zonal loss function.

Figure 26: A comparative visualization of the prediction error for the U-velocity field near the step, resulting from models trained with three different loss functions.

A direct comparison of the error distributions (Fig. 26a–26c) shows distinct characteristics for each loss function. The **MSE loss** (Fig. 26a) produces relatively moderate error levels, but these

errors are dispersed widely across the flow domain, extending far downstream in oscillatory bands. The **GMSE loss** (Fig. 26b) leads to a more localized error distribution compared to MSE, with stronger concentration near the shear layer and recirculation zone, though some downstream contamination remains visible. In contrast, the **Zonal loss** (Fig. 26c) yields the highest peak error magnitude in the immediate vicinity of the step corner, but this error is strongly confined to the vortex region. The rest of the domain remains relatively clean, with substantially reduced background error compared to the other two methods.

These observations indicate that MSE favors smoother but more widespread errors, GMSE provides partial improvement by capturing gradients yet still allows error leakage, while the Zonal strategy produces sharper but localized errors concentrated in the physically complex region of the flow.

7.4.1 Quantitative Error Analysis

To quantitatively assess the performance of each loss function, the L2 relative error norm was computed for the hold-out test case. The L2 relative error norm is defined by the following equation:

$$\text{L2 Error} = \frac{\|y - \hat{y}\|_2}{\|y\|_2} = \frac{\sqrt{\sum_{i=1}^N (y_i - \hat{y}_i)^2}}{\sqrt{\sum_{i=1}^N y_i^2}} \quad (25)$$

where N is the total number of data points in the domain being evaluated, and $\|\cdot\|_2$ denotes the Euclidean norm (L2 norm) of a vector. The resulting value is a dimensionless quantity, often expressed as a percentage, which indicates the overall predictive discrepancy.

In this work, the L2 relative error norm is calculated based on the velocity vector field, which includes both the horizontal (U) and vertical (V) velocity components. It quantifies the difference between the true velocity vector, \mathbf{u} , and the predicted velocity vector, $\hat{\mathbf{u}}$, across all N points in the domain.

The formula is defined as:

$$\text{L2 Error} = \frac{\|\mathbf{u} - \hat{\mathbf{u}}\|_2}{\|\mathbf{u}\|_2} = \frac{\sqrt{\sum_{i=1}^N ((U_i - \hat{U}_i)^2 + (V_i - \hat{V}_i)^2)}}{\sqrt{\sum_{i=1}^N (U_i^2 + V_i^2)}} \quad (26)$$

where (U_i, V_i) are the components of the true velocity vector at point i , and (\hat{U}_i, \hat{V}_i) are the components of the predicted velocity vector.

The analysis was performed on both the full computational domain and specifically within the vortex recirculation region (where the true velocity $U < 0$). The results are summarized in Table 2.

Table 2: L2 Relative Error Comparison for the Test Case.

Method	L2 Error (Full Domain)	L2 Error (Vortex Region)
MSE	2.1739%	14.6135%
GMSE	2.2070%	17.4516%
Zonal	2.2254%	11.9413%

The quantitative results reveal a crucial performance trade-off. While the standard **MSE** loss achieves the lowest error on the full domain, this is likely because the model's optimization is dominated by the large, low-gradient freestream areas, which are easier to predict. This

global optimization, however, comes at the cost of reduced accuracy in the more complex and physically significant vortex region.

Conversely, the Zonal loss method demonstrates the lowest error within the vortex region by a significant margin. This superior performance is a direct result of its design. The Zonal loss function explicitly segregates the domain based on a physical criterion ($U < 0$) and applies a higher weight to the loss calculated within the vortex. This forces the model to prioritize learning the complex flow dynamics in that specific zone, resulting in a more physically faithful reconstruction where it matters most.

This analysis suggests that for engineering applications where the fidelity of a specific, complex flow feature (e.g., recirculation, separation bubbles, shockwaves) is more critical than the overall global error metric, a targeted approach like the Zonal loss function is the more effective strategy.

