KF-SPINN: SEPARABLE PHYSICS-INFORMED NEURAL NETWORKS FOR KINETIC–FLUID REGIMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Kinetic equations, such as the BGK model of the Boltzmann equation, describe particle dynamics across both continuum and rarefied regimes within a single partial differential equation (PDE), but the additional microscopic velocity dimension makes classical grid-based solvers computationally expensive. Physics-informed neural network (PINN)-based approaches, such as separable PINNs, offer a promising mesh-free alternative for high-dimensional PDEs. However, existing PINNs for BGK solvers struggle in the low-Knudsen-number (Kn) regime, where rapid kinetic relaxation and sharp, evolving shock layers make training challenging. In this work, we present KF-SPINN, a framework that addresses these difficulties through complementary solution and input representations. Specifically, KF-SPINN (i) structurally encodes fast relaxation through the Chapman–Enskog-inspired solution ansatz, (ii) incorporates multiscale, time-modulated lifting features aligned with shock geometry to capture evolving shock profiles, and (iii) enhances training through conservation and entropy-transport residuals combined with a self-adaptive Charbonnier penalty. KF-SPINN improves robustness across the tested low-Kn regime and reduces mean macroscopic prediction errors by up to two orders of magnitude on 1D3V shock benchmarks relative to the strongest baseline evaluated, while keeping GPU memory requirements and training times modest.

1 INTRODUCTION

Fluid equations fail to accurately describe particle motion once the Knudsen number (Kn) – the ratio of the mean free path to the characteristic length scale of the system – is no longer small. Such breakdown scenarios include hypersonic flows where particle collisions are not frequent near the shocks (Bird, 1994). Kinetic equations, such as the Bhatnagar-Gross-Krook (BGK) model of the Boltzmann equation (Bhatnagar et al., 1954), instead describe the evolution of the statistical distribution of particles under collisions, remaining valid across both continuum and rarefied regimes within a single partial differential equation (PDE) (Villani, 2002).

However, the additional microscopic velocity space increases the dimension of the computational domain, rendering classical approaches computationally prohibitive. For example, a domain of three spatial and three velocity dimensions requires $O(N_x^3 N_v^3)$ degrees of freedom per time step, making grid-based numerical methods memory- and compute-intensive. For a comprehensive review of classical numerical methods for kinetic equations, we refer the reader to (Dimarco & Pareschi, 2014).

Recent neural network-based PDE solvers – exemplified by physics-informed neural networks (PINNs) (Raissi et al., 2019) – have emerged as a promising direction for solving high-dimensional PDEs (Cho et al., 2023; Hu et al., 2024). However, several unique characteristics of kinetic equations cause naive PINN approaches to fail. First, small pointwise errors in the high-velocity regime can be amplified by the velocity moments used to compute macroscopic quantities. This issue was previously mitigated by imposing large weights on the high-velocity regime, either adaptively (Oh et al., 2025) or through a theoretically grounded weighting scheme (Ko et al., 2026).

Second, while previous methods succeed in the high-Kn regime, they struggle to accurately approximate the solution in the low-Kn regime, especially where shocks develop. Rapid relaxation induces temporal stiffness (Pareschi & Russo, 2000), which is challenging for standard PINNs (Ji et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

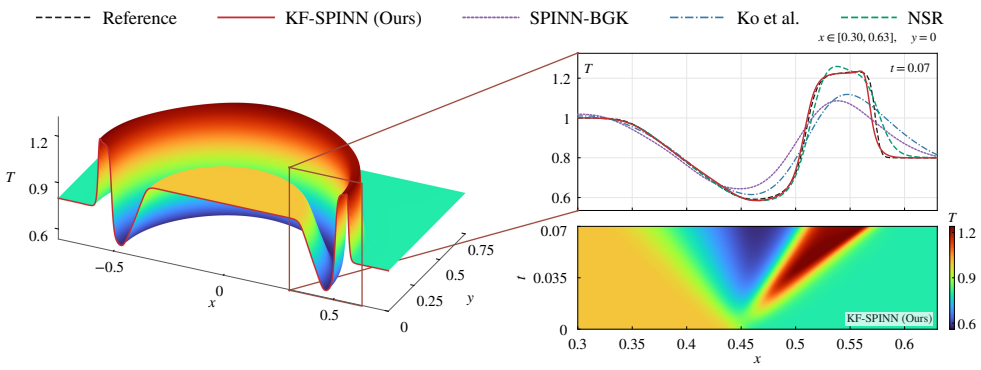


Figure 1: **Resolving kinetic shocks at low Kn in 5D+1D space.** We consider the 2D3V Riemann problem at $\mathrm{Kn} = 10^{-4}$, a time-dependent problem with two spatial and three velocity dimensions. The temperature field predicted by KF-SPINN at $t = 0.07$ is shown over $y \geq 0$ (left), with an enlarged comparison along $y = 0$ near the shock (top right). KF-SPINN shows less temperature overshoot and smearing near the finite-thickness shock layer than existing methods. Temporal evolution of KF-SPINN is shown at the bottom right.

2021). Meanwhile, shock layers become sharper as Kn decreases, for which the spectral bias (Wang et al., 2021), residual imbalance (Liu et al., 2024), and time entanglement problem (Dai et al., 2026) hinder training their evolving profiles. These features of the low-Kn kinetic equation make PINN training particularly challenging.

To address these challenges, we introduce KF-SPINN, a PINN-based method for solving the BGK model in the low-Kn regime, particularly in the presence of shocks. **Our key idea** is to address fast kinetic relaxation through the solution representation and evolving shock geometry through the input representation. Specifically:

- We introduce `CE-ramp`, a relaxation-aware solution representation that mitigates temporal stiffness and improves robustness across a range of small Knudsen numbers.
- We propose shock-aligned lifting features that combine multiple spatial scales with time modulation to capture sharp, evolving shock profiles.
- We complement these representations with a robust, structure-aware training objective that combines conservation and entropy-transport residuals with a self-adaptive Charbonnier penalty to improve macroscopic accuracy in problems involving shocks.

Together, these components establish a new baseline for PINN-based BGK solvers, reducing errors by orders of magnitude in problems involving shocks.

2 Background

BGK model. The BGK model (Bhatnagar et al., 1954), which mitigates the computational cost of the collision operator of the Boltzmann equation, represents the relaxation process of a particle distribution function $f(t, \boldsymbol{x}, \boldsymbol{v})$ towards a statistical equilibrium. The equation reads

$$\partial_t f + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f = \frac{1}{\mathrm{Kn}}(M[f] - f),$$

where

$$M(\boldsymbol{v}; \rho(t,\boldsymbol{x}), \boldsymbol{u}(t,\boldsymbol{x}), T(t,\boldsymbol{x})) = \frac{\rho}{(2\pi T)^{d_v/2}} \exp\left(-\frac{|\boldsymbol{v}-\boldsymbol{u}|^2}{2T}\right)$$

is a local Maxwellian distribution and

$$(\rho,\ \rho\boldsymbol{u}^\top,\ \rho T) = \int \left(1,\ \boldsymbol{v}^\top,\ \frac{1}{d_v}|\boldsymbol{v}-\boldsymbol{u}|^2\right) f(t,\boldsymbol{x},\boldsymbol{v})\, d^{d_v}\boldsymbol{v}$$

are the macroscopic moments. For a fixed t, we say f is in a local equilibrium state if $f = M$ for spatially varying $\rho, \boldsymbol{u}, T$ and a global equilibrium state if $f = M$ for spatially uniform $\rho, \boldsymbol{u}, T$.

Fluid limit and Chapman–Enskog theory. When Kn is small, the collision operator strongly drives f to the local equilibrium, and the balance between the collision and transport operators causes f to approach a specific submanifold of the function space called the hydrodynamic slow manifold (Gorban & Karlin, 2014). Expressing $f = f_{\mathrm{fast}} + f_{\mathrm{slow}}$, the non-hydrodynamic component f_{fast} is expected to decay on an $O(\mathrm{Kn})$ time scale, as can be shown explicitly for the BGK model, and f approaches the slow manifold. In this fluid limit, when the function is sufficiently smooth, the Chapman–Enskog (CE) theory provides an asymptotic expansion of f on the slow manifold (Chapman & Cowling, 1990):

$$f_{\mathrm{slow}} \sim f^{(0)} + \mathrm{Kn} \cdot f^{(1)} + \mathrm{Kn}^2 \cdot f^{(2)} + \cdots \tag{1}$$

where $f^{(i)}$'s are determined by macroscopic moments and their spatio-temporal derivatives. For the BGK model, truncating at $f^{(0)}$ results in $f = M[\rho, \boldsymbol{u}, T]$ (local equilibrium manifold) and the Euler equation, and truncating at $f^{(0)} + \mathrm{Kn} \cdot f^{(1)}$ results in $f = M - \mathrm{Kn}\mathcal{L}_E[M]$ (Navier–Stokes manifold) and the Navier–Stokes equation. For the details, see Appendix C.

3 Related Work

PINNs for the BGK model. Several prior works applied PINNs to the BGK model. Lou et al. (2021) eased the learning difficulty by using two separate neural networks for the relatively easy-to-represent equilibrium part and the small but non-negligible and harder-to-represent non-equilibrium part. Jin et al. (2024) target small-Kn multiscale kinetic problems by learning the macroscopic moments ρ, u, T with separate networks along with f itself, and impose a conservation-law residual to guide the network toward moment conservation. Li et al. (2024) introduce a multi-scale input and scale the non-equilibrium part proportionally to $\sqrt{f_{\mathrm{eq}}}$, using a low-rank structure (SVD, CP decomposition) for an efficient representation.

SPINN-BGK (Oh et al., 2025) builds on this line of work: exploiting the linearity of the moment integral, it introduces a separable network via finite-rank CP decomposition, substantially reducing computational cost, and incorporates inductive biases related to prior work – separating the equilibrium and non-equilibrium parts and scaling the latter with a Gaussian. Ko et al. (2026) introduced a velocity-weighted loss that rigorously controls the errors in f and macroscopic moments, and proposed empirically suitable coefficients for the SPINN-BGK model. For details see Appendix B.1.

PINNs for temporal stiffness and shock. Temporal stiffness associated with widely separated time scales can hinder PINN training. Ji et al. (2021) alleviate this difficulty by incorporating the underlying fast–slow structure into the model formulation. To address the slow learning of high-frequency components in PINNs due to Spectral bias, Wang et al. (2021) introduce multiscale random Fourier feature embeddings. Liu et al. (2024) introduce compression-based weighting that reduces the contribution of shock region that otherwise hinder convergence in smooth regions. Beyond these optimization issues, TINNs (Dai et al., 2026) address the difficulty of representing evolving spatial structures with shared parameters by introducing time-dependent network parameterizations.

Motivated by these studies, KF-SPINN jointly addresses temporal stiffness and evolving shock layers in PINNs for the BGK model, enabling accurate shock resolution in the low-Kn regime at modest computational cost.

4 Method

Learning BGK solutions in the low-Kn regime requires resolving both rapid kinetic relaxation and sharp spatial transitions near shocks. KF-SPINN addresses these challenges through a structural bias in the solution representation, feature embedding, and training objective (Figure 2). In the following text, we shall explain and motivate each component in KF-SPINN.

4.1 Relaxation-Aware Solution Representation

The micro-macro representation reads

$$f_\theta^{\mathrm{micro-macro}} = M_\theta + g_\theta, \tag{2}$$

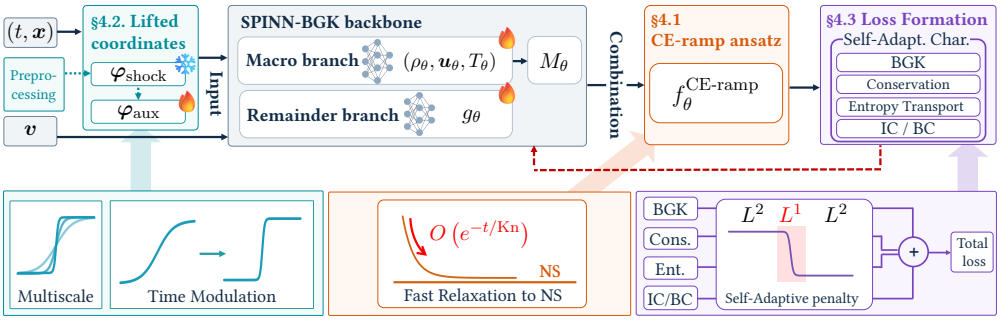


Figure 2: **Overview of the KF-SPINN architecture and training objective.** Lifting features augment $(t, \boldsymbol{x})$ for the SPINN-BGK backbone (Section 4.2), whose macro and remainder branches produce M_θ and g_θ, respectively. The CE-ramp representation combines them into $f_\theta^{\text{CE-ramp}}$ (Section 4.1). The objective applies a self-adaptive Charbonnier penalty to the BGK, conservation, and entropy-transport residuals, together with IC/BC losses (Section 4.3). The red dashed arrow indicates backpropagation.

where M_θ and g_θ aim to approximate the equilibrium and the remainder of the solution f. This provides the leading-order guidance to the network that the solution evolves in the fluid regime with low Kn. The CE theory suggests that g_θ captures the remainder of the $O(\text{Kn})$ scale, when the solution is sufficiently smooth and close to local equilibrium. Motivated by the high-order truncation, we consider the CE representation

$$f_\theta^{\text{CE}} = M_\theta - \lambda \cdot \text{Kn} \cdot \mathcal{L}[M_\theta] + \text{Kn}^p g_\theta, \tag{3}$$

where $(\lambda, p) = (1, 2)$ in theory. The CE representation f_θ^{CE} approximates the first-order term $\text{Kn} \cdot f^{(1)}$ by replacing $\mathcal{L}_E$ in eq. (23) with the transport operator $\mathcal{L} \equiv \partial_t + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}}$. As a consequence, it suffices to approximate the remainder of the $O(\text{Kn}^2)$ scale with $\text{Kn}^p g_\theta$, imposing a more structural bias on f_θ in the low Kn regime.

Compared to the micro-macro decomposition, the CE representation has a more accurate bias for the fluid regime, replacing the local equilibrium manifold with the NS manifold (see Appendix C). However, since certain problems have their initial condition on the local equilibrium manifold rather than the NS manifold, the structured part $M_\theta - \lambda \cdot \text{Kn} \cdot \mathcal{L}[M_\theta]$ of the CE representation has an $O(\text{Kn})$ error at $t = 0$ that rapidly decays. It causes temporal stiffness for the remainder network to learn, which motivates the CE-ramp representation.

The CE-ramp representation is formulated as

$$f_\theta^{\text{CE-ramp}} = M_\theta + \underbrace{e^{-t/\text{Kn}}(f_0 - \mathcal{M}[f_0])}_{\text{turn-off}} - \underbrace{(1 - e^{-t/\text{Kn}})\lambda \cdot \text{Kn} \cdot \mathcal{L}[M_\theta]}_{\text{turn-on}} + \text{Kn}^p g_\theta, \tag{4}$$

where f_0 is the initial condition. Compared to the CE representation, it adds a turn-on ramp on the NS correction, with an additional turn-off ramp to express the decay of the initial nonhydrodynamic layer. With local- equilibrium initial data, the turn-off ramp vanishes. Appendix D.1 shows the motivation and the bound of the remainder part – $O(\text{Kn}^2)$ in local equilibrium initial data and $O(\text{Kn})$ in general initial data.

In practice, we used the SPINN structure for the macroscopic and residual networks as in eq. (2) and eq. (18), and we let λ and p both be learnable. By directly encoding the convergence to the fluid regime, CE-ramp reduces the rapid variation in the target of the remainder network part. Appendices D.2, D.3, D.4 examine the empirical behavior of the representation and its consistency with CE theory.

