STABLE TEST-TIME ADAPTATION FOR HIGH-DIMENSIONAL SIMULATION REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine learning surrogates are increasingly used in engineering to accelerate costly simulations of partial differential equations (PDEs), yet distribution shifts between training and deployment often cause severe performance degradation, e.g., under unseen geometries or configurations. Test-Time Adaptation (TTA) can mitigate such shifts, but existing methods are largely developed for classification or image-based tasks on regular grids. When applied to high-dimensional regression on unstructured meshes, a standard setting in engineering simulation, estimating distributional statistics becomes severely ill-conditioned, leading to unstable adaptation. In this work, we propose a stable test-time adaptation method for high-dimensional simulation regressors. Our method uses E-optimal source subset selection to stabilize KL-based alignment between source and target representation distributions. The retained maximally informative source samples further enable stable source-risk regularization and parameter selection at test time. When applied to pretrained simulation surrogates, our method yields up to 12% improvement in out-of-distribution RMSE with modest computational overhead. To the best of our knowledge, this is the first systematic demonstration of effective TTA for high-dimensional simulation regression and generative design optimization, validated across diverse state-of-the-art benchmark datasets.

1 INTRODUCTION

Neural surrogates have become powerful tools for accelerating Partial Differential Equation (PDE) simulations across engineering and science. They perform well when test conditions match the training data, but performance often drops on unseen configurations (geometry, material types, structural dimensions, desired and physical parameters), i.e., when the data distribution shifts. This challenge gets more pronounced in industrial simulation and design optimization, where configurations can vary widely across iterations and frequently extend beyond the ranges at data generation and training time. In many cases, only large pretrained surrogate models are available, making full retraining costly or impractical. Additionally, access to the original training data is often restricted due to proprietary constraints or the dataset size. Together, these motivate the need for model- and task-agnostic approaches that enable stable online adaptation to shifted distributions without requiring costly simulations during design optimization.

The problem of tackling distribution shifts (Quinonero-Candela et al., 2008) is central to various long-standing research directions, such as domain adaptation (Ben-David et al., 2006), domain generalization (Blanchard et al., 2021), meta-learning (Hochreiter et al., 2001; Hospedales et al., 2022), and active learning (Settles, 2009). To counteract these shifts, a common goal is to adapt a model trained on a source distribution to a shifted target domain. For engineering tasks, where rapid adaptation is essential and target domain distributions are unavailable a priori, Test-Time Adaptation (TTA) is a particularly suitable approach, as it adapts models during inference with limited access to source data and with minimal computational overhead (Liang et al., 2020; Sun et al., 2020; Wang et al., 2021). TTA has proven effective across many application areas, such as medical imaging (Bateson et al., 2022), autonomous driving (Ni et al., 2024), and speech recognition (Lin et al., 2022).

While many current approaches tackle classification tasks (Liang et al., 2020; Wang et al., 2021), comparably little research can be found for high-dimensional regression settings (Liang et al., 2025).

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.

One interesting recent method tackling this gap is Significant-Subspace Alignment (SSA) (Adachi et al., 2025), capable of handling both classification and one-dimensional regression tasks. Further methods that are similar to the high-dimensional simulation tackle TTA in vision settings, for example depth-estimation (Liu et al., 2023), super-resolution (Park et al., 2020; Deng et al., 2023) and image dehazing (Liu et al., 2022).

However, these methods are tailored to the image domain, relying on inductive biases of regular grids and visual similarities between inputs and outputs. Engineering simulations differ substantially, since samples are represented as fields on unstructured non-Euclidean meshes with up to $\mathcal{O}(10^8)$ degrees of freedom and slight changes in low-dimensional inputs can induce complex changes in the solution fields (Ashton et al., 2026). Capturing these dynamics faithfully requires large surrogate models with high-dimensional latent representations. Since TTA does not allow for target labels, one cannot compute the target risk directly and optimize for it. However, it can be bounded by an importance-weighted source risk and a discrepancy between the latent source and target distributions (Nikzad-Langerodi et al., 2020). With this approach, three peculiarities arise: (i) high-dimensional latent spaces make latent distribution statistics ill-conditioned, leading to unstable alignment, (ii) adaptation requires selecting hyperparameters, yet the absence of target labels rules out standard supervised model selection, and (iii) only a small subset of source samples can be retained in our TTA setting, so both the source risk and model selection must rely on few samples.