7.5 A Convolutional Fusion-DeepONet for Rarefied Flows

To create a surrogate model capable of accurately predicting geometry-dependent rarefied flow fields from sparse data, we developed a novel hybrid neural operator architecture, which we term the **Convolutional Fusion-DeepONet (CF-DeepONet)**. This architecture is designed to learn the operator $\mathcal{G} : h \mapsto \vec{v}(y)$ that maps a scalar geometric parameter, the step height h , to the corresponding 2D velocity field \vec{v} at any query coordinate $y = (x, y)$ in the domain. The CF-DeepONet uniquely combines the patch-based spatial feature extraction of a convolutional neural network (CNN) with the multi-scale conditioning mechanism of the Fusion-DeepONet, an architecture proposed by Peyvan et al. for continuum flows. Our model is composed of three primary sub-networks: a Branch Network to process the geometric parameter, a Trunk Network to process spatial information, and a Head Network for the final prediction, as detailed below.

Branch Network. The Branch Network is responsible for encoding the input geometric parameter, the scalar step height $h \in \mathbb{R}$. The input h is first passed through a dense layer to expand its dimensionality to the model’s hidden width, d_{model} . This feature vector is then processed by a stack of L_b ResNet blocks, which apply a series of nonlinear transformations. Crucially, the outputs of the first L_t hidden layers are stored as a set of conditioning vectors $\{z^{(1)}, z^{(2)}, \dots, z^{(L_t)}\}$, where each $z^{(l)} \in \mathbb{R}^{d_{model}}$. These vectors serve as the geometry-aware signals that are fused into the Trunk Network at multiple scales. The final output of the branch network after all L_b blocks is a single feature vector denoted by b_{final} .

Trunk Network. The Trunk Network is designed to process spatial information from two distinct sources: the specific query coordinate y and a local snapshot of the velocity field around that coordinate. It consists of a convolutional front-end followed by a fusion-informed MLP.

- **Convolutional Front-End:** To provide the model with local physical context, we supply a $P_s \times P_s$ patch of the velocity field, $P \in \mathbb{R}^{P_s \times P_s \times 2}$, centered at the query coordinate y . This patch is processed by a series of 2D convolutional and average pooling layers to extract a low-dimensional feature vector, $y_{feat} = \text{CNN}(P)$. This vector, which encodes the local flow structure (e.g., gradients, curvature), is then concatenated with the coordinate vector $y = (x, y)$.
- **Fusion-Informed MLP:** The concatenated feature vector $[y, y_{feat}]$ is first projected by an initial dense layer into a high-dimensional space, yielding the initial trunk vector $t^{(0)} \in \mathbb{R}^{d_{model}}$. This vector then enters the core of the architecture: a fusion block consisting of L_t layers. At each layer l , the trunk vector $t^{(l-1)}$ is fused with the corresponding

conditioning signal $z^{(l)}$ from the Branch Network via an element-wise product (\odot). The result is then passed through a dense layer with a tanh activation function to produce the next-level trunk vector. This fusion process is described by the recurrence relation:

$$t^{(l)} = \text{Dense}_{\text{tanh}} \left(t^{(l-1)} \odot z^{(l)} \right), \quad \text{for } l = 1, \dots, L_t$$

This multi-scale conditioning allows the geometric information from the branch to influence the trunk's processing of spatial features at every level, creating a set of powerful, geometry-informed basis functions. The final output of this process is the trunk vector $t_{final} = t^{(L_t)}$.

Head Network and Final Prediction. In the final stage, the geometry-aware spatial features (t_{final}) are modulated by the global parameter features (b_{final}) through another element-wise product: $m = b_{final} \odot t_{final}$. This combined and richly-informed feature vector m is then processed by a final multi-layer perceptron, termed the Head Network, which consists of L_h dense layers. The Head Network maps the latent features to the physical space, producing the predicted 2D velocity vector $\vec{v}_{pred} \in \mathbb{R}^2$ at the query coordinate y .

