Towards Accurate Test-Time Adaptation for Neural Surrogates

Anonymous Author(s)
Affiliation
Address
email

Abstract

Neural surrogates that map input configurations (e.g., initial conditions and meshes) to simulation outputs are increasingly used in practical applications such as engineering design optimization. However, pre-trained models often experience significant performance drop on unseen problem configurations, such as different geometries, structural dimensions, and physical parameters. Test-Time Adaptation (TTA) mitigates distribution shifts by leveraging target configurations, online and at test-time. It avoids the need for costly re-training and doesn't require access to the original dataset, which is typically unavailable in practice. In this work we propose Representation Alignment for Simulations (SimRA), a novel method to improve performance at deployment, specific for multi-dimensional regression on simulation data. SimRA extends prior work on univariate regression [Adachi et al., ICLR 2025] with a novel feature weighting mechanism, ensuring stability in high-dimensional simulation settings. To our knowledge, this is the first study of TTA for neural surrogates. Empirical evaluations on diverse engineering tasks demonstrate strong performance and highlight the potential of TTA in the field.

1 Introduction

Neural surrogates have become powerful tools for solving Partial Differential Equation (PDE) simulations in engineering and science. They perform well when test conditions match the training data, but performance often drops on novel configurations (geometry, material types, structural dimensions, and physical parameters), i.e., when the data distribution shifts. This problem often arises in industrial design optimization, where parameters can vary significantly across iterations and go beyond the ranges known a priori. Furthermore, in such cases, access to the original source data is often limited by portability or proprietary restrictions, which makes zero-shot or source-free test-time adaptation crucial for practical deployment.

Several approaches have been proposed to address distribution shifts, including domain generalization [7], meta-learning [16], and active learning [36]. Unfortunately, many of these methods are impractical for engineering tasks where rapid adaptation is essential. In contrast, Test-Time Adaptation (TTA) adapts models at inference without source data and with minimal additional training effort [25, 40, 46]. TTA has proven effective in many domains, including medical imaging, object detection, and segmentation. While many works are known for classification [46, 56, 31, 53, 55, 17, 20, 10, 2, 18], comparably little research can be found for regression [27]. One outstanding method is Significant-Subspace Alignment (SSA) [3], capable of handling both classification and regression tasks. It is however restricted to one-dimensional regression outputs and depends on manual selection of feature parameters, potentially causing instability for high-dimensional data.

Submitted to the AI for Science workshop (NeurIPS 2025). Do not distribute.

Our contributions are summarized as follows:

- SimRA is the first TTA method for simulations, to our knowledge. SimRA not only consistently outperforms SSA, but also eliminates the need for feature pre-selection.
- We evaluate TTA on distribution shifts in industrial settings, trying to cover diverse configurations from realistic engineering design scenarios, on SIMSHIFT datasets [35].
- Since TTA for physical neural surrogates remains largely unexplored, we identify promising opportunities for innovation at the intersection of physics and adaptive machine learning.

2 Related Work

Neural surrogates have emerged as a widely used approach to accelerate traditional numerical simulation methods, by providing fast approximations of the solutions. In general, surrogate models are trained on the solutions from numerical solvers, paired with the corresponding initial conditions and configurations under which they were generated, e.g., [35, 8, 44, 43]. A particularly prominent line of work within neural surrogate modeling for PDEs is operator learning [21, 24, 28, 4, 49]. Such models aim to directly approximate the solution operator that maps initial functions (conditions and input terms) to output functions.

Test-Time Adaptation (TTA) refers to the emerging machine learning technique of adapting a pre-trained model to unlabeled target data, directly at inference time and prior to generating predictions. For this reason, TTA has recently attracted increasing attention as it offers a (nearly) free performance gain [27]. While the majority of existing TTA methods have been developed for low-dimensional classification tasks [26, 51], employing methodologies such as entropy minimization [46, 56, 31, 53, 55] and feature alignment [17, 20, 10, 2, 18], recent works have begun to extend these ideas to image segmentation [45, 15, 19]. Research in regression problems is very sparse, and standard TTA methods cannot be trivially applied. One potential reason is the use of Mean Squared Error in regression problems, which often leads to a focus on a narrow set of predictive features, reducing diversity [54]. Significant-Subspace Alignment (SSA) [3] addresses these limitations by selecting and aligning the important feature dimensions, and shows positive performance in the one-dimensional cases. In this work, we extend and refine SSA for neural surrogates and resolve instability issues arising in the high-dimensional regression setting. Finally, TTA should not be confused with Test-Time Training (TTT), often used in time series literature [39, 47, 42, 41]. While both solve the same problem, TTT typically refers to methods that employ time-series specific techniques, for example updating hidden states during sequential inference.