To address these challenges, we introduce, to the best of our knowledge, the first TTA framework explicitly targeting neural surrogates for high-dimensional engineering tasks under distribution shifts, covering simulation (regression) and design optimization (generation) problems (see Figure 1). At the core of our approach lies the use of maximally informative source statistics to stabilize the adaptation process, which we approach by E-optimal sample selection (Kiefer, 1959; Pukelsheim, 2006). This lets us compress the source manifold into a small subset that stabilizes the KL-divergence alignment in the high-dimensional regime (i), while the few retained source samples enable importance-weighted model selection (ii) and source-risk estimation (iii).

We summarize our contributions as follows:

- *Problem:* We identify and frame the stability challenges of TTA for high-dimensional simulation regression problems.
- *Method:* We propose E-optimal Test Time Adaptation (ETA), a TTA method for high-dimensional simulation regressors that relies on E-optimal sample selection. We show that this effectively stabilizes feature alignment (Theorem 1). In addition, we employ an importance-weighted source-risk regularizer and model selection.
- *Performance:* We demonstrate (Figure 3, Tables 1 and 2) that our approach reliably outperforms standard TTA methods on diverse engineering problems caused by distribution shifts, using a structural mechanics dataset, the SIMSHIFT (Setinek et al., 2025) benchmark for high-dimensional simulation regression, and EngiBench (Felten et al., 2025) for design generation.

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., Bonnet et al. (2022); Toshev et al. (2023; 2024); Setinek et al. (2025). A particularly prominent line of work within neural surrogate modeling for PDEs is operator learning (Li et al., 2020; Kovachki et al., 2021; Lu et al., 2021; Alkin et al., 2024; Wu et al., 2024). 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 pretrained 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 can offer (nearly) free performance gains (Liang et al., 2025). While the majority of existing TTA methods have been developed for low-dimensional classification tasks (Liang et al., 2020; Yang et al., 2021; Niu et al.,

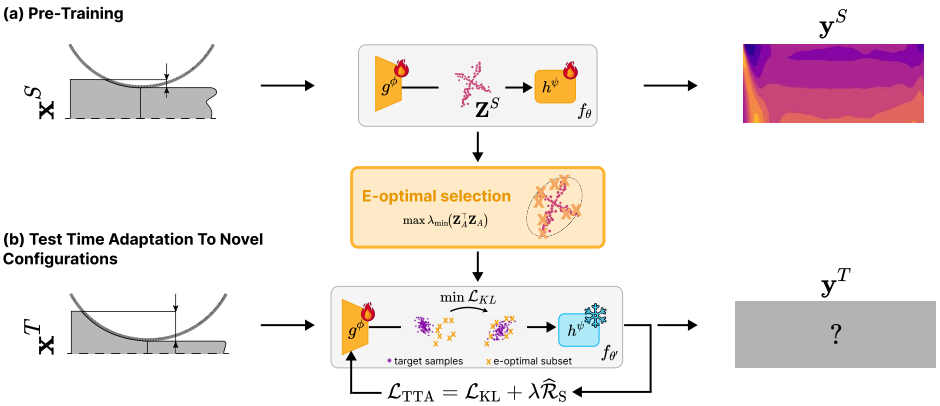

Figure 1: Our method applied to *hot rolling* task from Setinek et al. (2025). (a) Pre-training the surrogate f_θ on the source domain. E-optimal sample selection: extract latent representations for each source sample and select a representative subset $A^\star$. (b) Test-time adaptation of g^ϕ without full source data, using selected E-optimal statistics to adapt to a new target domain through feature alignment, source knowledge preservation, and parameter tuning.

2023; Sójka et al., 2023), employing methodologies such as entropy minimization (Wang et al., 2021; Zhou & Levine, 2021; Niu et al., 2022; Zhang et al., 2022; Zhao et al., 2023a) and feature alignment (Ishii & Sugiyama, 2021; Eastwood et al., 2021; Kojima et al., 2022; Adachi et al., 2023; Jung et al., 2023), recent works have begun to extend these ideas to image segmentation (He et al., 2021; Karani et al., 2021; Valanarasu et al., 2024). Research in TTA tackling regression problems is much sparser. Significant-Subspace Alignment (SSA) (Adachi et al., 2025) moves in this direction, showing positive results in the one-dimensional cases. Additionally, there are specialized methods designed for image regression tasks such as depth-estimation (Liu et al., 2023), super-resolution (Park et al., 2020; Deng et al., 2023), or image dehazing (Liu et al., 2022). Finally, TTA should not be confused with Test-Time Training (TTT), often used in time series literature (Sun et al., 2020; Wang et al., 2021; Sun et al., 2025). 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 (Muandet et al., 2013; Li et al., 2018; Holzleitner et al., 2024) and Unsupervised Domain Adaptation (UDA) (Gretton et al., 2006; Ganin et al., 2016; Sun & Saenko, 2016; Zellinger et al., 2017) can be effective in some scenarios, however their reliance on specific training, model selection and diverse training distributions limits their applicability. Meta-learning methods (Hochreiter et al., 2001; Finn et al., 2017) and active learning (Lewis & Gale, 1994; Musekamp et al., 2025) 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 can quickly adapt a pretrained model using 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 Setting