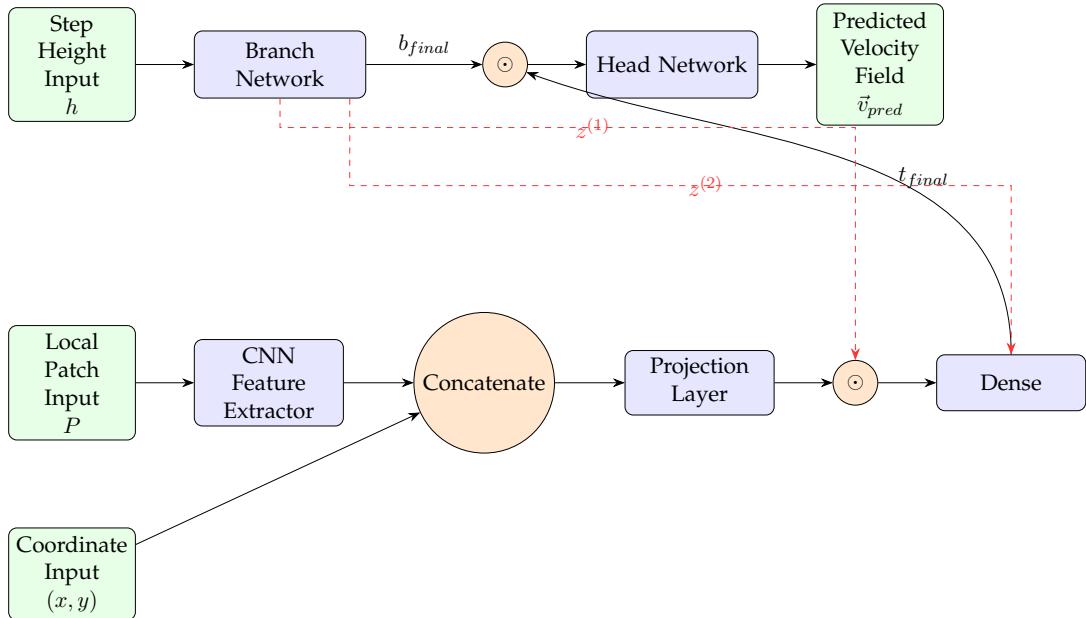


Figure 27: Flowchart of the Fusion-DeepONet Architecture

7.5.1 Discussions on the Fusion Model Performance

The performance of both the standard Convolutional DeepONet and the more advanced Convolutional Fusion-DeepONet was evaluated on the held-out test case ($h/H = 44\%$). A qualitative comparison of the predicted U-velocity and V-velocity fields against the ground truth DSMC data is presented in Figure 29. Visually, both models successfully capture the primary features of the flow, including the location and general shape of the recirculation zone and the velocity distribution in the main channel.

For a quantitative assessment, the Root Mean Squared Error (RMSE) was calculated across the entire velocity field for both models. The results are summarized in Table 3. Surprisingly, the analysis reveals that the standard DeepONet architecture achieves a lower overall error compared to the more complex Fusion-DeepONet.

Table 3: Quantitative performance comparison of the surrogate models on the test case.

Model Architecture	Root Mean Squared Error (RMSE)
Standard DeepONet	2.689
Fusion-DeepONet	3.589

7.5.2 Analysis of Model Performance and Overfitting

The superior performance of the simpler architecture in this study highlights a critical concept in machine learning: the trade-off between model complexity and data availability. The Fusion-DeepONet, with approximately 1.5 million trainable parameters, is a significantly more expressive model than the standard DeepONet, which has approximately 1.1 million parameters. While this increased capacity is theoretically advantageous, it also makes the model more susceptible to overfitting, especially when trained on a sparse dataset.

Direct evidence of this overfitting can be observed by comparing the training histories of the two models. The loss plot for the standard DeepONet (Figure 20) shows that the training and validation losses track each other closely, indicating good generalization. In contrast, the loss plot for the Fusion-DeepONet (Figure 28) reveals a significant and persistent gap between the training and validation loss curves. This divergence is the classic signature of overfitting: the model has begun to memorize the specific noise and artifacts of the seven training cases rather than learning the underlying physical operator. Consequently, its ability to generalize to the unseen test case is compromised, leading to a higher overall error.

This result is not a failure of the Fusion-DeepONet architecture itself, but rather a crucial finding on its data requirements for this class of problems. It suggests that for rarefied flow simulations where generating high-fidelity data is computationally expensive, a well-regularized, simpler architecture may provide more robust and accurate predictions than a more complex model that cannot be adequately constrained by the limited data.