4.2 Shock-Aligned Lifting Features

Although the CE-ramp representation addresses rapid temporal relaxation in low Kn regimes, resolving sharp spatial transitions near shocks remains a distinct challenge. Kinetic shock layers have finite, varying thickness during transient evolution and become thinner as Kn decreases (Bird, 1994).

To address these evolving spatial scales, we extend the lifting framework of Sun et al. (2026) with multiscale, time-modulated input features. These features and the training objective in Section 4.3 play complementary roles in resolving sharp shock layers.

Lifting method. Consider a solution u that has a sharp transition and define a level set function ψ of the shock, which represents the shock surface (zero level set) as $\Gamma = \{(t, \boldsymbol{x}) : \psi(t, \boldsymbol{x}) = 0\}$ (Osher & Sethian, 1988). Let $\varphi(t, \boldsymbol{x}) = 2(H(\psi(t, \boldsymbol{x})) - 0.5)$, where H is the (smoothed) Heaviside function. Then a solution ansatz connecting smooth functions u_L, u_R is

$$\hat{u}(t, \boldsymbol{x}, \varphi) = \frac{1 - \varphi}{2} u_L(t, \boldsymbol{x}) + \frac{1 + \varphi}{2} u_R(t, \boldsymbol{x}) \tag{5}$$

This formulation places the sharp transition in the supplied coordinate φ, leaving a smooth function $\hat{u}$ to be approximated in the lifted space by neural networks (Sun et al., 2026).

Multiscale features and Time modulation. TAL-PINN (Zhu et al., 2025) used a lifting method reflecting the shock width. Motivated by this adaptive embedding, we incorporate a discrete scale bank $\{\kappa_\ell\}_\ell$ into smoothed Heaviside functions, allowing the neural network to combine various levels of sharpness. We further introduce normalized polynomial time features $T_p(t) = (t/T)^p$ to modulate the contribution of each spatial scale over time, yielding a time-dependent embedding $H(t, y)$. Placing this modulation outside the smoothed Heaviside function avoids the sensitive derivatives.

Level set construction. For the one-dimensional Riemann problem considered here, let x_d denote the initial shock location and c_{shock} the shock speed obtained from the corresponding Euler Riemann solution (Toro, 2013). Then a level set will be $\psi_{\text{shock}}(t, x) = x - (x_d + c_{\text{shock}} t)$. Table 3 shows that shock alignment is crucial for accuracy and that adding coordinates $\psi_j(t, x) = \psi_{\text{shock}}(0, x) - c_j t$ based on the contact and rarefaction-edge speeds provides further gains. Importantly, these gains persist with randomly sampled auxiliary speeds when c_{shock} is retained (Table 15). Motivated by this flexibility, we preserve a dedicated shock level set and introduce learnable auxiliary coordinates in multiple dimensions:

$$\psi_j(t, \boldsymbol{x}) = \psi_{\text{shock}}(0, \boldsymbol{x}) - t c_{\eta_j}(t, \boldsymbol{x}) \tag{6}$$

where c_{η_j} are trainable. To obtain ψ_{shock}, we introduce a *two-stage procedure*: (1) train a network without lifting, (2) extract shock-surface points from its predictions with an algorithm inspired by Canny edge detection (Fujimoto et al., 2019; Canny, 1986), and (3) fit a level-set function ψ_{shock} using Implicit Geometric Regularization (Gropp et al., 2020). We then (4) train a second-stage network using both frozen ψ_{shock} and learnable auxiliary features. Construction details and supporting ablations appear in Appendices E and G.2, respectively.

Combining these methods, for each level set ψ_j including ψ_{shock}, we embed

$$\varphi_{j,p,\ell}(t, \boldsymbol{x}) = H_{p,\ell}(t, \psi_j(t, x)), \quad H_{p,\ell}(t, y) = T_p(t) \tanh(\kappa_\ell y) \tag{7}$$

and the resulting features $\{\varphi_{j,p,\ell}\}$ and $\boldsymbol{x}, t, t/T$ are provided to the neural network as the lifted coordinates.

4.3 Robust and Structure-Aware Training Objective

Since PINN solutions only approximately satisfy the governing equations, adding loss terms derived from these equations can improve training (Yu et al., 2022; Lee et al., 2025). We therefore supplement the BGK residual with a macroscopic conservation residual (Jin et al., 2024) and further introduce an entropy-transport residual implied by exact conservation. We complement these structural constraints with a self-adaptive Charbonnier penalty to mitigate the dominance of large residuals near shocks, which can hinder learning in smooth regions under the standard L^2 loss (Liu et al., 2024). Section 5.2 evaluates the contribution of each component.

We begin with the BGK residual

$$R_{\text{BGK}} = \partial_t f + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f - \frac{1}{\text{Kn}}(M[f] - f). \tag{8}$$

Macroscopic conservation residual. Enforcing macroscopic conservation in a neural approximation is challenging. APNN (Jin et al., 2024) multiplies eq. (8) by the collision invariants and integrates over velocity, removing the collision term. The resulting conservation residual is

$$R_{\text{cons}} = \partial_t \boldsymbol{U} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{F}, \tag{9}$$

where $\boldsymbol{U}$ and $\boldsymbol{F}$ are the conserved moments and kinetic fluxes, respectively. Their definitions and the derivation are provided in Appendix F.1. Empirically, the conservation residual modestly improves accuracy on smooth problems and substantially on Riemann problems.

Entropy-transport residual. Even with the conservation residual, errors persist near contact waves, which transport a jump in thermodynamic entropy $s = \log(T^{3/2}/\rho) + s_0$ (Toro, 2013). To emphasize this structure, we introduce the residual explicitly aligned to thermodynamic entropy transport:

$$R_{\text{Ent}} = (\frac{|\boldsymbol{u}|^2}{2} - \frac{5}{2}T, -\boldsymbol{u}^\top, 1) \cdot (R_\rho, R_{\boldsymbol{m}}, R_E) \tag{10}$$

Although R_{Ent} is implied by exact conservation, its explicit penalization provides a targeted optimization signal for entropy transport with negligible additional cost. See Appendix F.2 for its derivation and thermodynamic interpretation.

Self-adaptive Charbonnier loss. To better capture sharp transition layers, we introduce a self-adaptive variant of the Charbonnier penalty (Charbonnier et al., 1994), with its transition scale τ determined by the mean absolute residual:

$$\ell_{\text{ch}}(r) = \sqrt{r^2 + \tau^2} - \tau, \quad \tau = h \cdot \texttt{stop_gradient}(\overline{|r|}), \tag{11}$$

where $\overline{|r|}$ denotes the mean absolute value over all entries of each residual tensor and h is a sensitivity constant. The penalty combines quadratic sensitivity to small residuals with linear growth for large residuals relative to τ (Barron, 2019), providing robustness.

Finally, KF-SPINN, which is built on top of SPINN-BGK for computational efficiency, is trained by minimizing

$$L(\theta) = \sum_{k \in \mathcal{K}} \lambda_k \mathbb{E}[\ell_{\text{ch}}(R'_k)] \tag{12}$$

where $\mathcal{K} = \{\text{BGK,Cons,Ent, BC, IC}\}$. Following SPINN-BGK, we normalize each residual R_k to R'_k as described in Appendix F.3.

5 Results

Following SPINN-BGK, we use the Lion optimizer (Chen et al., 2023). KF-SPINN normalizes the time input and uses a cosine learning-rate schedule preceded by 5K iterations of linear warmup. For Riemann problems, we enforce the hard initial condition. By default, models are trained for 100k iterations on smooth problems and 200k iterations on Riemann problems, with the same iteration budget for the compared methods. Unless otherwise stated, we follow the benchmark configurations of SPINN-BGK.

We evaluate the macroscopic fields against reference solutions computed using the high-order conservative semi-Lagrangian method (Cho et al., 2021). We report relative L^2 errors for density and temperature; for velocity, we add one to the normalization denominator to avoid division by zero. Detailed problem definitions, model settings, and reference-solver configurations are provided in Appendices A, B, and H, respectively.

5.1 Main Results

In this section, we compare KF-SPINN to previous PINN-based methods for the BGK model: SPINN-BGK (Oh et al., 2025), Ko et al. (2026), NSR (Li et al., 2024) on one- and two-dimensional Riemann problems in the low-Kn regime to evaluate KF-SPINN's ability to resolve temporal stiffness and sharp shock layers. Unless stated otherwise, each model is trained for 200k iterations.

Table 1: Mean macroscopic relative L^2 error for the Riemann problems.

Problem	Kn	SPINN-BGK (Oh et al., 2025)	Ko et al. (2026)	NSR (Li et al., 2024)	**KF-SPINN**
1D3V	1e-04	4.19e-02	3.92e-02	3.23e-02	**6.35e-04**
	3e-04	3.93e-02	3.55e-02	2.37e-02	**2.22e-04**
	5e-04	3.79e-02	3.31e-02	1.95e-03	**1.68e-04**
	7e-04	3.50e-02	3.18e-02	6.10e-04	**1.20e-04**
	1e-03	3.35e-02	2.86e-02	1.65e-04	**1.03e-04**
	Runtime 200k (100k)	486s (293s)	486s (293s)	3800s (1900s)	1550s
	Memory Peak	574MiB	574MiB	4568MiB	966MiB
2D3V	1e-04	5.05e-02	4.84e-02	2.21e-02	**9.16e-03**
	3e-04	5.94e-02	4.60e-02	1.83e-02	**5.30e-03**
	5e-04	5.93e-02	4.60e-02	1.06e-02	**4.98e-03**
	7e-04	5.33e-02	5.13e-02	5.71e-03	**5.08e-03**
	1e-03	4.57e-02	5.11e-02	3.70e-03	**3.22e-03**
	Runtime 200k (100k)	1180s (590s)	1088s (544s)	2h (1h)	863s (total)
	Memory Peak	1938MiB	1938MiB	8666MB	1790MiB

1D3V Riemann. We consider the Sod shock-tube problems over $\mathrm{Kn} \in [10^{-4}, 10^{-3}]$ and evaluate the macroscopic fields at $t = 0.1$. Table 1 reports the mean macroscopic error averaged over five runs. KF-SPINN achieves the lowest error at every tested Kn, maintaining errors below 10^{-3} throughout the range.

2D3V Riemann. We next evaluate the two-stage method on the 2D3V Riemann problems. Since stage 1 only needs to locate the shock, we keep it lightweight by training a micro-macro model without lifting for 50k training iterations. We then fit the shock level-set function using 20k IGR iterations and train stage 2 with the CE-ramp ansatz and lifting features for 100k iterations. Table 1 shows the errors at $t = 0.07$, averaged over five independent runs of the full pipeline. KF-SPINN achieves the lowest error across the tested Kn. Figure 1 provides a solution profile at $\mathrm{Kn} = 10^{-4}$.

Computing cost. The training time and peak GPU memory measured on an NVIDIA A100-SXM4-40GB GPU are shown in Table 1. For the two-stage pipeline, runtime includes both training stages and level-set construction, while peak memory is the maximum observed over the complete sequential execution. Accuracy is evaluated after 200k iterations, except for 2D3V KF-SPINN, which follows the two-stage training schedule described above. We additionally report runtimes at 100k iterations for comparison with the original SPINN-BGK settings. In the 1D3V problem, CE-ramp and lifting incur additional computational cost under the baseline sampling configuration. To address this issue, in the 2D3V problem, we sample small sets of collocation points jointly in $(t, \boldsymbol{x})$, instead of the tensor-product grid used for separable representation. Further details are provided in Appendix B. With these settings, KF-SPINN achieves the lowest error among the compared methods while keeping GPU memory requirements and training times modest.

5.2 Ablation Study

We examine the contributions of the CE-ramp representation, lifting features, and training objective. The CE-ramp representation is evaluated on a 1D3V smooth problem to isolate temporal stiffness, while the lifting features and training objective are assessed on a 1D3V Riemann problem. The lifting ablations use $\mathrm{Kn} = 10^{-4}$ and $t = 0.1$. For the self-adaptive Charbonnier penalty, we set $h = 10^5 \times \mathrm{Kn}$.

Table 2: Mean macroscopic relative L^2 error of the 1D3V Smooth Problem

Method	Kn=1e-03	7e-04	5e-04	3e-04	1e-04
SPINN-BGK (Backbone)	6.95e-04	5.97e-04	4.90e-04	4.01e-04	2.70e-04
KF-SPINN w/ micro-macro	7.96e-04	5.82e-04	5.01e-04	3.28e-04	7.02e-04
KF-SPINN w/ CE	6.32e-04	4.74e-04	3.38e-04	2.17e-04	9.58e-05
KF-SPINN (w/ CE-ramp)	**3.84e-05**	**5.12e-05**	**3.50e-05**	**4.30e-05**	**5.24e-05**

Table 4: Mean macroscopic relative L^2 error of the 1D3V Riemann Problem.

Cons	Ent	Penalty	Kn=1e-03	7e-04	5e-04	3e-04	1e-04
		L^2	2.52e-03	8.40e-03	1.37e-02	1.81e-02	2.09e-02
		Charbonnier	2.16e-04	3.66e-03	3.66e-02	5.53e-02	6.23e-02
✓		Charbonnier	1.19e-04	1.56e-04	2.22e-04	4.08e-04	1.98e-03
✓	✓	L^1	9.34e-04	8.09e-04	8.02e-04	9.41e-04	1.50e-02
✓	✓	L^2	**1.00e-04**	**1.13e-04**	1.81e-04	1.70e-03	6.13e-03
✓	✓	Charbonnier	1.03e-04	1.20e-04	**1.68e-04**	**2.22e-04**	**6.35e-04**

CE-ramp representation. We first verify the CE-ramp representation on the 1D3V smooth problem (Oh et al., 2025, Section 4.1), whose initial condition is a local Maxwellian with smoothly varying macroscopic variables. Shock-capturing lifting is not needed for this problem, so we only examined the effect of the CE-ramp representation and the conservation-loss structure for temporal stiffness. The result is presented in Table 2. KF-SPINN (w/ CE-ramp) attains the lowest mean macroscopic error across all tested Kn values. Furthermore, table 18 shows that CE-ramp also reduces the mean macroscopic error on the 1D3V Riemann problem, demonstrating its benefit even in the presence of shocks.

Importance of lifting, especially with shock geometry. Table 3 shows the benefits of lifting and the particular importance of shock alignment. Without lifting, the mean macroscopic error is 2.49×10^{-2}. Incorporating features based on all four Euler wave speeds (Toro, 2013) reduces this error to 8.16×10^{-4}. The shock-aligned features play a central role: using them alone achieves the lowest error among the single-speed configurations (**bold**), while excluding them from the full configuration raises the error to 1.38×10^{-2}(dagger †). Removing any one of the remaining features changes the error only slightly. These results support using lifting features to improve accuracy, with explicit shock geometry as their most important individual component.

Table 3: Mean macroscopic relative L^2 error of the 1D3V Riemann Problem at $\mathrm{Kn} = 10^{-4}$.

c_1	c_2	c_3	c_{shock}	rel. L^2 error
				2.49e-02
✓				2.86e-02
	✓			2.35e-02
		✓		1.68e-02
			✓	**6.13e-03**
	✓	✓	✓	9.32e-04
✓		✓	✓	7.73e-04
✓	✓		✓	8.54e-04
✓	✓	✓		1.38e-02†
✓	✓	✓	✓	8.16e-04

Components of training objective. Table 4 shows that the benefits of the training objective in Section 4.3 become more evident as Kn decreases. Without the conservation and entropy-transport residuals, the error rises sharply in the low-Kn regime. Adding the conservation residual reduces this error, but it remains on the order of 10^{-3} at $\mathrm{Kn} = 10^{-4}$. Including the entropy-transport residual further reduces the error to the order of 10^{-4} and improves the predicted profiles near the contact and shock waves (see Figure 13). The self-adaptive Charbonnier penalty also improves accuracy over both L^1 and L^2 penalties at smaller Kn, while performing comparably to L^2 at larger Kn. These results support combining explicit macroscopic constraints with a robust residual penalty for low-Kn Riemann problems.

