

Dimensionally consistent loss construction for physics-informed flow and thermomechanical models

Luis Torres

Abstract

The loss function of a physics-informed neural network combines quantities that can have different units, scales, sampling measures, and sensitivities to the network parameters. Consequently, balancing numerical loss values is not equivalent to balancing physical accuracy. This review develops a dimensionally explicit treatment of loss construction for incompressible flow, heat conduction, and elastic-viscoplastic solids. It separates nondimensionalization, residual normalization, statistical weighting, adaptive optimization, and sampling. Two analytical examples show how a change of units can change an uncorrected optimum and how equal loss values can produce unequal parameter gradients. Published flow and plasticity studies are examined alongside gradient-based, relative-progress, and pointwise adaptation methods. The synthesis identifies what each method controls, which comparisons are meaningful, and what evidence is needed to establish an improvement in force, temperature gradient, or stress. A reproducible loss specification should connect every residual and multiplier to a governing equation, observation model, or documented numerical objective.

1. A loss is a mathematical model of the training task

In a standard PINN, a neural field is trained against observations and residuals of a differential problem [1]. The broader physics-informed-learning literature includes many ways to incorporate physical structure [2], but a common feature is the combination of heterogeneous objectives. A reported scalar loss can conceal which condition dominates, which field is accurate, and whether the fitted problem remains the one intended by the engineer. This matters when the result is used to calculate a load, a heat flux, or a deformation limit.

Consider a schematic objective with data, momentum, continuity, boundary, and initial terms. Before specifying its weights, one must define each residual, its units, its integration measure, and the reduction used over samples. A mean-square momentum residual has different units from a mean-square velocity mismatch. Taking their numerical sum without a scale convention makes the objective dependent on the chosen system of units. The optimizer will still produce updates, but their physical meaning is not invariant.

This narrative review selects literature for its relevance to loss construction and engineering interpretation. The comparison emphasizes the physical meaning and optimization role of adaptive methods. Three application anchors are used: conditional flow prediction [3], mixed-output plasticity [4], and thermal modeling of manufacturing [5]. They demonstrate why an apparently similar PINN

objective can encode different physical information. Architectural choices and published training outcomes are retained within their source-specific scope.

The analysis distinguishes five operations. Nondimensionalization changes variables and equations consistently. Residual normalization chooses a reference magnitude for a stated equation. Statistical weighting represents observation uncertainty under an assumed noise model. Adaptive loss balancing changes optimization priorities. Sampling determines where and how an integral is estimated. These operations may be combined, but none can silently substitute for the others. Table 1 gives the corresponding audit questions.

Operation	What is changed	Question that must be answered
Nondimensionalization	Variables and differential operators	Were all chain-rule factors retained?
Residual normalization	Magnitude of an equation mismatch	What physical or numerical reference defines the scale?
Statistical weighting	Relative influence of observations	What covariance or noise model is assumed?
Adaptive balancing	Training priorities over iterations	What update rule and objective are actually optimized?
Sampling	Locations and integration weights	Is the target norm preserved or intentionally changed?

2. Deriving scales from incompressible flow

Let a constant-density Newtonian fluid have reference length L , speed U , and convective time L/U . With pressure scale ρU^2 , the dimensionless momentum equation contains $1/\text{Re}$, where $\text{Re} = UL/\nu$. Writing a velocity prediction u and pressure p in dimensional coordinates gives the residuals

$$\begin{aligned} r_m &= \partial u / \partial t + (u \cdot \nabla)u + (1/\rho)\nabla p - \nu \nabla^2 u, \\ r_c &= \nabla \cdot u. \end{aligned}$$

The natural scales of these residuals are U^2/L and U/L , respectively. Their dimensionless counterparts are therefore $(L/U^2)r_m$ and $(L/U)r_c$. A boundary velocity mismatch is normalized by U , while a pressure observation is normalized by ρU^2 . An arbitrary common multiplier cannot produce all of these transformations simultaneously. A pressure gauge must also be specified where the equations otherwise leave an additive constant undetermined.