As shown in Figure 20, the DeepONet trained on step-height variations exhibits a relatively smooth convergence, with both the training and validation losses decreasing steadily and eventually reaching low values of the order of 10^{-2} . Although the error level is higher than in the Knudsen-number case (with 20 input samples), the DeepONet still achieves a robust approximation given that only 8 input variations were provided.

In contrast, the Fusion algorithm (Figure 28) does not reach the same level of accuracy. While the overall trend of the loss curves indicates convergence, the final training and validation errors plateau at significantly higher levels compared to the DeepONet case. Furthermore, the velocity MSE curves remain noisy and do not consistently decrease to the same extent. This performance gap highlights a fundamental limitation: with only 8 training samples for the height variation case, the Fusion model does not appear to fully leverage its more complex architecture. The Fusion strategy likely requires a larger and more diverse dataset to realize its potential advantages over the standard DeepONet. These results suggest that while DeepONet can generalize well even with relatively scarce data, Fusion-based models may demand substantially more training data to achieve competitive accuracy.

Figures 29 and 30 provide a direct comparison between the baseline DSMC results, the DeepONet predictions, and the Fusion-DeepONet approach for the unseen step-height ratio of $h/H = 0.44$.

In Figure 29, the U-velocity and V-velocity contours are presented side by side. The DeepONet predictions (middle row) capture both the streamwise velocity acceleration above the step and the recirculation structure below the step with high fidelity, closely resembling the reference

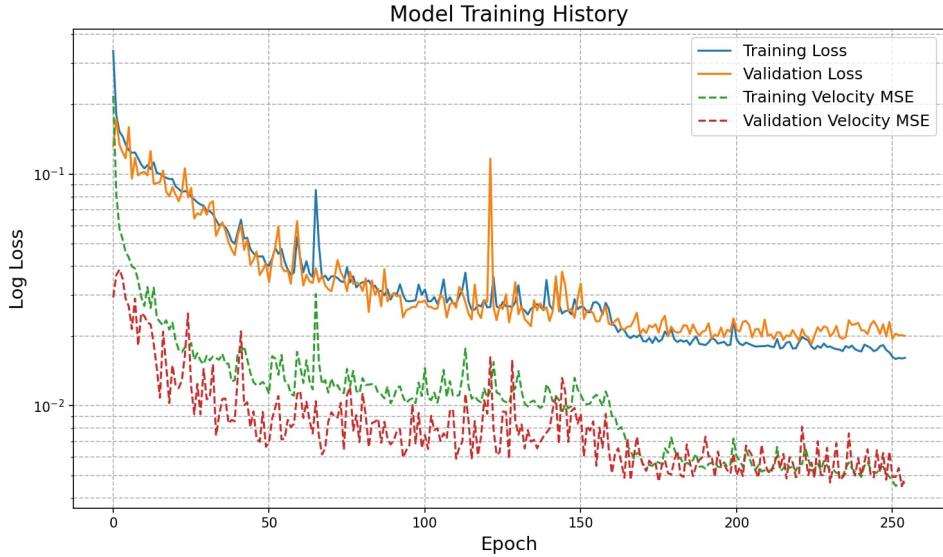


Figure 28: Loss Function and MSE for the Fusion Algorithm

DSMC fields (top row). The Fusion-DeepONet predictions (bottom row) display some deviations in the vortex core and in the shear layer. Slight differences are also observed in the V-velocity component.

The discrepancy is quantified in Figure 30, where error maps are shown for DeepONet (left) and Fusion-DeepONet (right). The DeepONet error field exhibits relatively low intensity, with the largest deviations localized near the separation and reattachment zones. On the other hand, the Fusion-DeepONet error field shows broader regions of elevated error, particularly in the shear layer and wake region, indicating that the Fusion approach introduces additional uncertainty in predicting small-scale flow features. These results confirm that DeepONet provides superior generalization in the scarce-data regime, while the Fusion model may require significantly more training samples to achieve comparable accuracy.