6 Conclusion

In this work, we presented KF-SPINN that resolves temporal stiffness and shocks in the low Kn regime of the BGK model. First, the theory-motivated CE-ramp solution representation resolves temporal stiffness caused by fast relaxation. Second, the lifting input, incorporating time modulation and multiscale features, captures sharp and time-varying shock layers. Finally, the shock-robust and structure-aware training objective mitigates optimization difficulties. Overall, KF-SPINN achieved significant accuracy improvements compared to the previous state-of-the-art PINN approaches, and pushed the boundary of the applicability of the neural network approaches for the BGK model.

Limitations and Future work. This work focused on the BGK model, which recovers the compressible Euler and Navier–Stokes equations as $\mathrm{Kn} \to 0$. We expect extending our method to a broader class of kinetic equations, such as the ellipsoidal-BGK model, would be straightforward.

Although most hyperparameter settings are shared across experiments (Appendix B), KF-SPINN remains sensitive to choices such as CE-coefficient initialization. The smooth-problem study in Appendix D.2 provides a starting point for developing broader parameter-selection guidelines. Our shock benchmarks comprise a one-dimensional shock tube and a radially symmetric two-dimensional Riemann problem; extending the two-stage construction to more general geometries and interacting shocks may introduce additional challenges for shock extraction and feature alignment. Beyond the temporal stiffness and sharp shock layers addressed here, future work could also explore other kinetic–fluid phenomena, including Knudsen layers (Bird, 1994).

Further work is needed to explain the mechanisms underlying the observed improvements and establish rigorous convergence guarantees as mentioned in Appendix D.3.

AI USE STATEMENT

We used large language models (LLMs) to improve the grammar, vocabulary, and presentation of the manuscript, assist with literature search and identify relevant prior work. We also used LLMs for programming to compare our methods with other approaches for which official implementations were not available, as well as to rapidly prototype our ideas. We manually verified the implementations to ensure that the proposed ideas were correctly implemented. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we will make the code publicly available upon acceptance. We provide detailed hyperparameter settings in Appendix B. The same GPU was used throughout each experiment, although different experiments may have used different GPUs. Numerical variability across GPUs may lead to minor differences in the reported errors, even under otherwise identical settings. However, all computational cost measurements were performed on the same GPU to ensure a consistent comparison.

REFERENCES

Jonathan T Barron. A general and adaptive robust loss function. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4326–4334. IEEE, 2019.

Mounir Bennoune, Mohammed Lemou, and Luc Mieussens. Uniformly stable numerical schemes for the Boltzmann equation preserving the compressible Navier–Stokes asymptotics. *Journal of Computational Physics*, 227(8):3781–3803, 2008.

P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. I. small amplitude processes in charged and neutral one-component systems. *Physical Review*, 94(3): 511–525, 1954. doi: 10.1103/PhysRev.94.511.

Graeme A Bird. *Molecular gas dynamics and the direct simulation of gas flows*. Oxford university press, 1994.

John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

Carlo Cercignani. The Boltzmann equation. In *The Boltzmann equation and its applications*, pp. 40–103. Springer, 1988.

Sydney Chapman and Thomas George Cowling. *The Mathematical Theory of Non-Uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases*. Cambridge University Press, 1990. ISBN 9780521408448.

Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st international conference on image processing*, volume 2, pp. 168–172. IEEE, 1994.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36:49205–49233, 2023.

Junwoo Cho, Seungtae Nam, Hyunmo Yang, Seok-Bae Yun, Youngjoon Hong, and Eunbyung Park. Separable physics-informed neural networks. *Advances in Neural Information Processing Systems*, 36:23761–23788, 2023.

Seung Yeon Cho, Sebastiano Boscarino, Giovanni Russo, and Seok-Bae Yun. Conservative semi-Lagrangian schemes for kinetic equations Part II: Applications. *Journal of Computational Physics*, 436:110281, 2021.

Chen-Yang Dai, Che-Chia Chang, Te-Sheng Lin, Ming-Chih Lai, and Chieh-Hsin Lai. TINNs: Time-induced neural networks for solving time-dependent pdes. *arXiv preprint arXiv:2601.20361*, 2026.

Giacomo Dimarco and Lorenzo Pareschi. Numerical methods for kinetic equations. *Acta Numerica*, 23:369–520, 2014.

Takeshi R Fujimoto, Taro Kawasaki, and Keiichi Kitamura. Canny-Edge-Detection/Rankine-Hugoniot-conditions unified shock sensor for inviscid and viscous flows. *Journal of Computational Physics*, 396:264–279, 2019.

Alexander N. Gorban and Ilya V. Karlin. Hilbert's 6th problem: Exact and approximate hydrodynamic manifolds for kinetic equations. *Bulletin of the American Mathematical Society*, 51(2): 187–246, 2014. doi: 10.1090/S0273-0979-2013-01439-3.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3789–3799. PMLR, 13–18 Jul 2020.

Zheyuan Hu, Khemraj Shukla, George Em Karniadakis, and Kenji Kawaguchi. Tackling the curse of dimensionality with physics-informed neural networks. *Neural Networks*, 176:106369, 2024.

Weiqi Ji, Weilun Qiu, Zhiyu Shi, Shaowu Pan, and Sili Deng. Stiff-PINN: Physics-informed neural network for stiff chemical kinetics. *The Journal of Physical Chemistry A*, 125(36):8098–8106, 2021.

Shi Jin, Zheng Ma, and Keke Wu. Asymptotic-preserving neural networks for multiscale kinetic equations. *Communications in Computational Physics*, 35(3):693–723, 2024.

Gyounghun Ko, Sung-Jun Son, Seung Yeon Cho, and Myeong-Su Lee. A theory-guided weighted L^2 loss for solving the BGK model via physics-informed neural networks. *arXiv preprint arXiv:2604.04971*, 2026.

Florian Kogelbauer and Ilya Karlin. Exact hydrodynamic manifolds for the linear Boltzmann BGK equation I: Spectral theory. *Continuum Mechanics and Thermodynamics*, 36:1685–1709, 2024. doi: 10.1007/s00161-024-01323-2. arXiv:2301.03069.

Florian Kogelbauer and Ilya Karlin. Exact hydrodynamic manifolds for the linear Boltzmann BGK equation II: Spectral closure. *Continuum Mechanics and Thermodynamics*, 37:45, 2025. doi: 10.1007/s00161-025-01379-8.

Myeong-Su Lee, Jaemin Oh, Dong-Chan Lee, Kangwook Lee, Sooncheol Park, and Youngjoon Hong. Forward and inverse simulation of pseudo-two-dimensional model of lithium-ion batteries using neural networks. *Computer Methods in Applied Mechanics and Engineering*, 438:117856, 2025.

C David Levermore. Moment closure hierarchies for kinetic theories. *Journal of statistical Physics*, 83(5):1021–1065, 1996.

Zhengyi Li, Bin Dong, and Yanli Wang. Learning invariance preserving moment closure model for Boltzmann–BGK equation. *Communications in Mathematics and Statistics*, 11(1):59–101, 2023.

Zhengyi Li, Yanli Wang, Hongsheng Liu, Zidong Wang, and Bin Dong. Solving the Boltzmann equation with a neural sparse representation. *SIAM Journal on Scientific Computing*, 46(2):C186–C215, 2024.

Li Liu, Shengping Liu, Hui Xie, Fansheng Xiong, Tengchao Yu, Mengjuan Xiao, Lufeng Liu, and Heng Yong. Discontinuity computing using physics-informed neural networks. *Journal of Scientific Computing*, 98(1):22, 2024.

Qin Lou, Xuhui Meng, and George Em Karniadakis. Physics-informed neural networks for solving forward and inverse flow problems via the Boltzmann-BGK formulation. *Journal of Computational Physics*, 447:110676, 2021. doi: 10.1016/j.jcp.2021.110676.

Jaemin Oh, Seung Yeon Cho, Seok-Bae Yun, Eunbyung Park, and Youngjoon Hong. Separable physics-informed neural networks for solving the BGK model of the Boltzmann equation. *SIAM Journal on Scientific Computing*, 47(2):C451–C474, 2025.

Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

Lorenzo Pareschi and Giovanni Russo. Implicit-explicit Runge-Kutta schemes for stiff systems of differential equations. *Recent trends in numerical analysis*, 3:269–289, 2000.

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.

Henning Struchtrup. Entropy and the second law of thermodynamics—the nonequilibrium perspective. *Entropy*, 22(7):793, 2020.

Qi Sun, Zhenjiang Liu, Lili Ju, and Xuejun Xu. Lift-and-embed learning method for solving scalar hyperbolic equations with discontinuous solutions. *Journal of Scientific Computing*, 107(3):87, 2026.

Eleuterio F Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

Cédric Villani. A review of mathematical topics in collisional kinetic theory. *Handbook of mathematical fluid dynamics*, 1:71–305, 2002.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021.

Tao Xiong, Juhi Jang, Fengyan Li, and Jing-Mei Qiu. High order asymptotic preserving nodal discontinuous Galerkin IMEX schemes for the BGK equation. *Journal of Computational Physics*, 284:70–94, 2015.

Bokai Yan and Shi Jin. A successive penalty-based asymptotic-preserving scheme for kinetic equations. *SIAM Journal on Scientific Computing*, 35(1):A150–A172, 2013.

Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, 2022.

Yameng Zhu, Weibing Deng, and Ran Bi. A two-stage adaptive lifting PINN framework for solving viscous approximations to hyperbolic conservation laws. *arXiv preprint arXiv:2511.04490*, 2025.

APPENDIX CONTENTS

A Problem Settings . . . 12
A.1 Riemann Problems . . . 12
A.2 Smooth Problems . . . 13
B Detailed Model and Settings . . . 13
B.1 Backbone: SPINN-BGK . . . 13
B.2 Detailed Settings . . . 14
C Chapman-Enskog Theory . . . 14
C.1 Background . . . 14
C.2 Euler Order . . . 16
C.3 Navier-Stokes Order . . . 16
D CE-Ramp Analysis . . . 17
D.1 CE-ramp ansatz Motivation . . . 17
D.2 CE-ramp Experiment . . . 19
D.3 Further Analysis of CE-Ramp . . . 21
D.4 Additional CE-Ramp Experiments . . . 22
E Two-Stage Method Details . . . 23
E.1 Stage-1 prediction . . . 24
E.2 Preprocessing for IGR . . . 24
E.3 Implicit Geometric Representation . . . 25
E.4 Stage-2 training . . . 25
F Loss Construction . . . 26
F.1 Conservation Loss Construction . . . 26
F.2 Entropy Transport Loss Construction . . . 26
F.3 Residual Normalization . . . 27
G Additional Results . . . 27
G.1 Additional Problems . . . 27
G.2 Additional Experiments . . . 27
H Reference Solver . . . 28
H.1 Numerical Method . . . 28
H.2 Refinement Error . . . 30

A PROBLEM SETTINGS

A.1 RIEMANN PROBLEMS

1D Riemann. This is the Sod-type shock-tube problem used in Section 4.2 of Oh et al. (2025) (originally Section 6.2 of Li et al. (2024)). The initial condition is a local Maxwellian with $(\rho_0, u_0, T_0) = (1, 0, 1)$ for $x < 0$ and $(0.125, 0, 0.8)$ for $x \geq 0$, smoothed as $\rho_0(x) = 1 - 0.875H(x)$, $T_0(x) = 1 - 0.2H(x)$ with $H(x) = (1 + \tanh(100x))/2$. The computational

domain is $x \in (-0.5, 0.5)$, $\boldsymbol{v} \in (-10, 10)^3$ (Table 7), with free-flow boundary conditions, evaluated at $t = T_f = 0.1$.

2D Riemann. This is the 2D Riemann problem used in Section 4.4 of Oh et al. (2025) (originally from Cho et al. (2021)). The initial condition is a local Maxwellian with $(\rho_0, u_0, T_0) = (1, 0, 1)$ inside the circle $x^2+y^2 < 0.2$ and $(0.125, 0, 0.8)$ outside, smoothed as $H(r^2) = (1+\tanh(100r^2))/2$ with $r^2 = 0.2 - x^2 - y^2$. The computational domain is $\boldsymbol{x} \in (-1, 1)^2$, $\boldsymbol{v} \in (-6, 6)^3$ (Table 7), with free-flow boundary conditions, evaluated at $t = T_f = 0.07$.

A.2 Smooth Problems

1D3V Smooth. This is the '1D smooth problem' of Section 4.1 in Oh et al. (2025) (originally Section 6.1 of Li et al. (2024)), with initial condition $\rho_0(x) = 1 + 0.5\sin(2\pi x)$, $u_0(x) = 0$, $T_0(x) = 1 + 0.5\sin(2\pi x + 0.2)$. The computational domain is $x \in (-0.5, 0.5)$, $\boldsymbol{v} \in (-10, 10)^3$, with periodic boundary conditions, evaluated at $t = T_f = 0.1$.

1D3V Bimodal(nonequilibrium initial condition). We adapt the two-stream (bimodal) initial condition of Yan & Jin (2013) (their Eqs. (52)–(53)), by extending it from 1D2V to 1D3V and modifying the initial velocity:

$$f_0(x, \boldsymbol{v}) = \tfrac{1}{2}M(\rho_0(x), u_0(x), T_0(x); \boldsymbol{v}) + \tfrac{1}{2}M(\rho_0(x), -u_0(x), T_0(x); \boldsymbol{v}), \tag{13}$$

$$\rho_0(x) = \frac{2 + \sin(2\pi x)}{3}, \qquad u_0(x) = \left(\tfrac{1}{5}\cos(2\pi x), 0, 0\right), \qquad T_0(x) = \frac{3 + \cos(2\pi x)}{4}, \tag{14}$$

The computational domain is also changed to $x \in (-0.5, 0.5)$ and $\boldsymbol{v} \in (-10, 10)^3$ to match the 1D3V Smooth problem's conventions, with periodic boundary conditions, evaluated at $t = T_f = 0.1$.

2D3V Smooth. This is the '2D smooth problem' of Section 4.3 in Oh et al. (2025), with the initial temperature modified from $T_0(x, y) = 1$ to $T_0(x, y) = 1 + 0.5\sin(2\pi x)$, while $\rho_0(x, y) = 1 + 0.5\sin(2\pi x)\sin(2\pi y)$ and $u_0 = 0$ are unchanged. The computational domain is $\boldsymbol{x} \in (-0.5, 0.5)^2$, $\boldsymbol{v} \in (-10, 10)^3$, with periodic boundary conditions, evaluated at $t = T_f = 0.1$.

2D3V Isothermal. This is the original, unmodified '2D smooth problem' of Section 4.3 in Oh et al. (2025): $\rho_0(x, y) = 1 + 0.5\sin(2\pi x)\sin(2\pi y)$, $u_0 = 0$, $T_0 = 1$. Since only the density perturbation drives the dynamics, it can also be identified as a 'density wave'. Domain and boundary conditions are as in 2D3V Smooth above.

B Detailed Model and Settings

B.1 Backbone: SPINN-BGK

Separable physics-informed neural networks. SPINN (Cho et al., 2023) represents a function using the sum of R separable functions:

$$f_\theta(z_1, \ldots, z_d) = \sum_{r=1}^{R} \prod_{k=1}^{d} f_{\theta_k}^{(r)}(z_k) \tag{15}$$

where $f_{\theta_k} : \mathbb{R} \to \mathbb{R}^R$ produces the factors associated with one input axis. It offers two computational advantages that make it suitable for kinetic equations: On tensor-product grid with N points per axis, computing the factors requires only dN network forward evaluations, producing RdN factor values. Moreover, integration with respect to an input axis is simplified to one-dimensional integrals, requiring only $O(RN_{\text{int}})$ operations:

$$\int f_\theta(z_1, \ldots, z_d) dz_i = \sum_{r=1}^{R} \prod_{k=1, k\neq i}^{d} f_{\theta_k}^{(r)}(z_k) \left(\int f_{\theta_i}^{(r)}(z_i) dz_i \right) \tag{16}$$

where N_{int} is the number of quadrature nodes.