The goal of test-time adaptation (TTA) is to adapt a model pretrained on a source domain so that it achieves low prediction error on an unseen target domain using only unlabeled target data. Let $f^\theta : \mathcal{X} \to \mathbb{R}^d$ denote a regression model trained on a dataset of labeled source samples $\mathcal{D}_S := (\mathbf{x}_S^{(i)}, \mathbf{y}_S^{(i)})_{i=1}^{N_S} \subset \mathcal{X} \times \mathbb{R}^d$. The source samples are assumed to be independently and identically distributed (i.i.d.) according to an unknown joint distribution $P_S(X, Y)$. At test time, the model receives an unlabeled target dataset $\mathcal{D}_T := (\mathbf{x}_T^{(i)})_{i=1}^{N_T} \subset \mathcal{X}$, whose samples are assumed to be i.i.d. according to an unknown marginal distribution $P_T(X)$. The objective of TTA is to find model parameters θ that minimize the target risk

$$\mathcal{R}_T(\theta) := \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim P_T} \left[\|f^\theta(\mathbf{x}) - \mathbf{y}\|_2^2 \right]. \tag{1}$$

Since target labels are unavailable during test time, the target risk cannot be evaluated directly. Consequently, TTA methods optimize unsupervised surrogate objectives that depend only on the unlabeled target data together with limited information retained from the source domain. In our setting, this limited information is represented by m retained samples, where $m \ll N_S$.

Following recent feature-based TTA (Adachi et al., 2025), we decompose the pretrained regression model as $f^\theta = h^\psi \circ g^\phi$, $\theta = (\psi, \phi)$, where $g^\phi : \mathcal{X} \to \mathbb{R}^n$ denotes the feature extractor and $h^\psi : \mathbb{R}^n \to \mathbb{R}^d$ is the regression head. This adaptation strategy is based on the standard *covariate shift* assumption (Sugiyama et al., 2007), whereby the input marginal distributions differ across domains while the conditional distribution is preserved, i.e., $P_S(X) \neq P_T(X)$ and $P_S(Y \mid X) = P_T(Y \mid X)$. Accordingly, the regression head is kept fixed while only the feature extractor is adapted using unlabeled target samples.

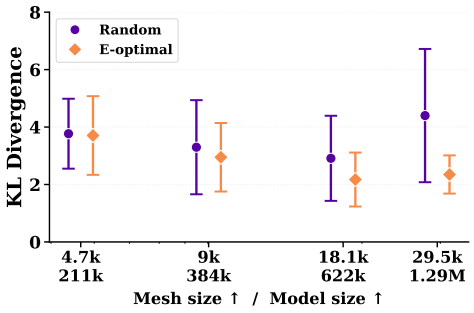

Figure 2: KL divergence with increasing mesh and model size. Variability increases for random sampling, while E-optimal sampling stabilizes the estimate.

TTA for neural simulation surrogates poses additional challenges, as discussed in Section 1, since adaptation requires estimating feature distributions from small unlabeled target batches in increasingly high-dimensional latent spaces. To quantify this, we fit diagonal Gaussians to the features ($\mathbf{z} = g^\phi(\mathbf{x})$) of a selected source subset (random or E-optimal) and to each target batch, and compute the symmetric KL-divergence between the two. Figure 2 underlines the problem, with increasing mesh and model size, the KL-divergence estimates become increasingly variable under random sampling, while E-optimal sampling stabilizes them.

4 Methodology

We propose E-optimal sample selection for stable KL divergence at test time (ETA), a method for stable test-time adaptation of neural simulation surrogates based on E-optimal source subset selection.