When $x = Lx^*$ and $t = (L/U)t^*$, differentiation requires $\partial/\partial x = L^{-1}\partial/\partial x^*$ and $\partial/\partial t = (U/L)\partial/\partial t^*$. Standardizing inputs to a convenient interval is not the same operation as deriving the dimensionless equations. If an implementation differentiates with respect to normalized coordinates but uses coefficients from the dimensional PDE, its residual represents a different equation. Automatic differentiation evaluates the implemented computation correctly; it does not decide whether the physical transformation was correct.

The pressure scale requires judgment in a low-Reynolds-number regime. A viscous scale $\mu U/L$ can be more natural than the inertial scale ρU^2 . Either can be valid if the complete equation is transformed consistently. Their relation is governed by Reynolds number, so the selected scale changes the apparent coefficients and conditioning. A parametric surrogate must document whether the scales depend on each case or remain fixed across the training family. Case-dependent normalization can otherwise obscure the meaning of a single global loss.

Mixed-variable laminar-flow models illustrate how output selection changes which residuals are present [6]. Flow-inference and fluid-structure studies introduce additional observation and motion equations [7, 8]. Those equations need their own scales rather than inherited momentum weights. The frequency-compensated flow study changes both representation and physics regularization [3]; its reported benefit should therefore be interpreted together with the loss definition and tested time range.

3. Thermal and mechanical residuals need separate reference quantities

For a conductive thermal problem, let ΔT be a nonzero characteristic temperature difference, T_0 a reference temperature, and t_0 a characteristic time. Set $T = T_0 + \Delta T T^*$. The volumetric energy residual and a surface-flux residual may be written as

$$\begin{aligned} r_T &= \rho c_p \partial T / \partial t - \nabla \cdot (k \nabla T) - Q, \\ r_q &= -k \nabla T \cdot \mathbf{n} - q_b. \end{aligned}$$

With constant properties, the scales $\rho c_p \Delta T / t_0$ and $k \Delta T / L$ apply to these residuals. The dimensionless conduction coefficient is the Fourier number $\alpha t_0 / L^2$, where $\alpha = k / (\rho c_p)$. A convective boundary introduces a Biot number hL/k . These numbers specify physical competition; a loss weight should not erase their effect. When ΔT is small, an absolute scale tied to the intended tolerance is preferable to dividing by a nearly zero measured excursion.

Thermal and melt-pool modeling in additive manufacturing adds moving heating and possibly flow-related transport [9]. Hybrid thermal inference includes uncertain parameters and observations [5]. Thermomechanical prediction adds stress to the learned system [10]. These applications do not share a universal normalization merely because all include temperature. Whether temperature is prescribed, solved, or inferred from data changes the interpretation of the thermal residual and its uncertainty.

For small-strain quasi-static mechanics, equilibrium has units of stress divided by length. A strain-compatibility residual is dimensionless, a constitutive residual has stress units, and an internal-variable evolution residual has its own rate scale. A displacement scale d_0 suggests strain d_0/L . A stress scale may follow a load level or a characteristic material response; choosing the elastic modulus without regard to the loading scale can make the physically relevant stress variation numerically small.

Arora and colleagues combine displacement, stress, plastic strain, and strength outputs and explicitly address scaling of the associated residuals [4]. Their setting is rate- and temperature-dependent monotonic plasticity, not arbitrary cyclic loading. Solid-mechanics field

learning [11] and thermodynamics-based constitutive networks [12] provide complementary examples of why mechanical admissibility depends on the chosen state and equations. A favorable total loss cannot certify an internal variable that was omitted from the model.