SPINN-BGK. Oh et al. (2025) adapt eq. (16) to the BGK Model using a micro–macro decomposition (Bennoune et al., 2008; Xiong et al., 2015)

$$f_\theta^{\text{micro}-\text{macro}} = M_\theta + g_\theta, \qquad M_\theta = M(\boldsymbol{v}; \rho_\theta, \boldsymbol{u}_\theta, T_\theta). \tag{17}$$

The macro branch $(\rho_\theta, \boldsymbol{u}_\theta, T_\theta)$ accepts inputs $(t, \boldsymbol{x})$ while g_θ uses the full set of variables $(t, \boldsymbol{x}, \boldsymbol{v})$:

$$g_\theta(t, \boldsymbol{x}, \boldsymbol{v}) = \sum_{r=1}^{R} f_{\theta_t}^{(r)}(t) \prod_{k=1}^{d_x} f_{\theta_{x_k}}^{(r)}(x_k) \prod_{l=1}^{d_v} f_{\theta_{v_l}}^{(r)}(v_l) \exp\left(-\frac{a^2}{2}(v_l - \mu_l)^2\right) \tag{18}$$

Here, $a > 0$ and $\boldsymbol{\mu}$ are trainable parameters. The Gaussian function promotes velocity-tail decay while preserving separability.

We note that macroscopic moments are obtained by integrating $f_\theta^{\text{micro}-\text{macro}}$, not by solely computing $\rho_\theta, \boldsymbol{u}_\theta, T_\theta$. Explicitly, for general monomial $\boldsymbol{v}^\alpha := \prod_{l=1}^{d_v} v_l^{\alpha_l}$ with multi-index $\alpha = (\alpha_1, \ldots, \alpha_{d_v}) \in \mathbb{N}^{d_v}$, the corresponding macroscopic moment is calculated as

$$\begin{aligned}\int_{\mathbb{R}^{d_v}} \boldsymbol{v}^\alpha f_\theta(t, \boldsymbol{x}, \boldsymbol{v})\, d\boldsymbol{v} &= \rho_\theta \prod_{l=1}^{d_v} \mu_{\alpha_l}(u_{\theta,l}, T_\theta) \\ &+ \sum_{r=1}^{R} f_{\theta_t}^{(r)}(t) \prod_{k=1}^{d_x} f_{\theta_{x,k}}^{(r)}(x_k) \prod_{l=1}^{d_v} \left(\int_{\mathbb{R}} v_l^{\alpha_l} f_{\theta_{v,l}}^{(r)}(v_l) \exp\left(-\frac{a^2}{2}(v_l - \mu_l)^2\right) dv_l \right)\end{aligned} \tag{19}$$

(where $\mu_{\alpha_l}(u, T)$ is α_l-th raw moment of gaussian distribution.)

B.2 Detailed Settings

Baseline settings. We follow SPINN-BGK for many of our settings. Table 5 shows the network settings, Table 6 shows the optimizer settings, Table 7 shows the per-problem training settings, Table 9 shows the lifting settings, and Table 10 shows the loss settings. CE-ramp settings are $\lambda_{\text{init}} = 1$, $p_{\text{init}} = 1$ for 1D3V Riemann problems and $\lambda_{\text{init}} = 0.75$, $p_{\text{init}} = 1$ for 2D3V Riemann problems.

For the smooth-family problems (1D3V Smooth, 1D3V Bimodal, 2D3V Smooth, 2D3V Isothermal), we use the same network and optimizer settings as the Riemann problems, except that the shock-aligned lifting features are not used. The remaining differences (branch-input dimensions, per-problem training configuration, and loss settings) are annotated in Tables 5, 8, and 10.

2D3V Riemann problem settings. For the 2D3V Riemann problem stage 2, we sample collocation points instead of using a tensor-product grid, as lifting method breaks t–$\boldsymbol{x}$ separability of SPINN. We adjust the learning rate for faster convergence while keeping the learning rate for λ and p unchanged to avoid unstable updates, and increase the conservation-loss weight to $(0.03/\text{Kn})/T_f$, emphasizing conservation as Kn decreases.

C Chapman-Enskog Theory

C.1 Background

We first consider the general form of the Boltzmann equation with the classical Boltzmann collision operator $Q[f, f]$:

$$\partial_t f + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f = \frac{1}{\text{Kn}} \mathcal{Q}[f], \qquad \mathcal{Q}[f] \equiv Q[f, f], \tag{20}$$

where $\mathcal{Q}[f]$ is the collision integral, which drives f toward the local Maxwellian: $\mathcal{Q}[f] = 0$ if and only if f is locally Maxwellian (Cercignani, 1988). On the slow manifold, f is completely determined pointwise by the macroscopic moments and their spatial derivatives (which are functions of space and time only), so that the long-term behavior of $f(t, \boldsymbol{x}, \boldsymbol{v})$ can in principle be described by evolution equations of macroscopic moments alone. However, explicit construction of the slow manifold and the macroscopic dynamics is challenging, particularly for nonlinear kinetic equations (Gorban & Karlin, 2014).

Table 5: KF-SPINN network settings for the Riemann and Smooth baselines. Differences from SPINN-BGK are marked with an asterisk *.

Setting	Value
Shared solution-network architecture	
Branch architecture	$d_{\text{in}} \to 128 \to 128 \to 256$
Hidden activation	$\sin(\omega_0 z)$, with $\omega_0 = 10$
Output activation / rank	Linear / $R = 256$
Macroscopic output weight	5×256
Macroscopic output maps	$\rho_\theta = \exp(-a_\rho)$, $\boldsymbol{u}_\theta = (a_{u_1}, a_{u_2}, a_{u_3})$, $T_\theta = \exp(-a_T)$
Time-branch input*	t/T_f
Velocity-branch input	One scalar v_i for each of three branches
Gaussian envelope	Trainable center $\boldsymbol{\mu} \in \mathbb{R}^3$ and scalar precision a; initialized to $\mathbf{0}$ and 1
Branch-input dimensions: macroscopic / kinetic*	
1D3V Riemann	$(1, 63)$ / $(1, 63, 1, 1, 1)$
2D3V Riemann stage 1	$(1, 1, 1)$ / $(1, 1, 1, 1, 1, 1)$
2D3V Riemann stage 2	$(1, 64)$ / $(1, 64, 1, 1, 1)$
1D3V Smooth / Bimodal	$(1, 1)$ / $(1, 1, 1, 1, 1)$
2D3V Smooth / Isothermal	$(1, 1, 1)$ / $(1, 1, 1, 1, 1, 1)$
Initialization of solution branches	
First-layer weights	$\mathcal{U}(-1/d_{\text{in}}, 1/d_{\text{in}})$
Subsequent-layer weights	$\mathcal{U}(-b, b)$, $b = \sqrt{6/d_{\text{in}}}/\omega_0$, using each layer's input width
Branch biases	$\mathcal{U}(-1/\sqrt{d_{\text{in}}}, 1/\sqrt{d_{\text{in}}})$
Macroscopic readout weights	$\mathcal{U}(-\sqrt{6/R}, \sqrt{6/R})$

Table 6: Optimizer settings for the solution networks. Differences from SPINN-BGK are marked with an asterisk *.

Setting	Value
Optimizer	Lion
(β_1, β_2)	$(0.9, 0.99)$
Weight decay	0
Initial learning rate*	0
Peak learning rate (1D3V)	10^{-5}
Peak learning rate (2D3V stage 1)*	3×10^{-5}
Peak learning rate (2D3V stage 2, networks)*	10^{-4}
Peak learning rate (2D3V stage 2, $\log\lambda$, $\log p$)*	10^{-5}
Final scheduled learning rate	0
Warmup*	Linear, 5,000 updates
Decay	Cosine after warmup
Checkpoint selection	Minimum fixed-grid validation loss at the intervals

Moment method. One way of constructing the macroscopic evolution equations is the moment method. Multiplying eq. (20) by $\boldsymbol{v}^\alpha$ and integrating over $\boldsymbol{v}$-space gives a time-evolution equation for the macroscopic moment $\int \boldsymbol{v}^\alpha f(t, \boldsymbol{x}, \boldsymbol{v})\, d\boldsymbol{v}$, but it involves the flux of the next-higher moment, $\int \boldsymbol{v} \otimes \boldsymbol{v}^\alpha f(t, \boldsymbol{x}, \boldsymbol{v})\, d\boldsymbol{v}$. This recursive chain leaves the system of moment evolution equations unclosed.

The moment method terminates this infinite chain by approximating the higher-moment flux (or f itself) in terms of lower-order moments. Higher-order moments often carry direct physical meaning (e.g., heat flux), which gives such approximations physical grounding (Levermore, 1996). Neural moment-closure models are an example of a kinetic neural-network model built on the moment method (Li et al., 2023).

Table 7: KF-SPINN training settings for Riemann problems. Differences from SPINN-BGK are marked with an asterisk *.

Setting	1D3V Riemann	2D3V Riemann (stages 1 / 2)
Final time T_f	0.1	0.07
Spatial domain	$[-0.5, 0.5]$	$[-1, 1]^2$
Velocity domain	$[-10, 10]^3$	$[-6, 6]^3$
Decomposition*	CE-ramp	micro-macro (stage 1) / CE-ramp (stage 2)
Iterations*	200,000	50,000 / 100,000
Training collocation*	$(12, 32, 32, 12, 12)$	$(12, 16, 16/(512), 12, 12, 12)$
Validation collocation*	$(24, 48, 48, 12, 12)$	$(16, 20, 20/(4096), 16, 16, 16)$

Table 8: KF-SPINN training settings for the smooth-family problems.

Setting	1D3V Smooth / Bimodal	2D3V Smooth / Isothermal
Final time T_f	0.1	0.1
Spatial domain	$[-0.5, 0.5]$	$[-0.5, 0.5]^2$
Velocity domain	$[-10, 10]^3$	$[-10, 10]^3$
Decomposition	CE-ramp	CE-ramp
Iterations	100,000	100,000
Training grid	$(12, 32, 32, 12, 12)$	$(12, 12, 12, 12, 12, 12)$
Validation grid (configured)	Same as training grid	Same as training grid

Chapman–Enskog theory. CE theory, on the other hand, takes a different approach: it assumes that f on the slow manifold can be expressed as a series in powers of Kn. Substituting the expansion eq. (1) into eq. (20) gives

$$\partial_t(f^{(0)} + \mathrm{Kn} f^{(1)}) + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}}(f^{(0)} + \mathrm{Kn} f^{(1)}) = \frac{1}{\mathrm{Kn}} \left(\mathcal{Q}[f^{(0)}] + \mathrm{Kn} \cdot D\mathcal{Q}|_{f^{(0)}}[f^{(1)}] + \cdots \right), \quad (21)$$

where $D\mathcal{Q}|_{f^{(0)}}$ denotes the Fréchet derivative of $\mathcal{Q}$ evaluated at $f^{(0)}$.

C.2 Euler Order

At order $O(\mathrm{Kn}^{-1})$, eq. (21) gives $\mathcal{Q}[f^{(0)}] = 0$, so $f^{(0)}$ is a local Maxwellian: $f^{(0)} = M[f^{(0)}]$. By the standard solvability condition of the Chapman–Enskog expansion – that each correction $f^{(k)}$, $k \geq 1$, has zero macroscopic mass, momentum, and energy – the moments (ρ, u, T) of the full distribution f coincide with those of $f^{(0)}$ alone, so

$$f \approx f^{(0)} = M[f^{(0)}] = M[\boldsymbol{v}; \rho(t, \boldsymbol{x}), u(t, \boldsymbol{x}), T(t, \boldsymbol{x})]$$

in the leading order. Examining the time-evolution equation of ρ, u, T at this order yields the Euler equations. The space of functions of the form $f = M[\boldsymbol{v}; \rho(t, \boldsymbol{x}), u(t, \boldsymbol{x}), T(t, \boldsymbol{x})]$ is called the local-equilibrium (or Euler) manifold.

C.3 Navier-Stokes Order

At the next order $O(\mathrm{Kn}^0)$, eq. (21) gives $D\mathcal{Q}|_{f^{(0)}}[f^{(1)}] = \partial_t f^{(0)} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f^{(0)}$. Evaluating the right-hand side through the leading-order (Euler) evolution of (ρ, u, T) and inverting the Fréchet derivative gives the Kn^1-order correction (Bennoune et al., 2008, Sec. 3.2):

$$f \approx M[f] + \mathrm{Kn} \cdot D\mathcal{Q}|^{-1}_{M[f]}[\mathcal{L}_E M[f]]. \quad (22)$$

For a general Boltzmann collision operator, computing the inverse of the Fréchet derivative of the collision operator requires solving a velocity-space integral equation. For the BGK model, this simplifies to

$$f \approx M - \mathrm{Kn} \cdot \mathcal{L}_E[M] + O(\mathrm{Kn}^2), \quad (23)$$

Table 9: KF-SPINN Lifting method settings.

Setting	1D3V Riemann	2D3V Riemann stage 2
Number of level-set channels	4	4
Shock Extraction	$c_{\text{shock}} = 1.8444733671$	IGR
Shock coordinate	$x - c_{\text{shock}}t$	Frozen fitted SDF $\psi_{\text{IGR}}(t, \boldsymbol{x})$
Auxiliary coordinates	$x - c_j t, j = 1, 2, 3$	$\psi_{\text{IGR}}(0, \boldsymbol{x}) - tc_{\eta_j}(t, \boldsymbol{x}), j = 1, 2, 3$
Auxiliary speeds	$\mathcal{U}[-2.5, 2.5)$	Three learnable networks $c_{\eta_j}(t, \boldsymbol{x})$
Sharpnesses κ	$\{10, 25, 50, 100, 200\}$	$\{10, 25, 50, 100, 200\}$
Modulation powers q	$\{0, 1, 2\}$	$\{0, 1, 2\}$
Number of lifted features	$4 \times 5 \times 3 = 60$	$4 \times 5 \times 3 = 60$
Additional spatial-branch inputs	$(x, t, t/T_f)$	$(x_1, x_2, t, t/T_f)$
Total spatial-branch input size	63	64

Table 10: Loss settings for the Riemann baselines and the smooth baselines. Here $m = (\rho, \rho u_1, \rho u_2, \rho u_3, 2E)$ is the stored conserved-moment vector; R_m denotes its conservation residual.

Setting	Riemann (1D3V / 2D3V)	Smooth-family
Initial-condition enforcement	Hard constraint, linear t/T_f gate	Same
$(\lambda_{\text{BGK}}, \lambda_{\text{BC}}, \lambda_{\text{cons}}, \lambda_{\text{ent}})$	$(1, 1, 3/T_f, 0.1/T_f)$ / $(1, 1, (0.03/\text{Kn})/T_f, 1/T_f)$	$(1, 1, 0.01/T_f, 0)$; R_{Ent} not used
Residual penalty	Self-Adaptive Charbonnier	Standard relative L^2
Adaptive transition scale	$\tau = h \cdot \texttt{stop_gradient}(\overline{\lvert r \rvert})$	N/A (no Charbonnier)
Sensitivity h	10^5Kn	N/A (no Charbonnier)
BGK normalization	$R_{\text{BGK}}/(\lvert f_\theta \rvert + 10^{-3})$	Same
Conservation normalization	$T_f R_m/(1 + \lvert m \rvert)$, componentwise	Same
Entropy normalization	$T_f R_{\text{Ent}}/[\max(\rho, 10^{-3}) \max(T, 10^{-3})]$	N/A (unused)
Boundary normalization	Relative to the boundary distribution magnitude plus 10^{-3}	N/A (periodic BC)

where $\mathcal{L}_E \equiv \partial_{t,E} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}}$, and $\partial_{t,E}$ evaluates time derivatives using the leading-order Euler equations:

$$\partial_{t,E}\rho \equiv -\nabla_{\boldsymbol{x}} \cdot (\rho u)$$
$$\partial_{t,E}u \equiv -(u \cdot \nabla_{\boldsymbol{x}})u - \frac{T}{\rho}\nabla_{\boldsymbol{x}}\rho - \nabla_{\boldsymbol{x}}T$$
$$\partial_{t,E}T \equiv -(u \cdot \nabla_{\boldsymbol{x}})T - \frac{2}{d_v}T(\nabla_{\boldsymbol{x}} \cdot u).$$

With the substitution above, $\partial_{t,E}M$ contains no explicit time derivatives, and becomes a function of (ρ, u, T) and their spatial derivatives only. The space of the function f that is expressed by eq. (22), or eq. (23) (in the BGK model) is called the first-order approximation of hydrodynamic slow manifold, or Navier–Stokes manifold in this paper (Gorban & Karlin, 2014).