Domain generalization, meta-learning, and active learning represent alternative strategies that can be used to improve model robustness and generalization under distribution shifts. Domain generalization [29, 23] and Unsupervised Domain Adaptation (UDA) [38, 14, 52, 13] While effective in some scenarios, their reliance on specific training, model selection and diverse training distributions limits their applicability. Meta-learning methods [12] and active learning [22, 30] are similarly motivated, but generally assume access to ground-truth information in the shifted domain. In our setting, all these approaches face a significant practical limitation: none of them can quickly adapt a pre-trained model leveraging unlabeled data at test-time, as they all rely on a priori knowledge and training. This motivates our exploration of TTA as a more suitable solution.

3 Problem

Following [50, 27], we assume access to a regressor $f_\theta : \mathcal{X} \to \mathbb{R}^d$ pre-trained on a *source* sample $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^{N^{\text{src}}} \in \mathcal{X} \times \mathbb{R}^d$ drawn from a source distribution P^{src}, e.g., $f_\theta = g \circ \phi$ in Fig. 1. We also assume access to some real matrix-valued source statistics $\Sigma^{\text{src}}, \mu^{\text{src}}, \sigma^{\text{src}}$.

The goal is, for any new *unlabeled* sample $(\mathbf{x}_i^{\text{tgt}})_{i=1}^{N^{\text{tgt}}}$ drawn from the input marginal of a *target* distribution $P^{\text{tgt}} \neq P^{\text{src}}$, to find θ^{tgt} (using the source statistics but not the source sample) which minimizes the risk

$$\mathcal{R}(f_\theta) = \frac{1}{N^{\text{tgt}}} \sum_{i=1}^{N^{\text{tgt}}} \left\| f_\theta(\mathbf{x}_i^{\text{tgt}}) - \mathbf{y}_i^{\text{tgt}} \right\|_2^2 .$$

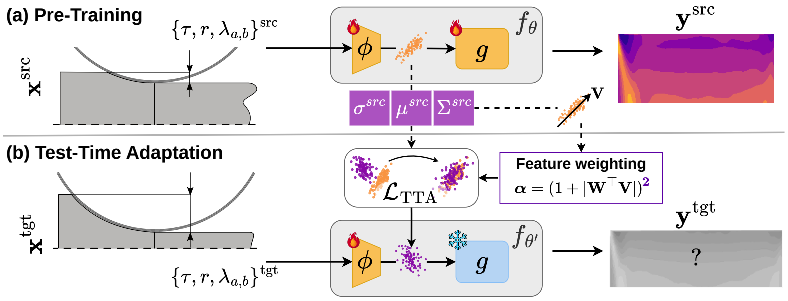

(a) Pre-Training
$\{\tau, r, \lambda_{a,b}\}^{\text{src}}$
$\mathbf{x}^{\text{src}}$
ϕ
g
f_θ
$\mathbf{y}^{\text{src}}$
σ^{src} μ^{src} Σ^{src}
$\mathbf{V}$
(b) Test-Time Adaptation
$\mathcal{L}_{\text{TTA}}$
Feature weighting
$\boldsymbol{\alpha} = (1 + |\mathbf{W}^\top \mathbf{V}|)^2$
$\mathbf{y}^{\text{tgt}}$
$\mathbf{x}^{\text{tgt}}$
$\{\tau, r, \lambda_{a,b}\}^{\text{tgt}}$
$f_{\theta'}$
?

Figure 1: Overview of training and test-time adaptation. (a) Pre-training on the source domain using input parameters: thickness (τ), post-rolling reduction (r), and temperature coefficients (λ_a, λ_b). The *representation learner* ϕ and the *predictor* g are optimized jointly. (b) Test-time adaptation on the target domain, where only the input parameters are available. Here ϕ is adapted and g is frozen.

Note that we have no access to any target labels $(\mathbf{y}_i^{\text{tgt}})_{i=1}^{N^{\text{tgt}}}$. That is, the target risk $\mathcal{R}(f_\theta)$ cannot be evaluated directly and importance weighting [37] cannot be used without further modification.

We study the problem above using the SIMSHIFT benchmark dataset [35], designed to evaluate how surrogate models adapt to distribution shifts on real-world industrial simulation tasks. In the benchmark, the inputs $\mathbf{x}$ represent parameters like geometry, material properties, or operating conditions, while the labels $\mathbf{y}$ correspond to high-dimensional fields, such as stresses, deformation and temperature. The target distribution originates from unseen parameter configurations and the goal is to predict the corresponding fields.