4. Worked example: an uncorrected unit change moves the optimum

The following deliberately inconsistent two-target problem isolates the effect of scaling. Let one parameter θ be asked to match targets +1 and -1 through residuals $a(\theta-1)$ and $b(\theta+1)$, with positive weights w_1 and w_2 . The least-squares objective and stationary point are

$$J(\theta) = w_1 a^2 (\theta-1)^2 + w_2 b^2 (\theta+1)^2,$$

$$\theta^* = (w_1 a^2 - w_2 b^2) / (w_1 a^2 + w_2 b^2).$$

Differentiating J and setting its derivative to zero yields the second expression; positivity makes the stationary point the unique minimum. With $a = b = 1$ and equal weights, $\theta^* = 0$. If only the numerical representation of the first residual is multiplied by 1000, the uncorrected optimum becomes approximately 0.999998. Restoring its weight to 10^{-6} restores $\theta^* = 0$. These values follow directly from the stationary-point expression.

Residual scales	Weights	Analytically determined optimum
$a = 1, b = 1$	$w_1 = w_2 = 1$	$\theta^* = 0$
$a = 1000, b = 1$	$w_1 = w_2 = 1$	$\theta^* = 999999/1000001$
$a = 1000, b = 1$	$w_1 = 10^{-6}, w_2 = 1$	$\theta^* = 0$

The targets in this example are intentionally incompatible. In an exactly realizable, noise-free problem, multiplying a residual by a positive constant does not change the common zero set. Real training, however, uses finite models, finite samples, noisy data, and incomplete optimization. Under those conditions the compromise among terms matters. The example explains why a change of units can alter an approximate solution even though the ideal physical equations have not changed.

A similar distinction applies to mean versus sum reductions. Duplicating all samples in one summed term doubles that term and its gradient. Duplicating samples in a mean does not have that deterministic effect. Adding genuinely new points can change the empirical estimate even with a mean, so the invariance is specifically about duplication of the same information. Reproducible studies should report both the sample counts and the reduction convention.

This example also clarifies an engineering tolerance interpretation. A normalized residual r/s compares a mismatch with a reference scale s ; the squared multiplier is $1/s^2$. The scale may be physical, statistical, or a desired numerical tolerance. Those interpretations should not be conflated. In particular, a tolerance selected to prioritize a design quantity is not automatically a measurement standard deviation.

5. Equal losses do not imply equal optimization influence

For a squared residual vector $r(\theta)$, the parameter gradient is $2J_r^T r$, where J_r is the residual Jacobian. Two residual vectors can have equal norms and very different Jacobians. Consider $L_1 = \theta^2$ and $L_2 = 100(\theta - 1.1)^2$ at $\theta = 1$. Both losses equal one, but their derivatives are 2 and -20. Equalizing loss values would not equalize their local influence on the update, and the gradients point in opposite directions.

Gradient-pathology analysis and neural-tangent-kernel studies examine related imbalance and convergence effects in PINNs [13, 14]. Their analyses motivate diagnostics rather than a universal prescription. Gradient norms depend on which parameters are inspected, whether output branches are shared, and how gradients are aggregated. A large norm may indicate influence, but it does not indicate a useful direction. Negative inner products between task gradients can reveal conflict that a magnitude-only rule cannot resolve.

Frequency embeddings further change the residual Jacobian. Differentiation of a sinusoidal feature introduces its frequency, and second derivatives introduce the frequency squared. Fourier-feature mappings and periodic activation networks can therefore alter both representation and conditioning [15, 16]. Multiscale Fourier-feature analysis connects this interaction to PDE learning [17]. Comparing weighted losses across different feature bandwidths without physical diagnostics can mistake a change in objective geometry for improved engineering accuracy.

The shape of the penalty matters as well. A logarithmic residual penalty grows differently from a squared residual and can reduce the influence of large mismatches. For example, $\log(1+|r|)$ is approximately $|r|$ near zero, whereas r^2 is quadratic. If a modified loss uses a norm inside a logarithm, the exact definition of that norm and any preceding squaring must be stated. The modified criterion discussed in the plasticity paper [4] should be reproduced from its mathematical specification, rather than assumed interchangeable with ordinary mean-square error.