D CE-Ramp Analysis

D.1 CE-ramp ansatz Motivation

Motivation for the turn-on term. Consider the constant-Kn BGK equation, $\mathcal{L}f = \frac{1}{\text{Kn}}(M[f] - f)$, and suppose first a local-equilibrium initial condition $f(0) = M(0) \equiv M_0$. Define $M(t) \equiv M[f(t)]$, and

$$f_{\text{NS}} \equiv M - \text{Kn}\,\mathcal{L}_E[M], \qquad \Phi \equiv \frac{f - f_{\text{NS}}}{\text{Kn}}, \qquad \Phi(0) = \mathcal{L}_E[M_0].$$

Substituting $f = M - \text{Kn}\,\mathcal{L}_E[M] + \text{Kn}\Phi$ into the BGK equation and rearranging gives the exact equation

$$\text{Kn}\,\partial_t\Phi + \Phi = (\mathcal{L}_E[M] - \mathcal{L}M) + \text{Kn}\,\mathcal{L}(\mathcal{L}_E[M]) - \text{Kn}\,\boldsymbol{v} \cdot \nabla_{\boldsymbol{x}}\Phi \equiv \text{Kn}\,Q(t). \tag{24}$$

Assume that, for sufficiently small Kn and a smooth initial condition, $\mathcal{L}_E[M] - \mathcal{L}M = O(\mathrm{Kn})$ and $Q(t) = O(1)$ uniformly in t, with constants independent of Kn. The Duhamel formula then gives

$$\Phi(t) = e^{-t/\mathrm{Kn}}\mathcal{L}_E[M_0] + \int_0^t e^{-(t-s)/\mathrm{Kn}} Q(s)\, ds = e^{-t/\mathrm{Kn}}\mathcal{L}_E[M_0] + O(\mathrm{Kn}). \tag{25}$$

The $O(\mathrm{Kn})$ error contributes only $O(\mathrm{Kn}^2)$ to f. Further assuming $\mathcal{L}_E[M(t)] - \mathcal{L}_E[M_0] = O(t)$, replacing the $\mathcal{L}_E[M_0]$ term by $\mathcal{L}_E[M(t)]$ costs only $\mathrm{Kn}\, e^{-t/\mathrm{Kn}} \|\mathcal{L}_E[M(t)] - \mathcal{L}_E[M_0]\| \le C_t\, \mathrm{Kn}\, t\, e^{-t/\mathrm{Kn}} \le (C_t/e)\, \mathrm{Kn}^2$, and replacing $\mathcal{L}_E[M(t)]$ by $\mathcal{L}[M(t)]$ costs another $\mathrm{Kn} \cdot O(\mathrm{Kn}) = O(\mathrm{Kn}^2)$. Therefore

$$f(t) = M(t) - \mathrm{Kn}(1 - e^{-t/\mathrm{Kn}})\, \mathcal{L}[M(t)] + O(\mathrm{Kn}^2). \tag{26}$$

This is the turn-on term of eq. (4). The assumptions in this paragraph were checked with the reference solution of 1D3V smooth problems, as shown in Figure 3. For $\mathrm{Kn} \sim 10^{-4}$, the time discretization of the reference solver is $\Delta t \sim \mathrm{Kn}$, so results from $\mathrm{Kn} < 5 \times 10^{-4}$ are omitted.

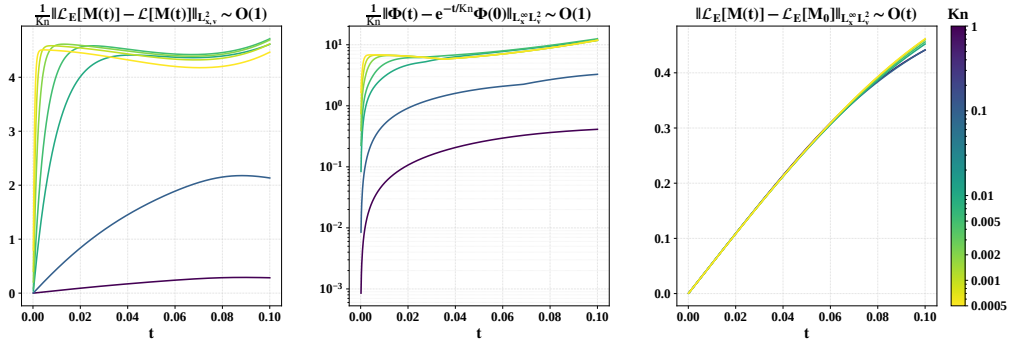


Figure 3: Numerical checks of the assumptions (turn-on)

Motivation for the turn-off term. Now suppose a nonequilibrium initial condition $f(0) = f_0$. Again define $M(t) \equiv M[f(t)]$, $h \equiv f - M$, $h(0) = h_0$, and

$$\left(\mathcal{L} + \frac{1}{\mathrm{Kn}}\right) h = -\mathcal{L}M. \tag{27}$$

Define the transport operator $(S_\tau[g])(\boldsymbol{x}, \boldsymbol{v}) \equiv g(\boldsymbol{x} - \boldsymbol{v}\tau, \boldsymbol{v})$. The Duhamel formula along characteristics gives

$$h(t) = e^{-t/\mathrm{Kn}} S_t[h_0] - \int_0^t e^{-(t-s)/\mathrm{Kn}} S_{t-s}[\mathcal{L}M(s)]\, ds. \tag{28}$$

Since the initial condition is fixed independently of Kn, $\boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} h_0$ is independent of Kn. Assume further that $\mathcal{L}M$ is also uniformly bounded independently of Kn, in $L^2_{x,v}$. The integral term is then $O(\mathrm{Kn})$, so

$$h(t) = e^{-t/\mathrm{Kn}} S_t[h_0] + O(\mathrm{Kn}). \tag{29}$$

Since $\|S_t[h_0] - h_0\|_{L^2_{x,v}} \le t\, \|\boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} h_0\|_{L^2_{x,v}}$, replacing $S_t[h_0]$ term by the frozen value costs only

$$e^{-t/\mathrm{Kn}} \|S_t[h_0] - h_0\|_{L^2_{x,v}} \le Ct\, e^{-t/\mathrm{Kn}} \le (C/e)\, \mathrm{Kn}.$$

Therefore we have

$$f(t) = M(t) + e^{-t/\mathrm{Kn}}\big(f_0 - M[f_0]\big) + O(\mathrm{Kn}), \tag{30}$$

uniformly. This is exactly the turn-off term of eq. (4). Incorporating the $O(\mathrm{Kn})$ turn-on correction motivated above preserves this error order:

$$f(t) = M[f(t)] + e^{-t/\mathrm{Kn}}\big(f_0 - M[f_0]\big) - \mathrm{Kn}(1 - e^{-t/\mathrm{Kn}})\mathcal{L}[M[f(t)]] + O(\mathrm{Kn}), \tag{31}$$

which is the motivation for the CE-ramp ansatz.

The assumptions in this paragraph were checked using the reference solution for the 1D3V bimodal problem, as shown in Figure 4. Here we plotted the stronger $\|\cdot\|_{L^\infty_x L^2_v}$ norm.

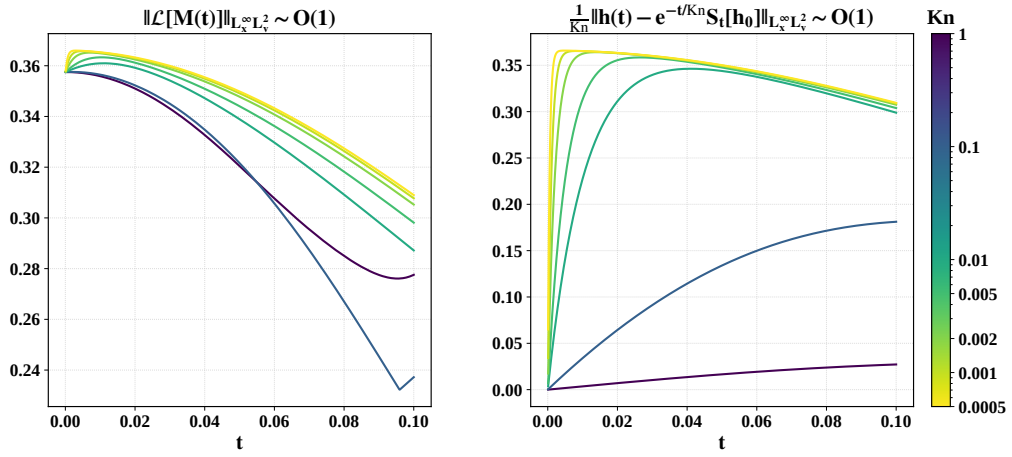


Figure 4: Numerical checks of the assumptions (turn-off)

Behavior of the target functions. We define the target functions $R^{(1)}(t)$, $R^{(2)}(t)$, $R^{(2)}_{\mathrm{ramp}}(t)$

$$R^{(1)}(t) = f(t) - M[f(t)] \quad \text{(micro-macro)} \tag{32}$$

$$R^{(2)}(t) = f(t) - (M[f(t)] - \mathrm{Kn} \cdot \mathcal{L}[M[f(t)]]) \quad \text{(CE)} \tag{33}$$

$$R^{(2)}_{\mathrm{ramp}}(t) = f(t) - \left(M[f(t)] - \mathrm{Kn}(1 - e^{-t/\mathrm{Kn}}) \cdot \mathcal{L}[M[f(t)]]\right) \quad \text{(CE-ramp)} \tag{34}$$

These are the target functions of the remainder networks g_θ, $\mathrm{Kn}^p g_\theta$ of SPINN-BGK, CE, CE-ramp. In our 1D3V problem setting($T_f = 0.1$, $k \sim 1$), in $\mathrm{Kn} \in [10^{-4},\ 10^{-2}]$, the time domain only includes the fast relaxation to the slow manifold, and not the slow convergence to global equilibrium of $O((\mathrm{Kn}k_{\min}^2)^{-1})$ timescale (where $k_{\min}$ is the smallest wavenumber scale of the problem) (Kogelbauer & Karlin, 2024; 2025).

Therefore, assuming equilibrium initial condition, the asymptotic behaviors of target functions at $t = 0$ and $t = T_f$ are

$$\begin{aligned} R^{(1)} &: \ 0 \ \to \ -\mathrm{Kn} \cdot \mathcal{L}[M[f(T_f)]] \\ R^{(2)} &: \ \mathrm{Kn} \cdot \mathcal{L}[M[f_0]] \ \to \ O(\mathrm{Kn}^2) \cdot f^{(2)}(T_f) \\ R^{(2)}_{\mathrm{ramp}} &: \ 0 \ \to \ O(\mathrm{Kn}^2) \cdot f^{(2)}(T_f). \end{aligned} \tag{35}$$

Figure 5 plots the L^2 norm of each target function in the 1D3V smooth problem, calculated with the $\mathrm{Kn} = 10^{-3}$ reference solution. We can see the asymptotic behaviors of the target functions of eq. (35); they all have an initial transient with the width $O(\mathrm{Kn})$, but their absolute magnitudes decrease from SPINN-BGK → CE → CE-ramp, easing the learning difficulty for the overall network.

For further investigation, we plotted the ratio $\frac{|R^{(2)}(t)|_{L^2}}{|R^{(2)}(0)|_{L^2}}$. Since $|R^{(2)}(t)|_{L^2}$ can be interpreted as a 'distance' between f and the NS manifold, Figure 6 clearly shows that distance between f and the NS manifold decays as $e^{-t/\mathrm{Kn}}$ at early times.

D.2 CE-RAMP EXPERIMENT

In this subsection, *CE SPINN* and *CE-ramp SPINN* denote SPINN-BGK with its micro–macro representation replaced by the CE and CE-ramp representations (eqs. (3) and (4)), respectively and all other settings follow SPINN-BGK.

Parameter tuning (Smooth problem). The parameter tuning was conducted on the 1D3V smooth problem, defined in Appendix A.2. We determined $p_{\mathrm{init}} = 1$ by coarse fixed grid search for $\mathrm{Kn} \times$

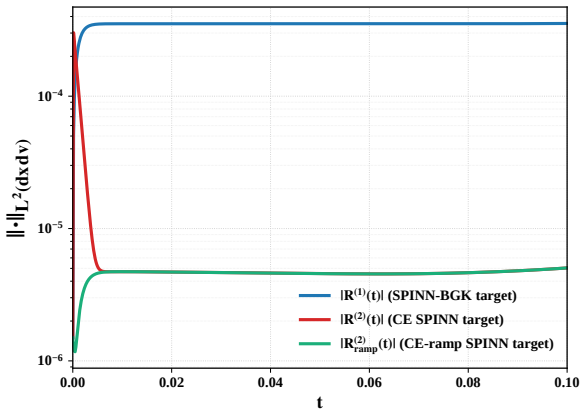


Figure 5: The L^2 norm of the target function for each representation in 1D smooth problem, $\mathrm{Kn} = 10^{-3}$

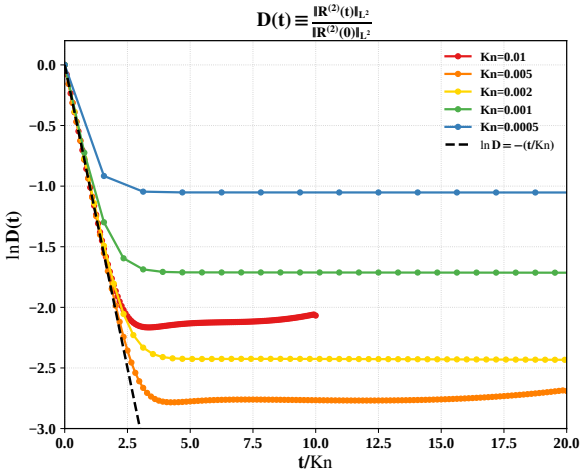


Figure 6: Temporal decay of $\|R^{(2)}(t)\|_{L^2}$ for a 1D3V Smooth problem, shown on a logarithmic scale relative to its initial value.

(λ, p), finding that $p_{\text{init}} = 1$ performed best for most of the combinations. For CE and CE-ramp test for smooth problems we used

$$\lambda_{\text{init}}(\mathrm{Kn}) = \frac{0.750}{1 + \exp\big(2.690 \cdot (\log_{10} \mathrm{Kn} + 1.841)\big)}$$

determined from coarse parameter sweep. We later examined the validity of this schedule with finer grid search. For CE SPINN, this $\lambda_{\text{init}}(\mathrm{Kn})$ was almost identical to the optimal $\lambda_{\text{init}}(\mathrm{Kn})$ curve. For CE-ramp SPINN, the ramp term reduced sensitivity to λ_{init} in large Kn region, and loss plateau of $\lambda_{\text{init}} = 0.7 \sim 0.9$ was observed in low Kn region; we therefore retained the original λ_{init} schedule in all cases.