4.1 Target Risk Bound

Let $\mathcal{R}_S(\theta)$ and $\mathcal{R}_T(\theta)$ denote the source and target risks respectively (see Eq. (1)). Under the covariate shift assumption, standard domain adaptation theory relates the target risk to the source risk and a discrepancy between the source and target feature distributions. In particular, for approximately Gaussian latent distributions $\mathcal{N}(\mu^\phi, \Sigma^\phi)$, existing regression bounds yield

$$\mathcal{R}_T(\theta) \leq \mathcal{R}_S(\theta) + \sqrt{2\, D_{\mathrm{KL}}\Big(\mathcal{N}(\mu_S^\phi, \Sigma_S^\phi) \,\Big\|\, \mathcal{N}(\mu_T^\phi, \Sigma_T^\phi)\Big)} + \varepsilon, \tag{2}$$

where ε collects the remaining domain adaptation and Gaussian-approximation terms (Nikzad-Langerodi et al., 2020).

4.2 Surrogate for Target Risk Bound

Since the target risk is inaccessible during deployment, this bound motivates minimizing a computable surrogate consisting of a source risk term and a source–target distribution discrepancy. However, in our test-time adaptation setup, access to the full source dataset is not allowed. Instead, we store only a reasonably small subset consisting of m source samples. Such a subset $A \subseteq \mathcal{D}_S$, $|A| = m$ simultaneously enables estimation of source feature statistics, empirical source risk, and subsequent regularization and model selection during test-time adaptation. In particular, we approximate the source risk in Eq. (2) by the empirical risk $\widehat{\mathcal{R}}_A(\theta) = \frac{1}{m}\sum_{(\mathbf{x},\mathbf{y})\in A} \|f^\theta(\mathbf{x}) - \mathbf{y}\|_2^2$ over A and the distribution discrepancy by a symmetric Gaussian KL objective,

$$J_A(T;\phi) := \frac{1}{2}\left[D_{\mathrm{KL}}\Big(\mathcal{N}(\mu_A^\phi, S_A^\phi) \,\Big\|\, \mathcal{N}(\mu_T^\phi, S_T^\phi)\Big) + D_{\mathrm{KL}}\Big(\mathcal{N}(\mu_T^\phi, S_T^\phi) \,\Big\|\, \mathcal{N}(\mu_A^\phi, S_A^\phi)\Big) \right], \tag{3}$$

where $S := \Sigma + \rho I$ with $\rho > 0$. This yields the following estimate for the target-risk upper bound in Eq. (2):

$$\mathcal{L}_{\mathrm{TTA}}(A, T; \theta) := \widehat{\mathcal{R}}_A(\theta) + 2\sqrt{J_A(T; \phi)}, \tag{4}$$

which allows us to approximately minimize the target risk.

Our objective is to stably (w.r.t. sample variations in the target batch) minimize the surrogate upper bound in Eq. (4).

4.3 E-optimal Subset Selection for Stable Estimation of Target Risk Bound

We measure the distance between $\mathcal{N}(\mu_T, \Sigma_T)$ and $\mathcal{N}(\mu_{T'}, \Sigma_{T'})$ for two target batches $T, T' \in \mathcal{T}$ as

$$d_{\mathcal{T}}(T, T') := \|\mu_T - \mu_{T'}\|_2 + \|\Sigma_T - \Sigma_{T'}\|_*, \tag{5}$$

where $\|\cdot\|_*$ denotes the trace norm, i.e., the sum of the singular values.

Definition 1 (stability)**.** *The objective $\mathcal{L}_{\mathrm{TTA}}(A, \cdot\,; \theta)$ is called stable if*

$$\lim_{k\to\infty} \mathcal{L}_{\mathrm{TTA}}(A, T_k; \theta) = \mathcal{L}_{\mathrm{TTA}}(A, T; \theta) \tag{6}$$

for any sequence $(T_k)_{k=1}^{\infty}$ with $\lim_{k\to\infty} d_{\mathcal{T}}(T_k, T) = 0$.

A sufficient condition for stability is a Lipschitz bound on J_A with respect to $d_{\mathcal{T}}$.[1] We therefore seek the representative subset A that minimizes this upper bound. Let $\lambda_A := \lambda_{\min}(S_A)$ the smallest eigenvalue of the regularized retained source covariance matrix S_A.