8 Concluding Remarks

This study presented a convolutional DeepONet surrogate model, augmented with a physics-guided zonal loss, for predicting rarefied flow over a micro backward-facing step. The work establishes both methodological innovations and practical implications, which can be summarized as follows:

- **Effectiveness of the Zonal Loss.** The proposed zonal loss provides superior accuracy in physically critical regions of the flow, particularly within recirculation bubbles where $U < 0$. As demonstrated in Table 2, global error metrics alone can be misleading by underestimating deficiencies in vortex resolution. By emphasizing localized errors, the zonal loss achieves a more faithful reconstruction of separated flow structures, ensuring that engineering-critical phenomena are captured with higher fidelity.
- **Insights from the Fusion-DeepONet Comparison.** The comparative analysis with Fusion-DeepONet, summarized in Table 3, should not be interpreted as a shortcoming of the hybrid model. Instead, it provides a valuable scientific observation: increasing architectural complexity without sufficient training data can degrade generalization performance. This highlights a fundamental trade-off between model sophistication and data availability in computationally expensive regimes. Such insights are crucial for guiding

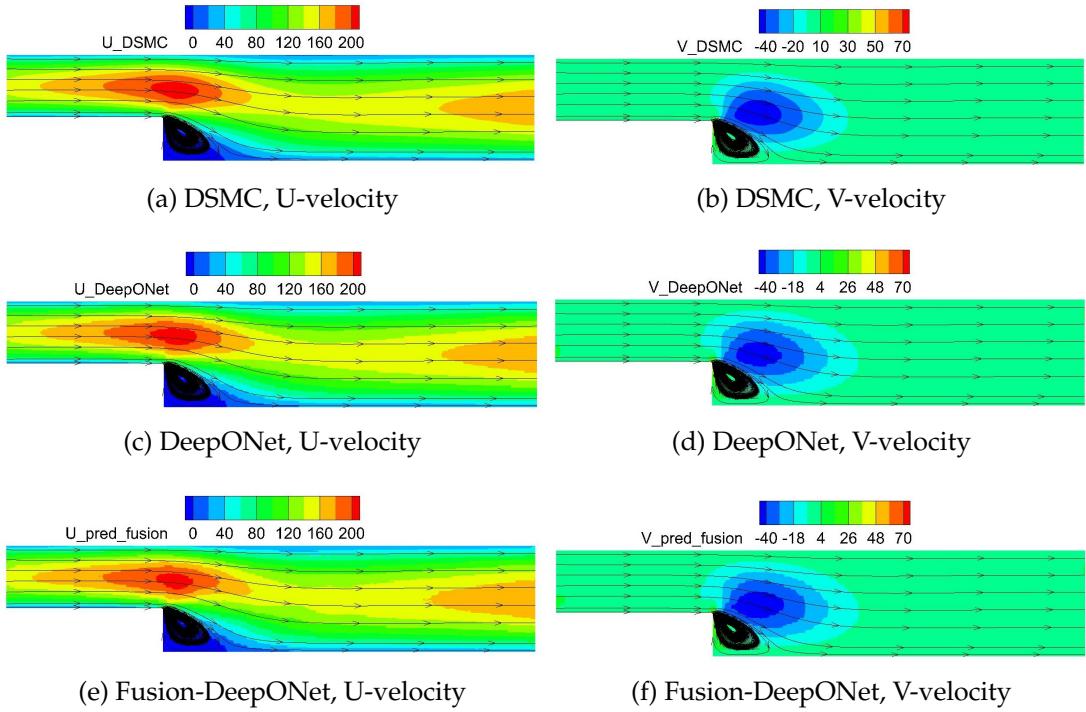


Figure 29: Qualitative comparison of U-velocity (left column) and V-velocity (right column) contours for DSMC (top), DeepONet (middle), and Fusion-DeepONet (bottom).

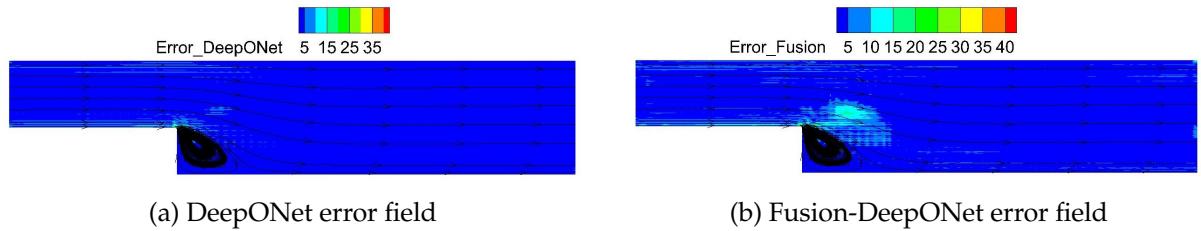


Figure 30: Comparison of velocity error distributions for the height-variation test case ($h/H = 0.44$). The DeepONet approach (a) produces visibly lower error intensity in the shear layer and recirculation region compared to the Fusion-DeepONet (b).

the design of surrogate models in rarefied gas dynamics.