4 Method

We build on SSA [3], extending it to handle multi-output regression for simulation-based tasks. In the original formulation, the regression network is divided into two connected parts: a *representation learner* ϕ that produces intermediate features, and a *predictor* g that maps these features to outputs. Adaptations occur in the representation stage, while the predictor remains unchanged. Fig. 1 sketches this process using hot rolling as an example, and distinguishes between (a) pretraining and (b) TTA with SimRA.

The idea is to adjust the features $\mathbf{z} := \phi(\mathbf{x})$, $\mathbf{z} \in \mathbb{R}^C$ such that the target features are similarly distributed as the source features. During TTA, target batch statistics are computed on-the-fly, while source statistics $\Sigma^{\text{src}}, \mu^{\text{src}}, \sigma^{\text{src}}$ (pre-computed and stored after training) are used to align the source and target feature distributions by minimizing the Kullback-Leibler divergence (see Appendix A).

A central element of SSA is its *pre-computed significant subspace*, which retains only the dominant eigenvalues by using a fixed, manually pre-selected subset of K principal components. Instead, we use all feature directions and apply *dimension weighting* via exponentiation of the weighting function:

$$\boldsymbol{\alpha} = (1 + |\mathbf{W}^\top \mathbf{V}^{\text{src}}|)^2, \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{K \times C}$ are weights from the first layer of the predictor g (for a C-dimensional $\mathbf{z}$) and $\mathbf{V}^{\text{src}} \in \mathbb{R}^{K \times K}$ is the principal component basis of the source features. Representation alignment is applied to the target features $\mathbf{z}^{\text{tgt}} := \phi(\mathbf{x}^{\text{tgt}})$. Each channel $\mathbf{z}_c^{\text{tgt}}$ is projected onto the source basis $\mathbf{V}^{\text{src}}$, reweighted by the corresponding factor $\boldsymbol{\alpha}_c$ with $c \in [0, C-1]$ as

$$\tilde{\mathbf{z}}_c^{\text{tgt}} \;=\; \left(\mathbf{z}_c^{\text{tgt}} - \mu^{\text{src}}\right) \mathbf{V}^{\text{src}} \boldsymbol{\alpha}_c, \tag{2}$$

where $\boldsymbol{\alpha}_c$ is the c-th row of $\boldsymbol{\alpha}$ (channel-wise). By using Eqs. (1) and (2) to do feature selection, SimRA can preserve a richer set of features, improving adaptation under distribution shift for

Table 1: Comparison of current baselines with TTA methods for all simulation datasets. Results are averaged across 20 TTA runs, over 2 models (40 seeds in total) with standard deviation reported.

(a) Rolling

Model	RMSE (↓)	MAE (↓)	R^2 (↑)
Source	$0.723_{\pm 0.046}$	$0.419_{\pm 0.014}$	$0.860_{\pm 0.018}$
UDA	$0.644_{\pm 0.041}$	$0.399_{\pm 0.012}$	$0.869_{\pm 0.017}$
SSA	$0.735_{\pm 0.097}$	$0.441_{\pm 0.055}$	$0.854_{\pm 0.039}$
SimRA	$\mathbf{0.699}_{\pm \mathbf{0.064}}$	$\mathbf{0.418}_{\pm \mathbf{0.041}}$	$\mathbf{0.868}_{\pm \mathbf{0.031}}$

(b) Motor

Model	RMSE (↓)	MAE (↓)	R^2 (↑)
Source	$0.127_{\pm 0.002}$	$0.061_{\pm 0.002}$	$0.987_{\pm 0.001}$
UDA	$0.119_{\pm 0.001}$	$0.061_{\pm 0.000}$	$0.987_{\pm 0.001}$
SSA	$0.125_{\pm 0.002}$	$0.062_{\pm 0.002}$	$0.986_{\pm 0.001}$
SimRA	$\mathbf{0.124}_{\pm \mathbf{0.002}}$	$\mathbf{0.061}_{\pm \mathbf{0.001}}$	$\mathbf{0.987}_{\pm \mathbf{0.001}}$

(c) Forming

Model	RMSE (↓)	MAE (↓)	R^2 (↑)
Source	$0.166_{\pm 0.020}$	$0.055_{\pm 0.006}$	$0.982_{\pm 0.004}$
UDA	$0.154_{\pm 0.009}$	$0.052_{\pm 0.003}$	$0.984_{\pm 0.002}$
SSA	$0.170_{\pm 0.026}$	$0.057_{\pm 0.010}$	$0.981_{\pm 0.006}$
SimRA	$\mathbf{0.164}_{\pm \mathbf{0.018}}$	$\mathbf{0.054}_{\pm \mathbf{0.006}}$	$\mathbf{0.983}_{\pm \mathbf{0.004}}$