Finally, residual imbalance may be a symptom of a poorly specified problem. A wrong boundary condition can produce persistent competition between data and physics. Reducing that term's weight may lower the total loss while concealing the inconsistency. Failure-mode analyses [18] and training guidance [19] support establishing the formulation and an in-range baseline before interpreting an adaptive rule as a physical improvement.

6. What adaptive methods actually control

Pointwise self-adaptive PINNs assign trainable emphasis to difficult locations [20]. Relative-loss balancing methods compare the progress of multiple objectives [21]. Gradient-based strategies instead inspect quantities related to the update direction [13]. These are different control signals. A method that helps a slowly converging boundary term can behave differently from one that concentrates attention on isolated interior residual peaks.

Method family	Information used for adaptation	Main interpretive limitation
Fixed dimensionless weights	Specified scales and priorities	May remain poorly conditioned
Gradient-based balancing	Residual influence on parameters	Norm balance need not resolve direction conflict
Relative-progress balancing	Change of loss over training	Progress depends on noise and initial values
Pointwise attention	Difficulty at individual locations	Changes spatial emphasis unless normalized appropriately
Noise-based data weighting	Assumed observation covariance	Does not represent model discrepancy automatically

Sampling can mimic a weight change. An unweighted empirical mean over points drawn from density q estimates a q -weighted residual norm. Importance sampling can preserve a different target measure when the appropriate density ratio is included [22]. Residual-adaptive studies compare strategies for selecting where equations are tested [23]. A paper that changes both weights and point locations should identify whether it preserves the same objective or intentionally defines a new one.

Causality-aware training changes the temporal organization of a transient optimization [24]. Gradient-enhanced training changes the residual information being penalized [25]. Neither is merely another multiplier on an unchanged collection of samples. Weak-form and domain-decomposed approaches also change the formulation and integration structure [26, 27, 28]. The appropriate comparison is consequently broader than a graph of final scalar losses.

Hard boundary constructions can remove selected penalty terms [29]. Constrained inverse-design methods add another setting in which feasibility and design objectives interact [30]. Here the loss values of constrained and unconstrained architectures have different meanings. A vanishing boundary penalty achieved algebraically should not be credited as a measured improvement in an interior engineering quantity unless the latter is independently evaluated.

7. Evidence needed for an engineering conclusion

A useful comparison begins with explicit equations and physical scales, followed by a fixed-weight dimensionless baseline. Each adaptive method should receive a documented development budget, including searches, derivative evaluations, and unsuccessful runs. Learning curves should show separate components, validation errors, and an engineering output such as drag, heat flux, or peak stress. An average over multiple initializations should be accompanied by its dispersion and failure frequency rather than only the best result.

Convergence theory establishes results under stated assumptions for selected PDE classes [31]. It does not imply that every finite neural optimizer reaches the required minimizer. Likewise,

conditional stability bounds for inverse data-assimilation problems [32] do not remove the need for informative observations. The relation between training residual and physical error is problem dependent. Reporting that relation, where available, is more informative than choosing an arbitrary common stopping loss across different equations.

When measurements are noisy, covariance-weighted discrepancies can be interpreted through a likelihood. Bayesian PINNs provide a framework for propagating observation uncertainty under model assumptions [33]. Combining low- and high-fidelity information adds discrepancy that should be modeled or diagnosed separately [34]. A weight that suppresses disagreement with an imperfect simulator may improve prediction, but it should not be presented as an estimate of instrument precision.

Implementation frameworks such as DeepXDE help reproduce residual constructions [35]. Benchmark resources provide broader comparison contexts [36, 37]. Neither replaces a unit-conversion check, a chain-rule verification, or an independent engineering diagnostic. Conventional finite-element comparisons further caution against equating low neural inference cost with low total solution cost [38]. Training and tuning belong in the accounting when assessing whether a balancing method is practically worthwhile.