One point to notice is that the theoretical value of $\lambda = 1.0$ was inadequate for the initial value. For both of CE SPINN and CE-ramp SPINN, $\lambda_{\text{init}} = 1.0$ resulted in 2–2.6 times larger macroscopic error than the optimal value. We interpret this as evidence that an overly strong bias $\lambda \sim 1$ hinders early training when the structured part is not accurate enough.

For the learned value of λ_{final} see the subsection D.3.

Parameter tuning (Riemann problem). For the Riemann problem, as fig. 7 shows, accuracy is relatively insensitive to λ_{init}, and we attribute this to the fact that shock is the main bottleneck of

learning. We chose $\lambda_{\text{init}} = 1.0$ for the 1D3V Riemann problem, $\lambda_{\text{init}} = 0.75$ for the 2D3V Riemann problem. For the higher-Kn Riemann problems, we used smaller values (Appendix G.1).

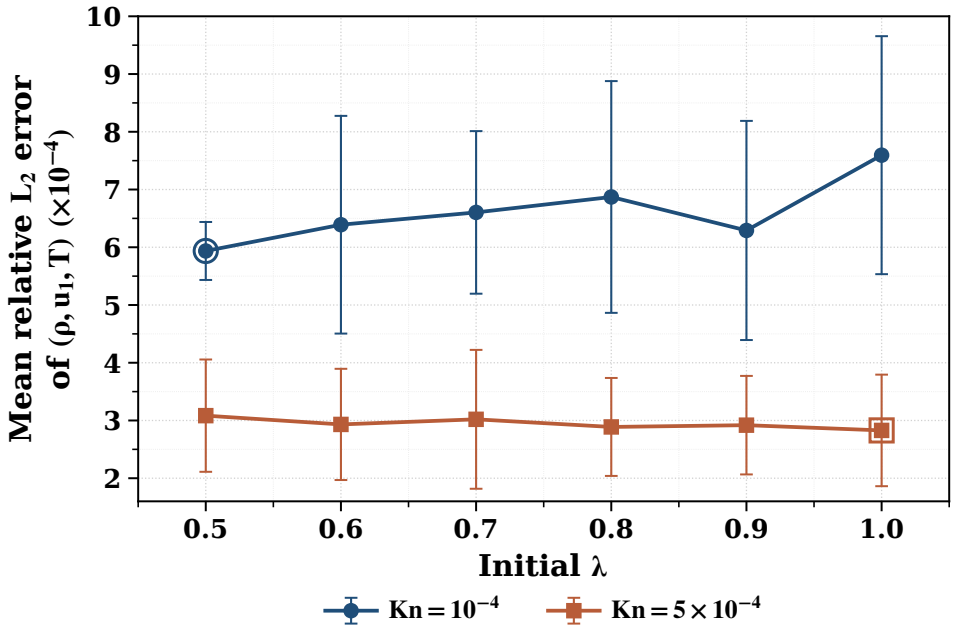


Figure 7: The effect of λ_{init} on macroscopic error for 1D3V Riemann problem at $\text{Kn} = 10^{-4}, 5 \times 10^{-4}$.

1D3V Smooth. We tested CE SPINN and CE-ramp SPINN with 1D3V smooth problem presented in Section 4.1 of Oh et al. (2025), comparing with original SPINN-BGK. As seen in figure 8, CE improves on SPINN-BGK in low Kn region, but exhibits an error bump near $\text{Kn} \sim 10^{-3}$ region. CE-ramp shows further improvement, particularly reducing the error bump near $\text{Kn} \sim 10^{-3}$.

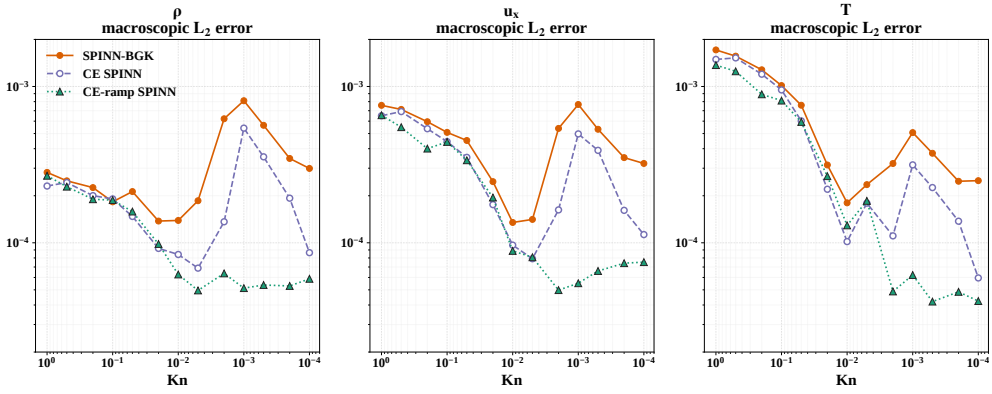


Figure 8: **1D3V Smooth**: SPINN-BGK vs. CE SPINN vs. CE-ramp SPINN, full Kn sweep.

D.3 Further Analysis of CE-Ramp

All further analyses use the results of the 1D3V smooth problem.

Theoretical values. Consider the CE-ramp ansatz for local equilibrium initial condition $f_\theta = M_\theta - (1 - e^{-t/\text{Kn}})\lambda\,\text{Kn}\,\mathcal{L}[M_\theta] + \text{Kn}^p g_\theta$. eq. (31) selects $\lambda = 1$ as its theoretical baseline. Also,

we expected given $\lambda \sim 1$, with sufficiently smooth problem(so that $\mathcal{L}_E \approx \mathcal{L}$) and well-trained representation (so that $M_\theta \approx M[f]$) the magnitude of remainder part is $|\mathrm{Kn}^p g_\theta| \sim O(\mathrm{Kn}^2)$, and with inaccurate λ value $|\mathrm{Kn}^p g_\theta| \approx \left| -(1-\lambda)(1-e^{-t/\mathrm{Kn}})\,\mathrm{Kn}\,\mathcal{L}M + O(\mathrm{Kn}^2) \right| \sim O(\mathrm{Kn})$.

Learned λ value study. Figure 9 left (a) shows $\lambda_{\text{final}} \to 1$ as $\mathrm{Kn} \to 0$ even though $\lambda_{\text{init}} \nrightarrow 1$. This provides supporting evidence for our CE and CE-ramp ansatze. Comparing the CE expansion with our ansatze, theoretical value of λ is exactly 1. However regarding the fact that CE expansion is intended for the low-Kn limit, it is natural that λ approaches 1 as $\mathrm{Kn} \ll 1$ and approaches 0 as Kn increases, thereby turning off the CE correction.

Magnitude of Remainder Network Part. Figure 9 (b) shows the relative magnitude of remainder network part $\mathrm{Kn}^p g_\theta$ in the learned solution. We observe the following in the low-Kn regime:

1. CE and CE-ramp SPINN show lower macroscopic error even with smaller remainder part.
2. The magnitude of remainder part of CE-ramp SPINN scales with Kn^2, in agreement with the CE expansion.
3. The magnitude of remainder part of CE SPINN shows similar behavior but has a relative bump of $\mathrm{Kn} \sim 10^{-3}$, which coincides with the position of the macroscopic error bump.

This provides strong evidence that expanding the structured part with the correct ansatz – thereby reducing the portion of f that the hard-to-learn microscopic network must represent – improves the PINN's learning, and that the magnitude of the learned remainder part is itself indicative of accuracy.

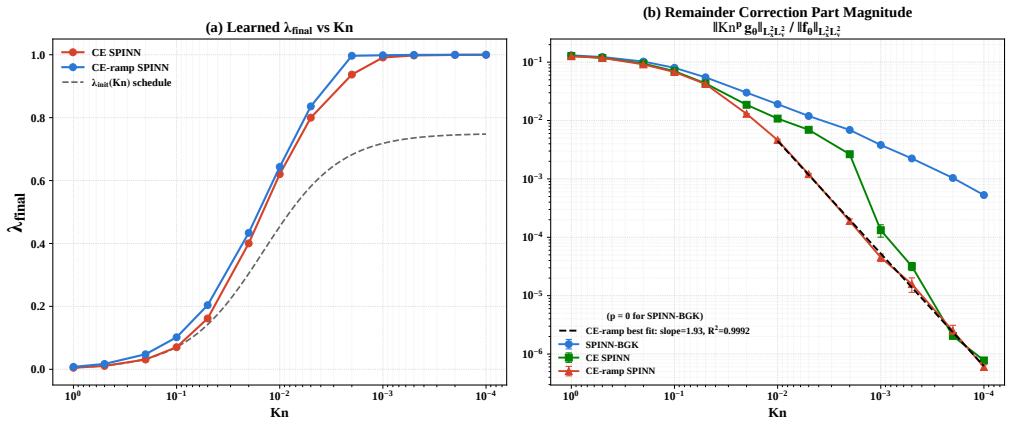


Figure 9: Learned λ_{final}, remainder part magnitude

D.4 Additional CE-Ramp Experiments

1D3V Bimodal (nonequilibrium IC). To check the performance of CE-ramp SPINN in the far-from-equilibrium initial condition, we tested the bimodal (two-stream) initial condition, where f_0 is the sum of two Maxwellians with distinct mean velocities, as motivated by Yan & Jin (2013). For the details see Appendix A. In this problem with low Kn, the solution has strong temporal stiffness due to rapid initial relaxation toward local equilibrium.

As seen in figure 10, SPINN-BGK and CE exhibit larger errors in the low Kn regime, whereas CE-ramp SPINN maintains $\sim 10^{-4}$ error level. This shows the initial-layer corrections of the CE-ramp ansatz work well in the far-from-equilibrium initial condition problems.

2D3V Smooth. This problem is based on '2D smooth problem' presented in Section 4.3 of Oh et al. (2025). As original problem had uniform initial velocity and temperature distribution, we changed the initial temperature from $T_0(x, y) = 1$ to $T_0(x, y) = 1 + 0.5\sin(2\pi x)$. As in 1D3V

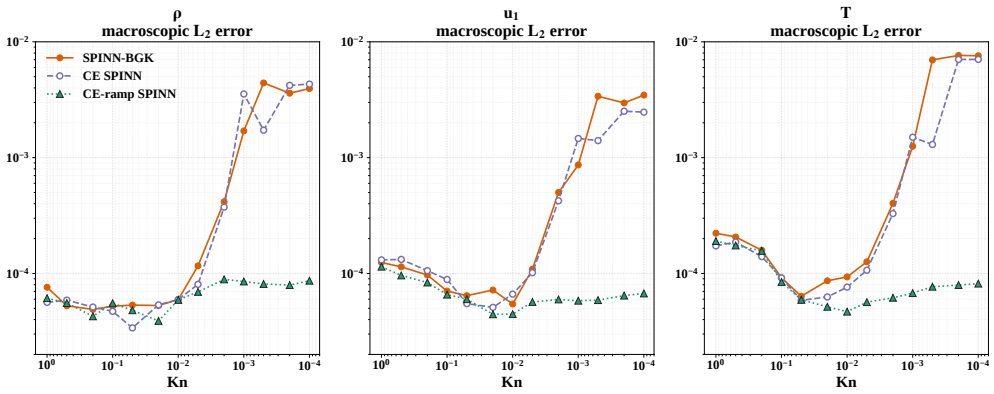


Figure 10: **1D3V Bimodal**: SPINN-BGK vs. CE SPINN vs. CE-ramp SPINN, full Kn sweep.

smooth problem, we could observe same pattern here: CE improvement over SPINN-BGK and error bump near $\mathrm{Kn} \sim 10^{-3}$, and resolution in the CE-ramp.

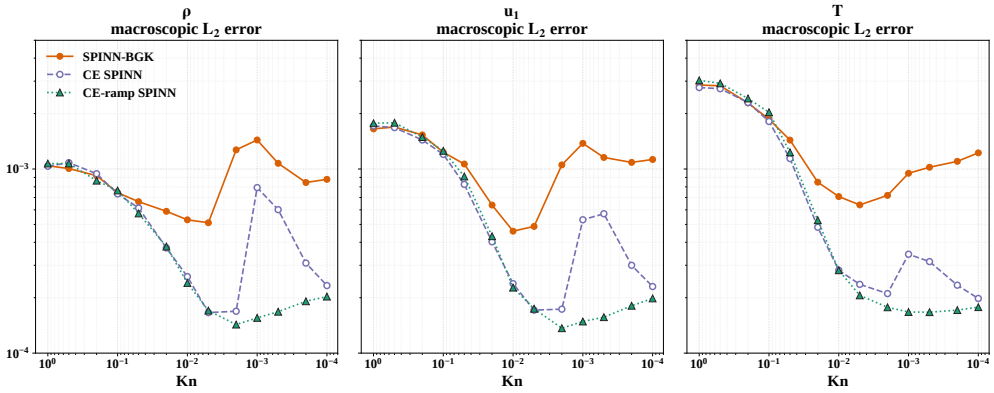


Figure 11: **2D3V smooth** (non-isothermal): SPINN-BGK vs. learnable CE vs. CE-ramp, full Kn sweep.

2D3V Isothermal. This is the original problem considered in Oh et al. (2025), where $\mathbf{u}_0 = \mathbf{0}$, $T_0(x, y) = 1$ and only initial density perturbation drives the dynamics. In this case we see the CE $\mathrm{Kn} \sim 10^{-3}$ error bump does not appear. This can be explained by Chapman-Enskog theory. When initial condition is in local equilibrium $f_0 = M[\boldsymbol{v}; \rho_0(\boldsymbol{x}), u_0(\boldsymbol{x}), T_0(\boldsymbol{x})]$, target function of remainder network eq. 33 has the form as follows:

$$R^{(2)}(0) = \mathrm{Kn} \cdot f_0 \left[\frac{1}{T_0} \left(c \otimes c - \frac{|c|^2}{d_v} I \right) : \nabla u_0 + \left(\frac{|c|^2}{2T_0} - \frac{d_v + 2}{2} \right) c \cdot \nabla \log T_0 \right], \quad c = \boldsymbol{v} - u_0. \tag{36}$$

Therefore in this problem $\nabla u_0 = \nabla T_0 = 0$ makes $R^{(2)}(0)$ vanish like $R^{(2)}_{\mathrm{ramp}}(0)$ (i.e. the initial condition lies on the intersection of equilibrium manifold and NS manifold) and the learning difficulty of CE and CE-ramp becomes comparable. This specific problem setting supports the interpretation that the improvement of CE-ramp over CE stems from resolving the initial bias mismatch at $t = 0$.

E Two-Stage Method Details

In this section, we describe the construction of the shock level set, used in the two-dimensional Riemann problem. Following Section 4.2, we first train a solution network without lifting, extract

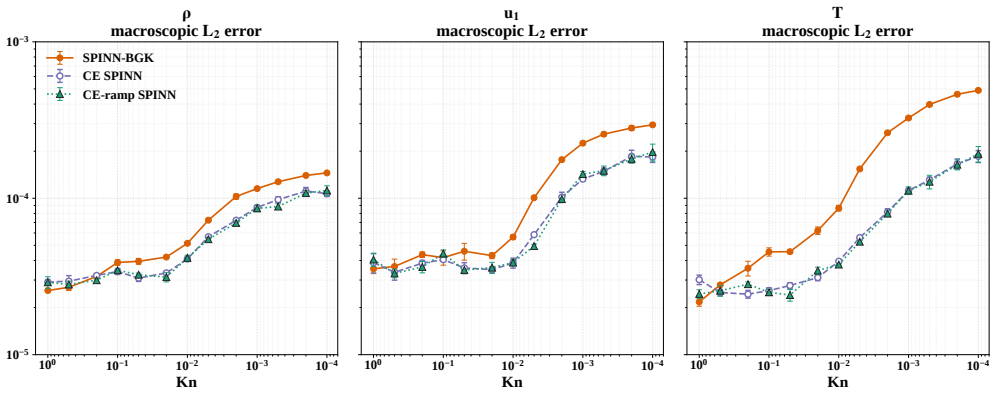


Figure 12: **2D3V Isothermal**: SPINN-BGK vs. learnable CE vs. CE-ramp, full Kn sweep.

shock-aware point cloud from its macroscopic fields, and fit a smooth level-set function using Implicit Geometric Regularization (IGR) (Gropp et al., 2020). The fitted geometry ψ_{shock} is then fixed during stage-2 training.