(d) Heatsink

Model	RMSE (↓)	MAE (↓)	R^2 (↑)
Source	$0.634_{\pm 0.012}$	$0.424_{\pm 0.004}$	$0.484_{\pm 0.027}$
UDA	$0.577_{\pm 0.005}$	$0.374_{\pm 0.001}$	$0.553_{\pm 0.002}$
SSA	$0.632_{\pm 0.014}$	$0.424_{\pm 0.003}$	$0.484_{\pm 0.026}$
SimRA	$\mathbf{0.631}_{\pm \mathbf{0.014}}$	$\mathbf{0.423}_{\pm \mathbf{0.003}}$	$\mathbf{0.485}_{\pm \mathbf{0.026}}$

simulation datasets. Furthermore, since our targets are vector-valued, we compute the Kullback-Leibler divergence independently for each channel, rather than combining them into a single space.

5 Experiments

Performance. To investigate the performance of the proposed method, we use the SIMSHIFT datasets [35], which span four distinct industrial simulation settings: *hot rolling*, *sheet metal forming*, *electric motor*, and *heatsink design*. All datasets have explicit source and target domain splits, dependent on the physical and meshing parameters used to generate the samples. Shifts happen in parametric space, as opposed to unstructured variations occurring in images. For insights into the dataset components and their creation, please refer to the Appendix of SIMSHIFT [35].

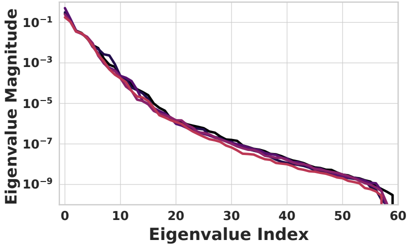
Table 1 summarizes the results across all datasets, comparing our method against SSA, *"unregularized"* pre-trained predictions (*"Source"*), and Unsupervised Domain Adaptation (UDA) applied to the pre-trained model. For implementation details refer to Appendix B. SimRA consistently outperforms SSA, establishing a new baseline for test-time adaptation in neural surrogate regression. While improvements over the source model may be marginal in some cases, SSA can destabilize the pre-trained model, whereas our approach does not degrade performance. Moreover, when using UDA as a lower bound, our method reduces the gap without fully closing it, leaving room for future improvement.

Interpretation and ablations. Different datasets can experience varying degrees of improvement from TTA, highlighting the unique complexities of each problem. By analyzing the eigenvalue distribution (Fig. 2b for hot rolling), we observe that improvements correlate with the explanatory power of the leading eigenvalues: datasets where a few components capture most of the variance exhibit stronger adaptation gains. In contrast, in the motor dataset, variance is spread across many eigenvalues, indicating a higher-dimensional problem structure and limiting the effectiveness of current TTA methods. See Appendix C for the full eigenvalue analysis.

To illustrate this point, we ablate the impact of the number of dimensions K of the feature subspace for the standard SSA algorithm in Fig. 2. Originally, K has to be chosen manually from the eigenvalue spectrum of the covariance matrix, requiring expert interaction. While the variance in the hot rolling datasets decays sharply after the first ten directions (Fig. 2b), several low-variance components remain correlated with regression targets. SSA results in Fig. 2a suggest that a handpicked K might not be optimal, and strict truncation can discard relevant information. For multivariate regression problems, feature selection is thus critical. Importantly, this choice is absent in SimRA, simplifying the tuning and yielding superior results. Fig. 2a shows that across different K-values, our method always outperforms SSA, irrespective to the chosen subspace size.

Method	K	RMSE ↓	MAE ↓
SSA	10	$0.859_{\pm 0.094}$	$0.548_{\pm 0.073}$
	20	$0.735_{\pm 0.097}$	$0.441_{\pm 0.055}$
	30	$0.736_{\pm 0.067}$	$0.441_{\pm 0.036}$
	50	$0.774_{\pm 0.057}$	$0.467_{\pm 0.032}$
	All	$0.827_{\pm 0.043}$	$0.508_{\pm 0.085}$
SimRA (ours)	All	$\mathbf{0.699}_{\pm 0.064}$	$\mathbf{0.418}_{\pm 0.041}$

(a) Comparison of SimRA with SSA for different choices of k.



(b) Eigenvalue analysis.

Figure 2: Ablations with SSA [3] for the hot rolling dataset. (a) table with quantitative comparison, (b) eigenvalues analysis for different trained models, highlighting the fast decay. ∼60 % of the energy is on the first eigenvalues, favoring compact representations.