Theorem 1 (Stability via E-optimality)**.** *Assume that there exist constants $R, M < \infty$ such that $\|\mu_T - \mu_A\|_2 \le R$, $\|S_A\|_* \le M$ for every $A \in \mathcal{A}_m$ and $T \in \mathcal{T}$. Then,*

$$|\mathcal{L}_{\mathrm{TTA}}(A, T; \theta) - \mathcal{L}_{\mathrm{TTA}}(A, T'; \theta)| \le 2\sqrt{C_A\, d_{\mathcal{T}}(T, T')}, \tag{7}$$

where

$$C_A := \max\left\{ \frac{1}{4}\left(\frac{1}{\lambda_A} + \frac{M + R^2}{\rho^2}\right),\ \frac{R}{2}\left(\frac{1}{\lambda_A} + \frac{1}{\rho}\right) \right\}.$$

The proof is provided in the Appendix B. Theorem 1 shows that $\mathcal{L}_{\mathrm{TTA}}$ is stable, where the underlying Lipschitz bound on J_A is controlled by the smallest eigenvalue of the regularized retained source covariance matrix.

4.4 E-optimal Test Time Adaptation (ETA)

Motivated by our Theorem 1, we seek to find a subset $A^\star \in \arg\max_{A\in\mathcal{A}_m} \lambda_A$, where $\mathcal{A}_m$ denotes all admissible subsets of size m. Such a subset $A^\star$ is called *E-optimal* subset. We implement E-optimal subset selection using greedy forward selection (Algorithm 1), with a rank-aware PCA and pivoted-QR initialization detailed in Appendix C. The subset $A^\star$ is used in the adaptation at three points: feature alignment, source knowledge preservation, and parameter tuning, reformulating each to fit the high-dimensional regression problem setting.

Feature alignment is used to reduce the dissimilarity between source and target distributions (see Section 2). Following a recent TTA method for regression, SSA (Adachi et al., 2025), latent directions that strongly influence the prediction are the right candidates for alignment. In SSA, feature importance is defined for one-dimensional regression through manually selecting a significant subspace. We extend the idea by assigning a *positive importance weight* to every principal direction,

$$\alpha^{(k)} = 1 + \left\| \mathbf{W}\mathbf{v}_S^{(k)} \right\|_2, \tag{8}$$

where $\mathbf{v}_S^{(k)}$ denotes the k-th source principal direction and $W \in \mathbb{R}^{d\times n}$ is the stored weight of the architecture-specific linear layer at the selected feature split (Appendix C). At deployment, mean-centered target features $\mathbf{z}_T = g^\phi(\mathbf{x}_T)$ are projected onto the source principal components $\mathbf{V}_S$, and

[1] A function $f : \mathcal{X} \to \mathbb{R}$ is L-Lipschitz with respect to a metric $d : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ if $|f(x) - f(x')| \le L\, d(x, x')$ for all $x, x' \in \mathcal{X}$. The smallest admissible Lipschitz constant is $L = \sup_{x\neq x'} \frac{|f(x)-f(x')|}{d(x,x')}$.

Algorithm 1 Greedy E-Optimal sample selection.

Require: Source inputs x_S, feature mapping g^ϕ, subset size m, and regularization parameter $\rho > 0$
Ensure: Selected indices $\mathcal{I}$
$\mathbf{Z} \leftarrow g^\phi(x_S)$
$\mathbf{Z} \leftarrow Z - \text{mean}(Z, \text{axis} = 0)$
$\mathcal{I} \leftarrow \emptyset; \mathcal{U} \leftarrow \{1, \ldots, N_S\}$
$i^\star \leftarrow \arg\max_{i \in \mathcal{U}} \|\mathbf{Z}^{(i)}\|_2$
$\mathcal{I} \leftarrow \mathcal{I} \cup \{i^\star\}; \mathcal{U} \leftarrow \mathcal{U} \setminus \{i^\star\}$
while $|\mathcal{I}| < m$ **do**
  $G \leftarrow (\mathbf{Z}^{(\mathcal{I})})^\top \mathbf{Z}^{(\mathcal{I})} + \rho \mathbf{I}$
  $i^\star \leftarrow \arg\max_{i \in \mathcal{U}} \lambda_{\min}\big(G + \mathbf{Z}^{(i)} \mathbf{Z}^{(i)\top}\big)$
  $\mathcal{I} \leftarrow \mathcal{I} \cup \{i^\star\}; \mathcal{U} \leftarrow \mathcal{U} \setminus \{i^\star\}$
end while
return $\mathcal{I}$

reweighted by $\boldsymbol{\alpha}$. Finally, source and target feature distributions are aligned by minimizing the symmetric KL-divergence $J_{\mathrm{A}^\star}$.

Source knowledge preservation is realized by regularization on the subsampled source statistics, seen in Eq. (4). We weigh the added source risk according to the dataset sizes of the selected source and target samples. This anchors the model's source performance, preventing the feature alignment updates driven by $J_{\mathrm{A}^\star}$ from degrading its predictive accuracy.