E.1 Stage-1 prediction

Since the only goal of the stage-1 model is to locate the shock, we use micro-macro representation for stage-1 model and train for 50k iterations. The network architecture and optimizer follow the default settings in Tables 5 and 6. Specific settings for two-stage algorithm are in Table 7.

E.2 Preprocessing for IGR

Fujimoto et al. (2019) extract shock using Canny edge detection algorithm (Canny, 1986) with pressure p. Inspired by this method, we introduce score S detecting compression. We first construct the macroscopic moments $(\rho, \boldsymbol{u}, T)$ on uniform grid 257×257, with time grid $t_k = kT_f/10$, $k = 0, \ldots, 10$. Here T_f is the final time in Table 7. Let $p = \rho T$ and $\Delta x = 2L_x/256$ denote pressure and grid spacing, respectively. So our data begins with $\{(p_{ij}^{(k)}, \boldsymbol{u}_{ij}^{(k)})\}_{i,j=0}^{256}$ for each time t_k.

Low pass filtering. The stage-1 predictions may contain noise and oscillations. We therefore apply a spatial Gaussian filter, yielding $(\bar{p}, \bar{\boldsymbol{u}})$.

Score and Normal vectors. We combine the pressure gradient $A = \nabla_{\boldsymbol{x}} \bar{p}$ with the compression indicator $C = -\nabla_{\boldsymbol{x}} \cdot \bar{\boldsymbol{u}}$ to define score

$$S = |A| \max(C, 0), \quad \boldsymbol{n} = -\frac{A}{|A| + \epsilon} \tag{37}$$

Since pressure is continuous across a contact wave, $|A|$ reduces sensitivity to contact-like transitions, while $\nabla_{\boldsymbol{x}} \cdot \bar{\boldsymbol{u}} < 0$ condition – which is motivated by the physics dependent weight of Liu et al. (2024) – rewards compressive regions. At $t = 0$, we instead use $S = |A|$, since the zero initial velocity implies $C = 0$. Based on this score, we extract $\mathcal{G}_k$, the candidate grid points lying on shock surfaces.

Non-maximum suppression and Correction. We apply non-maximum suppression with S to remove non-maximum along the local normal. For each grid point $\boldsymbol{x}$, write $S_0 = S(\boldsymbol{x})$ and $S_\pm = S(\boldsymbol{x} \pm \Delta x\, \boldsymbol{n}(\boldsymbol{x}))$, evaluating off-grid scores by bilinear interpolation. We retain points satisfying $S_0 \geq S_-$ and $S_0 \geq S_+$. Because peak locations generally lie between grid points, we refine the retained locations by a quadratic fit along the normal:

$$\boldsymbol{x}^* = \boldsymbol{x} + \delta \boldsymbol{n}(\boldsymbol{x}), \qquad \delta = \Delta x \frac{S_- - S_+}{2(S_- - 2S_0 + S_+)} \tag{38}$$

Table 11: IGR network and training settings for the two-dimensional Riemann problem.

Setting	Value
Network	
Architecture	$3 \to 64 \to 64 \to 64 \to 1$
Hidden / output activation	$\tanh$ / linear
Input	$(t/T_f, x_1/L_x, x_2/L_x)$
Weight initialization	$W_{ij} \sim \mathcal{N}(0, 2/(d_{\text{in}} + d_{\text{out}}))$ per layer
Bias initialization	Zero
Training	
Loss weights	$(\lambda_{\text{surf}}, \lambda_{\text{normal}}, \lambda_{\text{eik}})$=$(1, 1, 0.01)$
Surface batch	512 points, sampled with replacement
Nearby / uniform Eikonal samples	512 / 512 per update
Normal-offset distribution	$\xi \sim \mathcal{N}(0, 0.1^2)$
Optimizer	Adam; $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon_{\text{Adam}} = 10^{-8}$
Learning-rate schedule	Cosine decay from 10^{-3} to a scheduled endpoint of 10^{-5}; no warmup
Updates / retained model	20,000 / final iterate

where small value is added to the denominator for numerical stability. So the selected points are:

$$\mathcal{D} = \{(t_k, \boldsymbol{x}^*_{ki}, \boldsymbol{n}_{ki}) : k = 0, \dots, 10,\ i = 1, \dots, N_k\}\,, \tag{39}$$

E.3 Implicit Geometric Representation

Motivated by IGR (Gropp et al., 2020), we train a neural network $\psi_\omega(t, \boldsymbol{x})$ using eq. (39), which approximates the shock level-set function ψ_{shock} after training ends. The details of model and training settings are in Table 11.

Training objective. For a surface minibatch $I = \{(t_i, \boldsymbol{x}_i, \boldsymbol{n}_i)\}_{i \in I}$, we minimize

$$\begin{aligned} L_{\text{surf}} &= \frac{1}{|I|} \sum_{i \in I} |\psi_\omega(t_i, \boldsymbol{x}_i)|, \\ L_{\text{normal}} &= \frac{1}{|I|} \sum_{i \in I} \|\nabla_{\boldsymbol{x}} \psi_\omega(t_i, \boldsymbol{x}_i) - \boldsymbol{n}_i\|_2\,, \\ L_{\text{eik}} &= \mathbb{E}_{\boldsymbol{z}} \left(\|\nabla_{\boldsymbol{x}} \psi_\omega(\boldsymbol{z})\|_2 - 1\right)^2, \\ L_{\text{IGR}} &= \lambda_{\text{surf}} L_{\text{surf}} + \lambda_{\text{normal}} L_{\text{normal}} + \lambda_{\text{eik}} L_{\text{eik}}. \end{aligned} \tag{40}$$

Here $||\cdot||_2$ is the standard Euclidean norm. The surface term encourages ψ_ω to vanish on the extracted points, the normal term fixes its local direction and scale to normal data, and the Eikonal term encourages unit spatial gradient. These terms encourage an approximate spatial signed-distance function.

Sampling and Optimization. For each sampled point $i \in I$, we construct a nearby point $\boldsymbol{z} = (t_i, \boldsymbol{x}_i + \xi_i \boldsymbol{n}_i)$ with $\xi_i \sim \mathcal{N}(0, 0.1^2)$ and uniform points on the training domain. Table 11 summarizes the model and training settings.

E.4 Stage-2 training

Finally, we get ψ_{shock} and introduce auxiliary network c_{η_j} in eq. (6) whose detailed settings are in Table 12. We train Stage 2 model specified in Table 5

Table 12: Auxiliary network settings for the two-dimensional Riemann problem. The three functions c_{η_j} share one hidden layer.

Setting	Value
Architecture	$3 \to 16 \to 3$; outputs $(c_{\eta_1}, c_{\eta_2}, c_{\eta_3})$
Hidden / output activation	$\tanh$ / linear
Input	$(t/T_f, x_1/L_x, x_2/L_x)$, $L_x = 1$
Hidden-layer initialization	$W_{ij} \sim \mathcal{N}(0, 1/3)$; zero bias
Output-layer initialization	$W_{ij} \sim \mathcal{N}(0, 10^{-4})$; $b_j \sim \mathcal{U}[-2.5, 2.5)$

F Loss Construction

F.1 Conservation Loss Construction

Define the collision invariants $\psi(\boldsymbol{v}) = (1, \boldsymbol{v}^\top, \frac{1}{2}|\boldsymbol{v}|^2)^\top$ and their associated moments and fluxes by

$$U = \int_{\mathbb{R}^3} \psi f d\boldsymbol{v} = (\rho, (\rho \boldsymbol{u})^\top, E)^\top, \qquad F = \int_{\mathbb{R}^3} \psi \otimes \boldsymbol{v} f d\boldsymbol{v}, \tag{41}$$

where $E = \frac{1}{2}\rho|\boldsymbol{u}|^2 + \frac{3}{2}\rho T$ is the total energy density. The conservation residual is

$$R_{\text{Cons}} = \partial_t U + \nabla \cdot F = \left(R_\rho, R_m^\top, R_E\right)^\top. \tag{42}$$

Proposition 1. $R_{\text{BGK}} = 0$ *implies* $R_{\text{Cons}} = 0$

Proof. Since $M[f] - f$ is collision-invariant free,

$$\int_{\mathbb{R}^3} \psi(\boldsymbol{v})\big(M[f] - f\big)\, d\boldsymbol{v} = 0. \tag{43}$$

Multiplying the BGK residual by ψ and integrating over velocity gives

$$\begin{aligned} \int_{\mathbb{R}^3} \psi R_{\text{BGK}}\, d\boldsymbol{v} &= \int_{\mathbb{R}^3} \psi \left(\partial_t f + \boldsymbol{v} \cdot \nabla f - \frac{1}{\text{Kn}}(M[f] - f)\right) d\boldsymbol{v} \\ &= \partial_t \int_{\mathbb{R}^3} \psi f\, d\boldsymbol{v} + \nabla \cdot \int_{\mathbb{R}^3} \psi \otimes \boldsymbol{v}\, f\, d\boldsymbol{v} \\ &= \partial_t U + \nabla \cdot F = R_{\text{Cons}}. \end{aligned}$$

Thus, vanishing BGK residual implies vanishing conservation residual. □

Since $R_{\text{BGK}} \neq 0$ in general PINN training, explicitly penalizing R_{Cons} emphasizes macroscopic conservation.

F.2 Entropy Transport Loss Construction

For a monatomic ideal gas in our non-dimensional variables, the specific thermodynamic entropy is

$$s = \frac{3}{2}\log T - \log \rho + s_0, \tag{44}$$

where s_0 is a constant (Toro, 2013). Let $\mathbf{c} = \boldsymbol{v} - \boldsymbol{u}$ and define

$$P = \int_{\mathbb{R}^3} \mathbf{c} \otimes \mathbf{c}\, f\, d\boldsymbol{v}, \qquad p = \rho T, \qquad \Sigma = P - pI, \qquad Q = \frac{1}{2}\int_{\mathbb{R}^3} |\mathbf{c}|^2 \mathbf{c} f\, d\boldsymbol{v}. \tag{45}$$

Thus, Q is the heat flux and Σ is the nonequilibrium part of the pressure tensor, fixing the stress sign convention used below.

Proposition 2. *Define material time derivative* $D_t \equiv \partial_t + \boldsymbol{u} \cdot \nabla$. *The entropy-transport residual satisfies the following identity:*

$$R_{\text{Ent}} = \rho T D_t s + \nabla \cdot Q + \Sigma : \nabla \boldsymbol{u}. \tag{46}$$

Moreover, $R_{\text{Cons}} = 0$ *implies* $R_{\text{Ent}} = 0$.

Proof. Set $e = \frac{3}{2}T$, so that $E = \rho e + \frac{1}{2}\rho|\boldsymbol{u}|^2$. The conservation residuals can be equivalently written as (Jin et al., 2024)

$$\begin{aligned} R_\rho &= \partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}), \\ R_m &= \partial_t (\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u} + P), \\ R_E &= \partial_t E + \nabla \cdot (E\boldsymbol{u} + P\boldsymbol{u} + Q). \end{aligned} \tag{47}$$

Following the formulation in Struchtrup (2020) , we rewrite the conservation laws using the material derivative $D_t = \partial_t + \boldsymbol{u} \cdot \nabla$, to obtain:

$$D_t \rho + \rho \nabla \cdot \boldsymbol{u} = R_\rho \tag{48}$$

$$\rho D_t \boldsymbol{u} + \nabla \cdot P = R_m - \boldsymbol{u} R_\rho \tag{49}$$

$$\rho D_t e + \nabla \cdot Q + P : \nabla \boldsymbol{u} = R_E - \boldsymbol{u} \cdot R_m + \left(\frac{1}{2}|\boldsymbol{u}|^2 - e \right) R_\rho \tag{50}$$

The first two identities follow from the product rule and third one with the symmetry of P.

Differentiating $s = \frac{3}{2}\log T - \log \rho + s_0$ and using eq. (48) gives

$$\begin{aligned} \rho D_t e &= \rho T D_t s + T D_t \rho \\ &= \rho T D_t s + T(R_\rho - \rho \nabla \cdot \boldsymbol{u}) \end{aligned} \tag{51}$$

Substituting this expression into eq. (50) gives,

$$R_E - \boldsymbol{u} \cdot R_m + \left(\frac{1}{2}|\boldsymbol{u}|^2 - \frac{5}{2}T \right) R_\rho = \rho T D_t s + \nabla \cdot Q + \Sigma : \nabla \boldsymbol{u} \tag{52}$$

□

Thus, R_{Ent} explicitly penalizes thermodynamic entropy-transport balance.

F.3 Residual Normalization

Following SPINN-BGK, we normalize each residual by a denominator D, detached using `stop_gradient`, before applying the loss penalty. For the BGK residual, we use $D = |f| + \epsilon$ to emphasize errors in the velocity tails. For the conservation residual, we use $D = 1 + |m|$ componentwise, where $m = (\rho, (\rho \boldsymbol{u})^\top, 2E)^\top$, to balance moment scales while avoiding division by near-zero values. For the entropy-transport residual, we use $D = \max(\rho, 10^{-3}) \max(T, 10^{-3})$, reusing the convention implemented for the conservation residual. These settings are summarized in Table 10.

G Additional Results

G.1 Additional Problems

Smooth problems. We further evaluate KF-SPINN on the smooth problems. As shown in Table 13, we observe that KF-SPINN achieves the lowest error in most cases, whereas CE-ramp SPINN – lacking the hard-IC and conservation-loss contributions – acts as a fast surrogate.

More 1D3V Riemann problems. Although our main focus is low Kn regime, we also evaluate KF-SPINN on the Riemann problems at larger Kn to demonstrate Kn across robustness. We use $\lambda_{\mathrm{init}} = 0.1$ for $\mathrm{Kn} = 1, 0.1$ and $\lambda_{\mathrm{init}} = 0.5$ for $\mathrm{Kn} = 0.01$. Since the contact wave is not so strong, we turn off R_{Ent}. Table 14 shows that KF-SPINN is also better than SPINN-BGK and Ko et al. (2026) and comparable to NSR, with fast training (See Table 1) in the range of high Kn.

G.2 Additional Experiments

Motivation for two-stage method. To test whether the remaining features (except shock) require exact wave speeds, we retain c_{shock} and sample additional speeds uniformly from $(-2.5, 2.5)$. Table 15 shows that three random auxiliary speeds reduce the average error from 5.57×10^{-3} to

Table 13: Mean macroscopic relative L^2 error for the Smooth problems.