- **Quantification of Computational Savings.** One of the most significant contributions of this work lies in computational efficiency. The trained surrogate is capable of generating predictions for new parametric cases in milliseconds, compared to the hours or days required for full DSMC simulations. This represents several orders-of-magnitude acceleration, thereby enabling many-query tasks such as uncertainty quantification, parametric sweeps, and design optimization that would otherwise be infeasible with DSMC alone.

In summary, the framework delivers both scientific and practical contributions. Scientifically, it introduces a physics-guided loss function that redefines error evaluation by aligning it with physically meaningful flow regions, and it sheds light on the interplay between model complexity and data availability in surrogate modeling. Practically, it establishes a pathway toward orders-of-magnitude acceleration in rarefied flow simulations, bridging the gap between the accuracy of DSMC and the speed required for real-world engineering design.

Beyond the present backward-facing step configuration, the methodology holds promise for a broad class of applications in micro- and hypersonic flows where DSMC remains indispensable yet prohibitively expensive. Future extensions could integrate additional physics-informed constraints, multi-fidelity training strategies, or coupling with uncertainty quantification frameworks, further enhancing the applicability and robustness of surrogate modeling in rarefied gas dynamics.

From a computational perspective, the efficiency gain of the proposed surrogate is striking. For instance, generating a single DSMC solution at low Knudsen numbers in the slip-flow regime typically requires about 24 hours of wall-clock time on a single Intel Core-i7 CPU core. By contrast, training the DeepONet surrogate on a high-end GPU (NVIDIA A100-SXM4-80GB) takes approximately 20–30 minutes. This comparison—“minutes vs. hours”—highlights the orders-of-magnitude acceleration achieved by the surrogate, effectively transforming tasks such as parametric sweeps, optimization, and uncertainty quantification from computationally prohibitive to practically feasible.

Future work will involve exploring adaptive weighting for the zonal loss function, wherein the hyperparameter α is learned during training rather than being fixed. Furthermore, the application of this framework to three-dimensional geometries will be investigated. The proposed surrogate modeling approach will be extended to other challenging rarefied flow problems, such as shockwave-boundary layer interactions in supersonic flows and gas-surface chemistry in atmospheric re-entry vehicles, where localized, non-equilibrium phenomena are critical. Ultimately, this rapid and uncertainty-aware surrogate model can be integrated into a complete multi-fidelity optimization framework for the design of MEMS devices and hypersonic vehicle components, enabling robust design under uncertainty in rarefied gas environments.

Appendix: Hyperparameter Details

9 Hyperparameter Details

For clarity and reproducibility, this section provides a comprehensive summary of the key hyperparameters used for training and model architecture across the four main experimental setups. The specifications are detailed across two tables for improved readability. Table 4 outlines the configurations for the Knudsen number study and the primary height ablation study. Table 5 details the configurations for the simplified height study and the derivative-enhanced Fusion model.

Table 4: Hyperparameter specifications for the Knudsen and primary Height studies.

Hyperparameter	Knudsen Number Study	Height Ablation Study
<i>Training Hyperparameters</i>		
Epochs	1500	1500
Batch Size	512	512
Base Learning Rate	1×10^{-4}	1×10^{-4}
Optimizer	AdamW	AdamW
Weight Decay	1×10^{-5}	5×10^{-5}
Clipnorm	1.0	Not specified
<i>Model Architecture</i>		
Branch Width / Depth	256 / 6	384 / 4
Trunk MLP Width / Depth	256 / 4	384 / 4
Head Width / Depth	512 / 4	768 / 4
Activation Functions	ReLU (CNN), Tanh (MLP)	ReLU (CNN), Tanh (MLP)
<i>Loss Function Details</i>		
Primary Loss Function	Zonal (Two-Zone)	MSE, Zonal, GMSE (Ablation)
Zonal Weight (α)	0.7	0.6
GMSE Parameters	N/A	$\sigma = 10.0, \gamma = 1.0, C_o = 0.2$

Table 5: Hyperparameter specifications for the simplified Height and Fusion studies.