6 Conclusion and Future Work

In this work, we make the initial step towards highly-accurate test-time adaptation methods for neural surrogates, and in general for high-dimensional multivariate regression. Our findings show that the proposed adjustments enable TTA to yield zero-shot improvements at negligible computational cost. Furthermore, the current state-of-the-art method, SSA [3], can be substantially simplified, removing the need for feature preselection, while also achieving improved performance and stability. The promotion can be interpreted via eigen-decomposition analysis of the feature space, offering insights into why our approach outperforms existing methods on the evaluated datasets.

In addition to the zero-cost gains, this line of research is particularly timely due to evolving compliance requirements. Article 15 of the EU Artificial Intelligence Act states that high-risk AI systems need to ensure appropriate levels of accuracy and robustness [1]. Should neural surrogates be deployed in safety critical domains, such as accelerating structural design in the automotive industry, accurate and reliable predictions becomes indispensable.

However, performance improvements only occur on some datasets, and the lower bounds established by UDA indicate that additional gains remain attainable. This points to the potential for a new class of TTA algorithms, specifically developed for physics simulation data. We foresee two paths to achieve *"physics-driven"* TTA that are to be explored: (i) use physics-informed constraints and priors [34, 9], ad-hoc and calibrated on the test case, to augment the expressiveness of the limited available test data, and (ii) incorporate uncertainty quantification to localize failure regions in the fields where adaptation is necessary. Orthogonally, exploring the impact of TTA in data-driven design optimization [11, 6, 33] represents another promising avenue for research. A concrete example would be the *EngiBench* dataset [11], which provides a standardized machine learning benchmark for engineering design.

References

[1] Artificial intelligence act (eu regulation 2024/1689), article 15: Accuracy, robustness and cybersecurity. `artificialintelligenceact.eu/article/15`, 2024. EU Artificial Intelligence Act (in force since 1 August 2024), Article 15 mandates appropriate levels of accuracy and robustness for high-risk AI systems, including resilience to errors, faults or inconsistencies.

[2] Kazuki Adachi, Shin'Ya Yamaguchi, and Atsutoshi Kumagai. Covariance-aware feature alignment with pre-computed source statistics for test-time adaptation to multiple image corruptions. In *2023 IEEE International Conference on Image Processing (ICIP)*, pp. 800–804. IEEE, 2023.

[3] Kazuki Adachi, Shin'ya Yamaguchi, Atsutoshi Kumagai, and Tomoki Hamagami. Test-time adaptation for regression by subspace alignment. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[4] Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 25152–25194.

Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2cd36d327f33d47b372d4711edd08de0-Paper-Conference.pdf.

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

[6] Arturs Berzins, Andreas Radler, Eric Volkmann, Sebastian Sanokowski, Sepp Hochreiter, and Johannes Brandstetter. Geometry-informed neural networks, 2025. URL https://arxiv.org/abs/2402.14009.

[7] Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *Journal of machine learning research*, 22 (2):1–55, 2021.

[8] Florent Bonnet, Jocelyn Ahmed Mazari, Paola Cinnella, and Patrick Gallinari. AirfRANS: High fidelity computational fluid dynamics dataset for approximating reynolds-averaged navier–stokes solutions. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://arxiv.org/abs/2212.07564.

[9] Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (pinns) for fluid mechanics: A review, 2021. URL https://arxiv.org/abs/2105.09506.

[10] Cian Eastwood, Ian Mason, Christopher KI Williams, and Bernhard Schölkopf. Source-free adaptation to measurement shift via bottom-up feature restoration. *arXiv preprint arXiv:2107.05446*, 2021.

[11] Florian Felten, Gabriel Apaza, Gerhard Bräunlich, Cashen Diniz, Xuliang Dong, Arthur Drake, Milad Habibi, Nathaniel J. Hoffman, Matthew Keeler, Soheyl Massoudi, Francis G. VanGessel, and Mark Fuge. Engibench: A framework for data-driven engineering design research, 2025. URL https://arxiv.org/abs/2508.00831.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint*, 2017. arXiv:1703.03400.

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks, 2015.

[14] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. In B. Schölkopf, J. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL https://proceedings.neurips.cc/paper_files/paper/2006/file/e9fb2eda3d9c55a0d89c98d6c54b5b3e-Paper.pdf.

[15] Yufan He, Aaron Carass, Lianrui Zuo, Blake E. Dewey, and Jerry L. Prince. Autoencoder based self-supervised test-time adaptation for medical image analysis. *Medical Image Analysis*, 72:102136, 2021. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2021.102136. URL https://www.sciencedirect.com/science/article/pii/S1361841521001821.