Problem	Kn	SPINN-BGK (Oh et al., 2025)	Ko et al. (2026)	NSR (Li et al., 2024)	CE-ramp SPINN	**KF-SPINN**
1D3V (Smooth)	1e-02	1.95e-04	**3.95e-05**	1.55e-04	9.05e-05	8.13e-05
	5e-03	2.55e-04	4.29e-05	4.57e-03	5.52e-05	**3.07e-05**
	1e-03	7.98e-04	2.98e-04	9.42e-04	5.09e-05	**3.02e-05**
	5e-04	5.01e-04	1.77e-04	5.66e-04	6.14e-05	**2.45e-05**
	1e-04	3.07e-04	8.99e-05	3.52e-04	6.44e-05	**3.20e-05**
1D3V (Non-eq. IC)	1e-02	6.91e-05	4.90e-05	1.03e-04	5.02e-05	**3.73e-05**
	5e-03	1.17e-04	6.84e-05	1.89e-04	6.09e-05	**3.07e-05**
	1e-03	1.27e-03	1.04e-03	4.84e-03	7.04e-05	**3.38e-05**
	5e-04	4.93e-03	4.71e-03	4.92e-03	7.23e-05	**3.42e-05**
	1e-04	5.00e-03	5.02e-03	4.96e-03	7.85e-05	**3.88e-05**
2D3V (Smooth)	1e-02	1.01e-03	**2.77e-04**	9.97e-04	4.20e-04	3.39e-04
	5e-03	1.01e-03	5.11e-04	4.17e-03	3.18e-04	**2.05e-04**
	1e-03	2.56e-03	1.20e-03	2.13e-03	4.08e-04	**1.48e-04**
	5e-04	2.17e-03	1.22e-03	2.26e-03	5.66e-04	**1.62e-04**
	1e-04	2.14e-03	1.42e-03	3.79e-03	9.27e-04	**1.72e-04**
2D3V (Isothermal)	1e-02	1.45e-04	9.79e-05	3.37e-04	8.88e-05	**5.96e-05**
	5e-03	2.52e-04	2.13e-04	4.37e-04	1.18e-04	**6.66e-05**
	1e-03	5.37e-04	5.14e-04	8.23e-04	3.31e-04	**1.01e-04**
	5e-04	6.20e-04	5.63e-04	9.45e-04	3.61e-04	**1.15e-04**
	1e-04	7.21e-04	6.61e-04	1.12e-03	4.44e-04	**1.41e-04**

Table 14: Mean macroscopic relative L^2 error for the Riemann problems.

Problem	Kn	SPINN-BGK (Oh et al., 2025)	Ko et al. (2026)	NSR (Li et al., 2024)	**KF-SPINN**
1D3V (Riemann)	1.00e+00	2.89e-03	1.24e-03	7.35e+01	**1.05e-03**
	1.00e-01	2.10e-03	7.93e-04	**4.64e-04**	5.30e-04
	1.00e-02	1.70e-03	1.29e-03	**5.65e-05**	1.19e-04

6.95×10^{-4}. Four or five random speeds yield similar, but slightly higher, average errors. Accurate non-shock wave speeds are therefore unnecessary for obtaining a substantial improvement in this test, although additional lifting directions remain useful, providing combination choices to neural networks, which motivates learnable network c_{η_j} of eq. (6).

Multiscale features. We vary the active scales within $10, 25, 50, 100, 200$, progressively adding scales from either the smallest or the largest κ (Table 16). Using all five scales yields the lowest errors across all macroscopic variables, reducing the average relative L^2 error from 1.97×10^{-2} without scaling to 7.74×10^{-4}. These results support combining multiple transition widths rather than relying on a single feature.

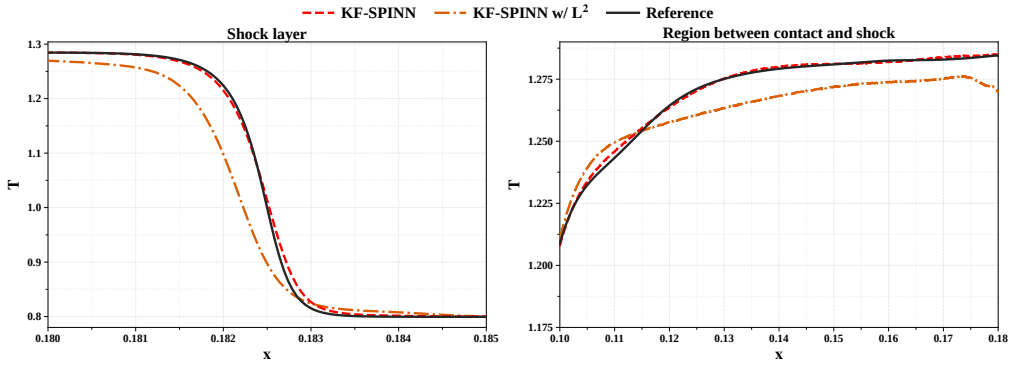
Temporal modulation. Table 17 isolates the effect of the polynomial time features. Linear and quadratic modulation each improve the average error over the configuration without temporal modulation. Combining both gives the lowest mean error for every macroscopic field. This supports allowing the lifting features to vary in amplitude over time.

CE-ramp for Riemann problem. Although resolving sharp spatial shock layers is the primary challenge in 1D3V Riemann problem, the CE-ramp ansatz provides an additional reduction in macroscopic errors. Table 18 compares configurations with and without CE-ramp. CE-ramp generally improves accuracy across the tested Knudsen numbers. Thus, its treatment of temporal relaxation remains beneficial even in the presence of shocks.

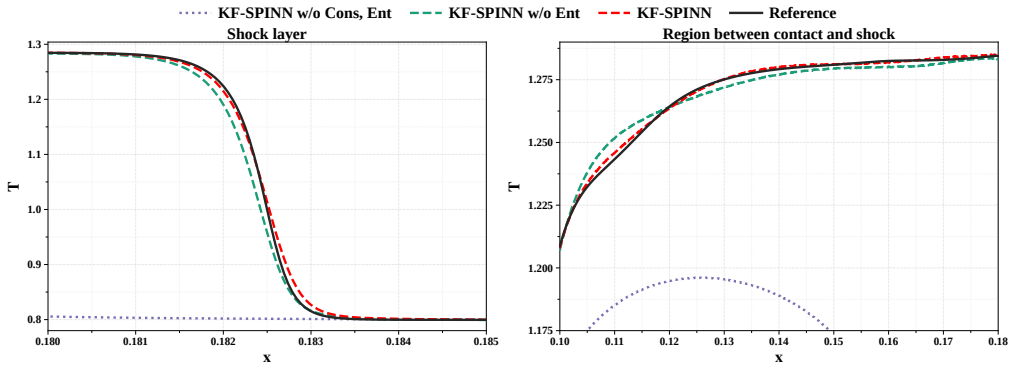
H Reference Solver

H.1 Numerical Method

We compute numerical reference solutions using a conservative semi-Lagrangian method (Cho et al., 2021). Transport is evaluated along characteristics using conservative CWENO reconstruction, while the BGK relaxation is treated implicitly. We use DIRK2 for the one-dimensional and smooth benchmarks, and constant-step BDF3 for the two-dimensional Riemann benchmark, initialized by two DIRK3 steps. An L^2 correction is used for discrete Maxwellian correction. References and



(a) L^2 Loss vs. Charbonnier Loss



(b) Structure-aware residuals

Figure 13: **Ablation of the training objective on the 1D Riemann problem.** Temperature profiles are shown across the shock layer (left) and between the contact and shock (right). **(a)** L^2 versus Charbonnier penalties, with both conservation and entropy-transport residuals included. **(b)** BGK residual alone, with the conservation residual, and with both additional residuals, all using the Charbonnier penalty. Black curves denote the numerical reference. The conservation residual recovers the shock transition, while the entropy-transport residual further improves the temperature behind the shock.

Table 15: Relative L^2 errors for different numbers of random auxiliary speeds, with c_{shock} retained in all cases, 1D Riemann Problem with $\text{Kn} = 10^{-4}$ at $t = 0.1$. Best and second-best means are shown in **bold** and underline, respectively.

# of Random Speeds	rel. L^2 error (mean)	rel. L^2 error (ρ)	rel. L^2 error (u_1)	rel. L^2 error (T)
0	5.57e-03	2.92e-03	6.82e-03	6.97e-03
1	1.82e-03	1.24e-03	1.85e-03	2.38e-03
2	1.07e-03	6.14e-04	1.07e-03	1.51e-03
3	**6.95e-04**	3.69e-04	**7.78e-04**	**9.38e-04**
4	<u>7.29e-04</u>	<u>3.27e-04</u>	<u>8.51e-04</u>	<u>1.01e-03</u>
5	7.73e-04	**2.82e-04**	9.40e-04	1.10e-03

Table 16: Relative L^2 errors on multiscale feature $\kappa \in \{10, 25, 50, 100, 200\}$, 1D Riemann Problem with $\text{Kn} = 10^{-4}$ at $t = 0.1$. Best means are shown in **bold**.

10	25	50	100	200	rel. L^2 error (mean)	rel. L^2 error (ρ)	rel. L^2 error (u_1)	rel. L^2 error (T)
					1.97e-02	9.49e-03	2.41e-02	2.54e-02
✓					7.19e-03	2.28e-03	9.40e-03	9.88e-03
✓	✓				4.47e-03	1.44e-03	5.78e-03	6.19e-03
✓	✓	✓			3.34e-03	1.05e-03	4.38e-03	4.58e-03
✓	✓	✓	✓		1.56e-03	5.68e-04	2.02e-03	2.07e-03
				✓	3.18e-02	1.57e-02	4.01e-02	3.96e-02
			✓	✓	6.36e-03	4.38e-03	4.54e-03	1.02e-02
		✓	✓	✓	7.25e-03	4.06e-03	7.00e-03	1.07e-02
	✓	✓	✓	✓	4.57e-03	2.54e-03	4.54e-03	6.63e-03
✓	✓	✓	✓	✓	**7.74e-04**	**3.68e-04**	**9.02e-04**	**1.05e-03**

more details are therein. Our implementation uses JAX with double precision. See Table 19 for refinement test results and settings.

H.2 Refinement Error

For a macroscopic field q, we use the refinement error

$$E_q = \frac{\left(\sum_i w_i |q_i - q_i^{\text{ref}}|^2\right)^{1/2}}{\left(\sum_i w_i |q_i^{\text{ref}}|^2\right)^{1/2} + \delta_q}, \qquad w_i = (\Delta x)^{d_x}, \tag{53}$$

where $\delta_q = 1$ for velocity components and $\delta_q = 0$ for density and temperature. Refinement error is measured with a spatially $2\times$ coarser grid for each axis.

We note that computational constraints prevented us from verifying that the 2D3V reference error is substantially smaller than the reported model errors. But, the refinement from a $Nx = 330^2$ grid to a $Nx = 660^2$ reduces the temperature overshoot and smearing near shock (Figure 14).

Table 17: Relative L^2 errors on time modulation, 1D Riemann Problem with $\mathrm{Kn} = 10^{-4}$ at $t = 0.1$. Best and second-best means are shown in **bold** and underline, respectively.

$T_1(t)$	$T_2(t)$	rel. L^2 error (mean)	rel. L^2 error (ρ)	rel. L^2 error (u_1)	rel. L^2 error (T)
		2.59e-03	1.35e-03	3.25e-03	3.18e-03
✓		1.32e-03	6.70e-04	1.66e-03	1.64e-03
	✓	<u>1.11e-03</u>	<u>5.19e-04</u>	<u>1.27e-03</u>	<u>1.54e-03</u>
✓	✓	**6.75e-04**	**3.63e-04**	**7.49e-04**	**9.11e-04**

Table 18: Effect of CE-ramp on the mean macroscopic relative L^2 error for the 1D3V Riemann problem. Both variants include conservation and entropy penalties.

Method	Kn=1e-03	7e-04	5e-04	3e-04	1e-04
KF-SPINN w/ CE-ramp	**1.03e-04**	**1.20e-04**	**1.68e-04**	**2.22e-04**	**6.35e-04**
KF-SPINN w/o CE-ramp	1.64e-04	1.79e-04	2.31e-04	3.23e-04	1.19e-03

Table 19: Reference discretizations and mean macroscopic relative L^2 refinement errors at $t = 0.1$ (Smooth, 1D3V Riemann) and $t = 0.07$ (2D3V Riemann).

Problem	Kn	Method	Grid (N_x^d, $N_{v_1}N_{v_2}N_{v_3}$)	Computational domain	rel. L^2 error (mean)
1D3V Riemann	1e-04	DIRK2	$(40960,\ 97 \times 25^2)$	$(-0.5, 0.5) \times (-10, 10)^3$	3.30×10^{-6}
	3e-04	DIRK2	$(20480,\ 97 \times 25^2)$	$(-0.5, 0.5) \times (-10, 10)^3$	1.95×10^{-6}
	5e-04	DIRK2	$(20480,\ 97 \times 25^2)$	$(-0.5, 0.5) \times (-10, 10)^3$	7.83×10^{-6}
	7e-04	DIRK2	$(20480,\ 97 \times 25^2)$	$(-0.5, 0.5) \times (-10, 10)^3$	4.68×10^{-6}
	1e-03	DIRK2	$(20480,\ 97 \times 25^2)$	$(-0.5, 0.5) \times (-10, 10)^3$	2.79×10^{-7}
2D3V Riemann	1e-04	BDF3	$(660^2,\ 21^3)$	$(-1, 1)^2 \times (-6, 6)^3$	9.19×10^{-3}
	3e-04	BDF3	$(660^2,\ 21^3)$	$(-1, 1)^2 \times (-6, 6)^3$	8.64×10^{-3}
	5e-04	BDF3	$(660^2,\ 21^3)$	$(-1, 1)^2 \times (-6, 6)^3$	8.15×10^{-3}
	7e-04	BDF3	$(660^2,\ 21^3)$	$(-1, 1)^2 \times (-6, 6)^3$	7.66×10^{-3}
	1e-03	BDF3	$(660^2,\ 21^3)$	$(-1, 1)^2 \times (-6, 6)^3$	6.94×10^{-3}
1D3V Smooth	1e-04	DIRK2	$(5120,\ 37^3)$	$(-0.5, 0.5) \times (-10, 10)^3$	1.076×10^{-7}
	3e-04	DIRK2	$(5120,\ 37^3)$	$(-0.5, 0.5) \times (-10, 10)^3$	1.077×10^{-7}
	5e-04	DIRK2	$(5120,\ 37^3)$	$(-0.5, 0.5) \times (-10, 10)^3$	1.078×10^{-7}
	7e-04	DIRK2	$(5120,\ 37^3)$	$(-0.5, 0.5) \times (-10, 10)^3$	1.079×10^{-7}
	1e-03	DIRK2	$(5120,\ 37^3)$	$(-0.5, 0.5) \times (-10, 10)^3$	1.080×10^{-7}

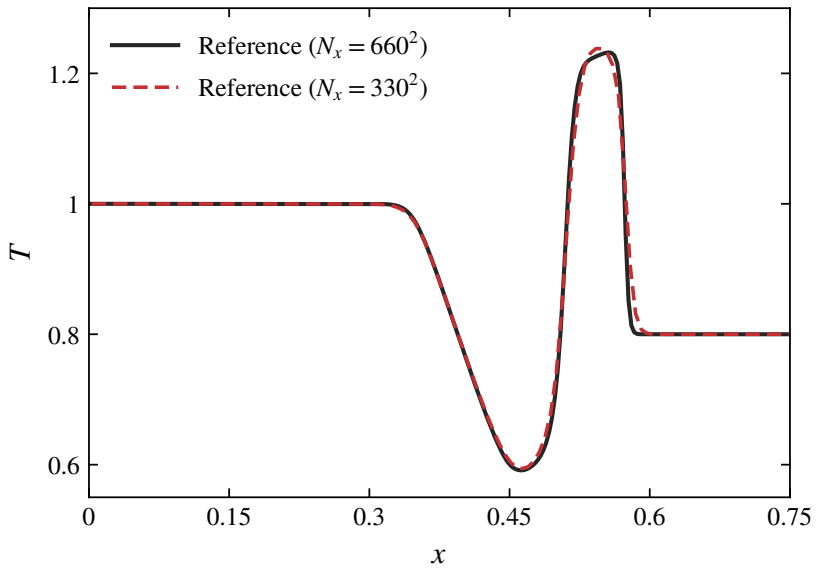


Figure 14: **Refinement of the 2D3V reference.** The temperature profile of 2D3V along $y = 0$ at $t = 0.07$ and $\mathrm{Kn} = 10^{-4}$. The finer grid reduces overshoot and shock smearing.