Hyperparameter	Simplified Height Study	Fusion DEL Study
<i>Training Hyperparameters</i>		
Epochs	1500	1500
Batch Size	512	512
Base Learning Rate	1×10^{-4}	1×10^{-4}
Optimizer	AdamW	AdamW
Weight Decay	5×10^{-5}	5×10^{-5}
Clipnorm	1.0	1.0
<i>Model Architecture</i>		
Branch Width / Depth	256 / 4	512 / 6
Trunk MLP Width / Depth	256 / 3	512 / 4
Head Width / Depth	512 / 3	1024 / 4
Activation Functions	ReLU (CNN), Tanh (MLP)	ReLU (CNN), Tanh (MLP)
<i>Loss Function Details</i>		
Primary Loss Function	Zonal (Two-Zone)	Fusion
Zonal Weight (α)	0.6	0.7
Gradient Weight (λ)	N/A	0.05

References

- [1] George Karniadakis, Ali Beskok, and Narayan Aluru. *Microflows and nanoflows: fundamentals and simulation*. Springer, 2005.
- [2] Ehsan Roohi, Hassan Akhlaghi, and Stefan Stefanov. *Advances in Direct Simulation Monte Carlo: From Micro-Scale to Rarefied Flow Phenomena*. Springer Nature Singapore, 2025.
- [3] W. G. Vincenti and C. H. Kruger. *Introduction to Physical Gas Dynamics*. John Wiley and Sons, New York, 1967.
- [4] Graeme A Bird. *Molecular gas dynamics and the direct simulation of gas flows*. Clarendon Press, 1994.
- [5] Amirmehran Mahdavi and Ehsan Roohi. A study on micro-step flow using a hybrid direct simulation monte carlo–fokker–planck approach. *Physics of Fluids*, 34(6):062007, 2022.
- [6] Jonathan CC Lo, Mark C Thompson, Kerry Hourigan, and Jisheng Zhao. A deep learning approach to classifying flow-induced vibration response regimes of an elliptical cylinder. *Physics of Fluids*, 36(4), 2024.
- [7] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [8] George E Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- [9] Lu Lu, Pengzhan Jin, Guofei Pang, Zongren Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021.
- [10] J. Sun, Y. Chen, and X. Tang. Physics-informed neural networks with two weighted loss function methods for interactions of two-dimensional oceanic internal solitary waves. *Journal of Systems Science and Complexity*, 37(2):545–566, 2024.
- [11] S. Subramanian, R. M. Kirby, M. W. Mahoney, and A. Gholami. Adaptive self-supervision algorithms for physics-informed neural networks. In *Proceedings of the 26th European Conference on Artificial Intelligence (ECAI 2023)*, pages 2234–2241. IOS Press, 2023.
- [12] C. Si and M. Yan. Convolution-weighting method for the physics-informed neural network: A primal-dual optimization perspective, 2025. preprint.
- [13] V. S. K. Malineni and S. Rajendran. Physics-informed neural network approaches for sparse data flow reconstruction of unsteady flow around complex geometries, 2025. preprint.
- [14] Seongun Choi, Jeonghu Lee, Yeulwoo Kim, and Sangho Lee. Machine learning-based prediction of turbulent flows over backward-facing steps with varying step angles using large eddy simulation data. *Physics of Fluids*, 37(6):065172, 2025.
- [15] S. Barwey, H. Kim, and R. Maulik. Interpretable a-posteriori error indication for graph neural network surrogate models. *Computer Methods in Applied Mechanics and Engineering*, 433:117509, 2025.
- [16] Anas Jnini, Harshinee Goordoyal, Sujal Dave, Flavio Vella, Katharine H. Fraser, and Artem Korobenko. Physics-constrained deeponet for surrogate cfd models: A curved