[16] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44 (9):5149–5169, 2021.

[17] Masato Ishii and Masashi Sugiyama. Source-free domain adaptation via distributional alignment by matching batch normalization statistics. *arXiv preprint arXiv:2101.10842*, 2021.

[18] Sanghun Jung, Jungsoo Lee, Nanhee Kim, Amirreza Shaban, Byron Boots, and Jaegul Choo. Cafa: Class-aware feature alignment for test-time adaptation. In *International Conference on Computer Vision (ICCV)*, pp. 19060–19071, 2023.

[19] Neerav Karani, Ertunc Erdil, Krishna Chaitanya, and Ender Konukoglu. Test-time adaptable neural networks for robust medical image segmentation. *Medical Image Analysis*, 68:101907, February 2021. ISSN 1361-8415. doi: 10.1016/j.media.2020.101907. URL http://dx.doi.org/10.1016/j.media.2020.101907.

[20] Takeshi Kojima, Yutaka Matsuo, and Yusuke Iwasawa. Robustifying vision transformer without retraining from scratch by test-time class-conditional feature alignment. *arXiv preprint arXiv:2206.13951*, 2022.

[21] Nikola B. Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. *CoRR*, abs/2108.08481, 2021. URL `https://arxiv.org/abs/2108.08481`.

[22] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. *arXiv preprint*, 1994. arXiv:cmp-lg/9407020.

[23] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization, 2017. URL `https://arxiv.org/abs/1710.03463`.

[24] Zongyi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *CoRR*, abs/2003.03485, 2020.

[25] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning (ICML)*, pp. 6028–6039. PMLR, 2020.

[26] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation, 2021. URL `https://arxiv.org/abs/2002.08546`.

[27] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, July 2024. ISSN 1573-1405. doi: 10.1007/s11263-024-02181-w. URL `http://dx.doi.org/10.1007/s11263-024-02181-w`.

[28] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, March 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5. URL `http://dx.doi.org/10.1038/s42256-021-00302-5`.

[29] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 10–18, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL `https://proceedings.mlr.press/v28/muandet13.html`.

[30] Daniel Musekamp, Marimuthu Kalimuthu, David Holzmüller, Makoto Takamoto, and Mathias Niepert. Active learning for neural PDE solvers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=x4ZmQaumRg`.

[31] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

[32] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4196–4206, 2023. URL `https://arxiv.org/abs/2212.09748`.

[33] Andreas Radler, Eric Volkmann, Johannes Brandstetter, and Arturs Berzins. Diverse topology optimization using modulated neural fields, 2025. URL `https://arxiv.org/abs/2502.13174`.

[34] Maziar Raissi, Paris Perdikaris, and 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: 10.1016/j.jcp.2018.10.045.

[35] Paul Setinek, Gianluca Galletti, Thomas Gross, Dominik Schnürer, Johannes Brandstetter, and Werner Zellinger. Simshift: A benchmark for adapting neural surrogates to distribution shifts, 2025. URL `https://arxiv.org/abs/2506.12007`.

[36] Burr Settles. Active learning literature survey. 2009.

[37] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000. ISSN 0378-3758. doi: https://doi.org/10.1016/S0378-3758(00)00115-4. URL `https://www.sciencedirect.com/science/article/pii/S0378375800001154`.

[38] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation, 2016. URL `https://arxiv.org/abs/1607.01719`.

[39] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. 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. 9229–9248. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/sun20b.html`.

[40] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning (ICML)*, pp. 9229–9248. PMLR, 2020.

[41] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts, 2020. URL `https://arxiv.org/abs/1909.13231`.

[42] Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): Rnns with expressive hidden states, 2025. URL `https://arxiv.org/abs/2407.04620`.

[43] Artur Toshev, Harish Ramachandran, Jonas A. Erbesdobler, Gianluca Galletti, Johannes Brandstetter, and Nikolaus A. Adams. JAX-SPH: A differentiable smoothed particle hydrodynamics framework. In *ICLR 2024 Workshop on AI4DifferentialEquations In Science*, 2024. URL `https://openreview.net/forum?id=8X5PXVmsHW`.

[44] Artur P. Toshev, Gianluca Galletti, Fabian Fritz, Stefan Adami, and Nikolaus A. Adams. Lagrangebench: a lagrangian fluid mechanics benchmarking suite. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, 2023.