8. Conclusions

Dimensionally consistent loss construction is a prerequisite for interpreting adaptive balancing. Physical scaling determines the equation represented; statistical weighting determines an observation model; numerical adaptation determines how a finite optimizer allocates effort. The analytical examples demonstrate that these distinctions can change the optimum and the update even in a one-parameter problem. Larger neural networks do not eliminate them.

For flow and thermomechanical surrogates, the defensible sequence is to verify the equations, normalize variables and residuals, document sampling and reductions, establish a fixed baseline, and then assess adaptation using independent quantities of interest. The objective is not equal numerical values for every term. It is a transparent approximation whose errors in the engineering decision can be measured and whose remaining limitations can be traced to formulation, data, representation, or optimization.

References

- [1] Maziar Raissi, Paris Perdikaris, George Em 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). DOI.
- [2] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, et al.. Physics-informed machine learning. *Nature Reviews Physics* 3, 422-440 (2021). DOI.
- [3] Tongtao Zhang, Biswadip Dey, Pratik Kakkar, Arindam Dasgupta, Amit Chakraborty. Frequency-compensated PINNs for Fluid-dynamic Design Problems. *ML4Eng Workshop, NeurIPS 2020*. arXiv:2011.01456 (2020). arXiv.
- [4] Rajat Arora, Pratik Kakkar, Amit Chakraborty, Biswadip Dey. Physics-informed neural networks for modeling rate- and temperature-dependent plasticity. *Machine Learning and the Physical Sciences Workshop, NeurIPS 2022*. arXiv:2201.08363v3 (2022). arXiv.

- [5] Shuheng Liao, Tianju Xue, Jihoon Jeong, et al.. Hybrid thermal modeling of additive manufacturing processes using physics-informed neural networks for temperature prediction and parameter identification. arXiv:2206.07756 (2022). arXiv.
- [6] Chengping Rao, Hao Sun, Yang Liu. Physics-informed deep learning for incompressible laminar flows. arXiv:2002.10558 (2020). arXiv.
- [7] Maziar Raissi, Alireza Yazdani, George Em Karniadakis. Hidden Fluid Mechanics: A Navier-Stokes Informed Deep Learning Framework for Assimilating Flow Visualization Data. arXiv:1808.04327 (2018). arXiv.
- [8] Maziar Raissi, Zhicheng Wang, Michael S. Triantafyllou, George Em Karniadakis. Deep Learning of Vortex Induced Vibrations. arXiv:1808.08952 (2018). arXiv.
- [9] Qiming Zhu, Zeliang Liu, Jinhui Yan. Machine learning for metal additive manufacturing: Predicting temperature and melt pool fluid dynamics using physics-informed neural networks. arXiv:2008.13547 (2020). arXiv.
- [10] R. Sharma, Y. B. Guo. Thermal-Mechanical Physics Informed Deep Learning For Fast Prediction of Thermal Stress Evolution in Laser Metal Deposition. arXiv:2412.18786 (2024). arXiv.
- [11] Ehsan Haghighat, Maziar Raissi, Adrian Moure, et al.. A deep learning framework for solution and discovery in solid mechanics. arXiv:2003.02751 (2020). arXiv.
- [12] Filippo Masi, Ioannis Stefanou, Paolo Vannucci, Victor Maffi-Berthier. Thermodynamics-based Artificial Neural Networks for constitutive modeling. arXiv:2005.12183 (2020). arXiv.
- [13] Sifan Wang, Yujun Teng, Paris Perdikaris. Understanding and mitigating gradient pathologies in physics-informed neural networks. arXiv:2001.04536 (2020). arXiv.
- [14] Sifan Wang, Xinling Yu, Paris Perdikaris. When and why PINNs fail to train: A neural tangent kernel perspective. arXiv:2007.14527 (2020). arXiv.
- [15] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, et al.. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. arXiv:2006.10739 (2020). arXiv.
- [16] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, et al.. Implicit Neural Representations with Periodic Activation Functions. arXiv:2006.09661 (2020). arXiv.
- [17] Sifan Wang, Hanwen Wang, Paris Perdikaris. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. arXiv:2012.10047 (2020). arXiv.
- [18] Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, et al.. Characterizing possible failure modes in physics-informed neural networks. arXiv:2109.01050 (2021). arXiv.
- [19] Sifan Wang, Shyam Sankaran, Hanwen Wang, Paris Perdikaris. An Expert's Guide to Training Physics-informed Neural Networks. arXiv:2308.08468 (2023). arXiv.
- [20] Levi McClenny, Ulisses Braga-Neto. Self-Adaptive Physics-Informed Neural Networks using a Soft Attention Mechanism. arXiv:2009.04544 (2020). arXiv.
- [21] Rafael Bischof, Michael Kraus. Multi-Objective Loss Balancing for Physics-Informed Deep Learning. arXiv:2110.09813 (2021). arXiv.
- [22] Mohammad Amin Nabian, Rini Jasmine Gladstone, Hadi Meidani. Efficient training of physics-informed neural networks via importance sampling. arXiv:2104.12325 (2021). arXiv.
- [23] Chenxi Wu, Min Zhu, Qinyang Tan, et al.. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. arXiv:2207.10289 (2022). arXiv.
- [24] Sifan Wang, Shyam Sankaran, Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks. arXiv:2203.07404 (2022). arXiv.
- [25] Jeremy Yu, Lu Lu, Xuhui Meng, George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. arXiv:2111.02801 (2021). arXiv.
- [26] E. Kharazmi, Z. Zhang, G. E. Karniadakis. Variational Physics-Informed Neural Networks For Solving Partial Differential Equations. arXiv:1912.00873 (2019). arXiv.