- backward-facing step case. In *Proceedings of the ICLR 2024 Workshop on AI4Differential Equations In Science*, 2024.
- [17] Ehsan Roohi and Ahmad Shoja-Sani. Data-driven surrogate modeling of dsmc solutions using deep neural networks. *Aerospace Science and Technology*, 168:110785, 2026.
 - [18] Ehsan Roohi, Ahmad Shoja-Sani, Bijan Goshayeshi, and Ahmad Peyvan. Learning rarefied gas dynamics with physics-enforced neural networks. *arXiv preprint arXiv:2509.06231*, 2025.
 - [19] Hong Xue and Shuhui Chen. Dsmc simulation of microscale backward-facing step flow. *Microscale Thermophysical Engineering*, 7(1):69–86, 2003.
 - [20] Hong Xue, Bin Xu, Yao Wei, and Jian Wu. Unique behaviors of a backward-facingstep flow at microscale. *Numerical Heat Transfer, Part A: Applications*, 47(3):251–268, 2005.
 - [21] Masoud Darbandi and Ehsan Roohi. Dsmc simulation of subsonic flow through nanochannels and micro/nano backward-facing steps. *International Communications in Heat and Mass Transfer*, 38(10):1443–1448, 2011.
 - [22] Craig White, Matthew K Borg, Thomas J Scanlon, and Jason M Reese. A dsmc investigation of gas flows in micro-channels with bends. *Computers & Fluids*, 71:261–271, 2013.
 - [23] Amir-Mehran Mahdavi, Nam TP Le, Ehsan Roohi, and Craig White. Thermal rarefied gas flow investigations through micro-/nano-backward-facing step: Comparison of dsmc and cfd subject to hybrid slip and jump boundary conditions. *Numerical Heat Transfer, Part A: Applications*, 66(7):733–755, 2014.
 - [24] Amir-Mehran Mahdavi and Ehsan Roohi. Investigation of cold-to-hot transfer and thermal separation zone through nano step geometries. *Physics of Fluids*, 27(7), 2015.
 - [25] Deepak Nabapure and Ram Chandra Murthy K. Dsmc simulation of rarefied gas flow over a 2d backward-facing step in the transitional flow regime: Effect of mach number and wall temperature. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 235(7):825–856, 2021.
 - [26] Abhimanyu Gavasane, Amit Agrawal, and Upendra Bhandarkar. Study of rarefied gas flows in backward facing micro-step using direct simulation monte carlo. *Vacuum*, 155:249–259, 2018.
 - [27] A Manela and L Gibelli. Free-molecular and near-free-molecular gas flows over backward facing steps. *Journal of Fluid Mechanics*, 889:A22, 2020.
 - [28] O Sazhin and A Sazhin. Transonic, supersonic, and hypersonic flow of rarefied gas into vacuum through channels with a forward-or backward-facing step. *Microfluidics and Nanofluidics*, 28(5):32, 2024.
 - [29] D Ben-Adva, G Tatsios, and A Manela. Kinetic description of flow detachment at a smooth micro-step: the near-free-molecular regime. *Theoretical and Computational Fluid Dynamics*, 39(1):11, 2025.
 - [30] Ahmad Peyvan, Varun Kumar, and George Em Karniadakis. Fusion-deeponet: A data-efficient neural operator for geometry-dependent hypersonic and supersonic flows. *arXiv preprint arXiv:2501.01934*, 2025.
 - [31] SK Stefanov. On the basic concepts of the direct simulation monte carlo method. *Physics of Fluids*, 31(6):067104, 2019.

- [32] Ehsan Roohi, Stefan Stefanov, Ahmad Shoja-Sani, and Hossein Ejraei. A generalized form of the bernoulli trial collision scheme in dsmc: Derivation and evaluation. *Journal of Computational Physics*, 354:476–492, 2018.
- [33] Maryam Javani, Ehsan Roohi, and Stefan Stefanov. Symmetrized generalized and simplified bernoulli-trials collision schemes in dsmc. *Computers & Fluids*, 272:106188, 2024.
- [34] Zachary Cooper-Baldock, Paulo E Santos, Russell SA Brinkworth, and Karl Sammut. A generalised novel loss function for computational fluid dynamics. *arXiv preprint arXiv:2411.17059*, 2024.
- [35] Samuel H Rudy and Themistoklis P Sapsis. Output-weighted and relative entropy loss functions for deep learning precursors of extreme events. *Physica D: Nonlinear Phenomena*, 443:133570, 2023.