[45] Jeya Maria Jose Valanarasu, Pengfei Guo, Vibashan VS, and Vishal M. Patel. On-the-fly test-time adaptation for medical image segmentation. In *MIDL (Medical Imaging with Deep Learning)*, 2023. Episodic, zero-shot adaptation with adaptive batch-normalization.

[46] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

[47] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=uXl3bZLkr3c`.

[48] Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53681–53705. PMLR, 21–27 Jul 2024. URL `https://proceedings.mlr.press/v235/wu24r.html`.

[49] Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. In *International Conference on Machine Learning*, 2024.

[50] Zehao Xiao and Cees GM Snoek. Beyond model adaptation at test time: A survey. *arXiv preprint arXiv:2411.03687*, 2024.

[51] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Exploiting the intrinsic neighborhood structure for source-free domain adaptation, 2021. URL `https://arxiv.org/abs/2110.04202`.

[52] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning, 2019. URL `https://arxiv.org/abs/1702.08811`.

[53] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.

[54] Shihao Zhang, Linlin Yang, Michael Bi Mi, Xiaoxu Zheng, and Angela Yao. Improving deep regression with ordinal entropy. *International Conference on Learning Representations*, 2023.

[55] Bowen Zhao, Chen Chen, and Shu-Tao Xia. Delta: degradation-free fully test-time adaptation. *arXiv preprint arXiv:2301.13018*, 2023.

[56] Aurick Zhou and Sergey Levine. Bayesian adaptation for covariate shift. *Advances in neural information processing systems*, 34:914–927, 2021.

A Supplementary Approach Information

Significant-Subspace Alignment is a TTA method for one-dimensional regression. It consists of two steps: *feature alignment* and *significant-subspace alignment*. In the first step, source statistics such as mean μ^{src} and covariance Σ^{src} are computed after source training. In the second step, a significant subspace is detected by selecting the top eigenvalues λ_k of the source covariance Σ^{src}. Each subspace direction v_k^{src} is then weighted by its influence on the regression output:

$$\boldsymbol{\alpha}_k = 1 + |\mathbf{w}^\top \mathbf{v}_k^{\text{src}}|,$$

where $\boldsymbol{\alpha}_k \geq 1$ ensures that dimensions that strongly affect the regression output are emphasized.

At test time, the precomputed source statistics are used to project the target features into the significant subspace. From the projected target features, their mean and variance $(\tilde{\mu}_k^{\text{tgt}}, \tilde{\sigma}_k^{\text{tgt}2})$ are calculated and aligned with the corresponding source statistics $(0, \lambda_k^{\text{src}})$. The adaptation objective is a weighted symmetric Kullback-Leibler divergence between assumed Normal distributions:

$$\mathcal{L}_{\text{TTA}} = \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{\alpha}_k \left(\frac{(\tilde{\mu}_k^{\text{tgt}})^2 + \lambda_k^{\text{src}}}{\tilde{\sigma}_k^{\text{tgt}2}} + \frac{(\tilde{\mu}_k^{\text{tgt}})^2 + \tilde{\sigma}_k^{\text{tgt}2}}{\lambda_k^{\text{src}}} - 2 \right). \tag{3}$$

B Experimental Setup

In the following paragraphs, we detail the experimental setup, including the selected models and our training and testing strategy, based on established methods [3].

B.1 Model Architecture

We employ a single model architecture to evaluate our TTA method. The model is taken from the SIMSHIFT benchmark [35], implemented in PyTorch, and designed for conditional regression. Node coordinates are provided as inputs and embedded using sinusoidal positional encodings. Conditioning is applied through a dedicated network that processes the simulation input parameters.

Conditioning Network. The conditioner maps simulation parameters into a latent representation of dimension 8. It consists of a sinusoidal encoding, followed by a small MLP, which includes two LayerNorms to stabilize training.

Transolver. The Transolver architecture [48] starts by encoding node coordinates using sinusoidal position embeddings, followed by an MLP that produces initial feature vectors. A learned mapping then assigns each node to a slice, enabling attention operations both within slices and between them. The processed features are passed through an MLP readout to generate the final field outputs. Two conditioning mechanisms are available: concatenating the conditioning vector with input features or applying it via DiT-based modulation across the network. Conditioning is done with the dit-based modulation [32]. Where a latent dimension of 128, a slice base of 32, and four attention layers are used. We additionally employ a larger model with 56, 128, and 8 layers for the more complex datasets.

B.2 Test-Time Adaptation Setup

In our experiments, we train baseline models for each dataset on 2 seeds, using the training pipeline from SIMSHIFT. We employ the small model variants for hot rolling, electric motor design, and sheet metal forming, and the large model variant for the more complex heatsink. For the TTA experiments, we utilize source test data to compute statistical information, then adapt and evaluate the models on target test data. The implementation of TTA follows the algorithmic framework provided in the SSA repository, with modifications described in Section 4. The adaptation process is limited to a single epoch, as in [3].