- [27] Ehsan Kharazmi, Zhongqiang Zhang, George Em Karniadakis. hp-VPINNs: Variational Physics-Informed Neural Networks With Domain Decomposition. arXiv:2003.05385 (2020). arXiv.
- [28] Ben Moseley, Andrew Markham, Tarje Nissen-Meyer. Finite Basis Physics-Informed Neural Networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. arXiv:2107.07871 (2021). arXiv.
- [29] N. Sukumar, Ankit Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. arXiv:2104.08426 (2021). arXiv.
- [30] Lu Lu, Raphael Pestourie, Wenjie Yao, et al.. Physics-informed neural networks with hard constraints for inverse design. arXiv:2102.04626 (2021). arXiv.
- [31] Yeonjong Shin, Jerome Darbon, George Em Karniadakis. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. arXiv:2004.01806 (2020). arXiv.
- [32] Siddhartha Mishra, Roberto Molinaro. Estimates on the generalization error of Physics Informed Neural Networks (PINNs) for approximating a class of inverse problems for PDEs. arXiv:2007.01138 (2020). arXiv.
- [33] Liu Yang, Xuhui Meng, George Em Karniadakis. B-PINNs: Bayesian Physics-Informed Neural Networks for Forward and Inverse PDE Problems with Noisy Data. arXiv:2003.06097 (2020). arXiv.
- [34] Xuhui Meng, George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. arXiv:1903.00104 (2019). arXiv.
- [35] Lu Lu, Xuhui Meng, Zhiping Mao, George E. Karniadakis. DeepXDE: A deep learning library for solving differential equations. arXiv:1907.04502 (2019). arXiv.
- [36] Zhongkai Hao, Jiachen Yao, Chang Su, et al.. PINNacle: A Comprehensive Benchmark of Physics-Informed Neural Networks for Solving PDEs. arXiv:2306.08827 (2023). arXiv.
- [37] Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, et al.. PDEBENCH: An Extensive Benchmark for Scientific Machine Learning. arXiv:2210.07182 (2022). arXiv.
- [38] Tamara G. Grossmann, Urszula Julia Komorowska, Jonas Latz, Carola-Bibiane Schönlieb. Can Physics-Informed Neural Networks beat the Finite Element Method?. arXiv:2302.04107 (2023). arXiv.