In our specific setup, task-dependent parameters, such as thickness or temperature, are encoded through a conditioner network. The conditioner is divided into two components: a main body and a final linear layer. We extract features from the main body's output and define the split between representation learner and predictor at this point—the conditioner serves as the representation learner, while Transolver acts as the predictor. This choice reflects the observation that most task-related parameter shifts occur within the conditioner [35]. For training SimRA we modify the weighted symmetric Kullback-Leibler divergence from Eq. (3) to account for Eq. (2) by removing $\boldsymbol{\alpha}_k$

$$\mathcal{L}_{\text{TTA}} = \frac{1}{2} \sum_{k=1}^{K} \left(\frac{(\tilde{\mu}_k^{\text{tgt}})^2 + \lambda_k^{\text{src}}}{\tilde{\sigma}_k^{\text{tgt}\,2}} + \frac{(\tilde{\mu}_k^{\text{tgt}})^2 + \tilde{\sigma}_k^{\text{tgt}\,2}}{\lambda_k^{\text{src}}} - 2 \right). \tag{4}$$

During testing, we adapt only the layer normalization [5] parameters of the conditioner, keeping all other model parameters fixed. To ensure robustness, we repeat each experiment with 20 different random seeds per model. This is particularly important since layer normalization is updated online, after every batch. All experiments are conducted with a fixed batch size of 32.

For comparison, we also report the best performing UDA algorithm as a lower bound. These models are trained according to the procedure outlined in SIMSHIFT [35]. We run the DeepCoral algorithm [38] with the provided λ ranges. After applying selection and emsembling strategies on top of the UDA algorithm, we showcase the best-performing model for each dataset. It is important to note that the UDA training process requires significantly more compute budget than the TTA approach: instead of requiring a single pre-trained model, UDA model selection relies on multiple models for robustness, each trained independently with different λ values.

C Additional results

To investigate the difference in effectiveness of TTA algorithms to different datasets, we analyze the features extracted by the representation learner. Specifically, the eigenvalue spectra of the corresponding covariance matrices are compared for each dataset. In Fig. 3, the eigenvalues for three out of four datasets (hot rolling, sheet metal forming, and heatsink design) show a similar decay: the first five eigenvalues already capture up to $\sim$60 % of the total variance. This is also confirmed by Table 2. In contrast, the electric motor dataset exhibits a much slower decay, suggesting that the variance is distributed across a larger number of components. This implies that electric motor requires a higher-dimensional representation to preserve the same level of information, whereas the other datasets are more efficiently represented in a low-dimensional manifold.

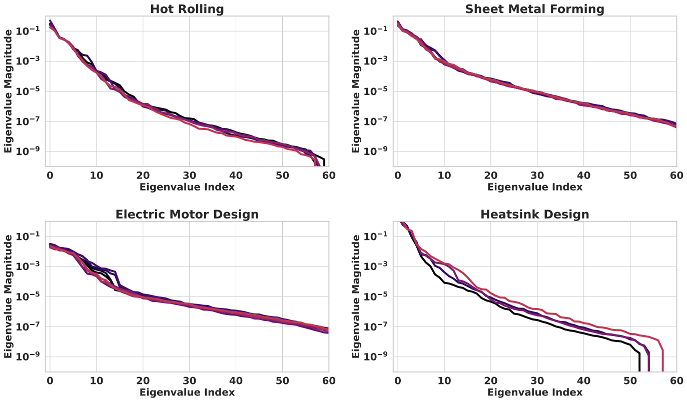


Figure 3: Eigenvalues analysis of *hot rolling*, *sheet metal forming*, *electric motor* and *heatsink design*, expressing the diverging decay throughout the datasets.

Table 2: Percentage of variance explained by the top 10 eigenvalues for each dataset.

Dataset	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9
Rolling	57.9 %	23.6 %	7.1 %	5.4 %	3.4 %	1.4 %	0.7 %	0.3 %	0.2 %	0.1 %
Forming	47.6 %	18.3 %	14.2 %	8.6 %	5.6 %	3.0 %	1.3 %	0.8 %	0.4 %	0.2 %
Motor	25.0 %	19.5 %	15.2 %	13.2 %	10.4 %	7.6 %	4.3 %	2.7 %	1.3 %	0.7 %
Heatsink	63.9 %	22.3 %	8.5 %	4.2 %	0.5 %	0.2 %	0.1 %	0.1 %	0.0 %	0.0 %