Parameter tuning We integrate Importance Weighted Validation (IWV) (Shimodaira, 2000) using E-optimal samples to tune the test-time adaptation learning rate. Since the target risk in Eq. (1) cannot be computed directly without access to target labels, we estimate it via an importance-weighted source risk under the covariate shift assumption $p_S(\mathbf{y} \mid \mathbf{x}) = p_{\mathrm{T}}(\mathbf{y} \mid \mathbf{x})$:

$$\widehat{\mathcal{R}}_T(f_\theta) = \frac{1}{m} \sum_{i=1}^{m} \beta(\mathbf{z}^{(i)}) \, \|f_\theta(\mathbf{x}_S^{(i)}) - \mathbf{y}_S^{(i)}\|_2^2, \tag{9}$$

where $(\mathbf{x}_S^{(i)}, \mathbf{y}_S^{(i)})_{i=1}^m$ denotes the set of E-optimal source samples and $\beta(\mathbf{z}) = p_T(\mathbf{z})/p_S(\mathbf{z})$ is the density ratio. We estimate this ratio in latent space $\mathbf{z} = g^\phi(\mathbf{x})$ under the Gaussian data generating assumption. Using the estimate in Eq. (9), we perform model selection via line search over the TTA learning rate, evaluating performance after each adaptation step and stopping once further updates no longer improve the objective.

5 Experiments

This section presents an empirical analysis of ETA for Simulation. We introduce the benchmarks used for evaluation and demonstrate the performance of our method in complex industrial use cases, where neural surrogates are employed to approximate costly numerical simulations or to directly generate candidate designs. In all experiments, the parameters of the E-optimal algorithm (see Algorithm 1) are fixed with $m = 8$ indices.

5.1 Setup

Our evaluation is conducted on seven benchmark datasets from three sources: our own Plate 2D dataset, four industrial simulation tasks from the SIMSHIFT benchmark (Setinek et al., 2025), and two design optimization datasets from EngiBench (Felten et al., 2025). Plate 2D is a synthetic simulation dataset for static linear elasticity at varying mesh resolutions. SIMSHIFT evaluates how surrogate models adapt to distribution shifts on a range of real-world industrial simulation tasks, while EngiBench provides design optimization datasets, optimizers, and simulators for evaluating designs. For all datasets, the inputs $\mathbf{x}$ represent parameters like geometry, material properties, desired or operating conditions. The "labels" $\mathbf{y}$ correspond to high-dimensional fields such as stresses or deformation for SIMSHIFT and Plate 2D, and material density of the generated design for EngiBench.

The target distributions are generated from unseen parameter configurations, and the goal is to predict the corresponding fields outside the training regime. While Plate 2D and SIMSHIFT formulate the problem as a regression task with neural operators (Kovachki et al., 2021), EngiBench treats it as an inverse problem solved by generative models.

Following the descriptive methodology, we collect m E-optimal source samples prior to model adaptation. These samples are used as a fixed source reference throughout our TTA framework. Additionally, the target dataset is split by value ranges of the distribution-shift parameter to construct adaptation batches (for details on the corresponding parameters for each dataset, see Appendix C). The diversity in task formulation and training paradigm across the benchmarks highlights the model-agnostic nature of our method.

5.2 Stable adaptation in high-dimension: Plate

We first evaluate adaptation potential on a synthetic 2D plate dataset for solid mechanics, simulating static linear elasticity under mixed Dirichlet–Neumann boundary conditions across varying mesh resolutions. We define the source and target domain split, as described in Setinek et al. (2025), where we train models on the full datasets and analyze the t-SNE visualizations of the latent feature spaces as the input conditions are varied. A detailed analysis of this procedure and corresponding visualizations are provided in E. Figure 3 compares ETA against the unadapted source model (*Source*) and the TTA regression baseline (SSA) over increasing mesh and model size (full table in Section F).

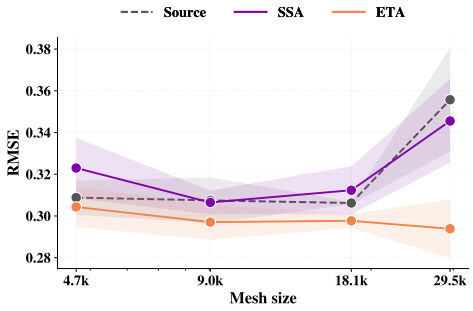

Figure 3: RMSE across mesh resolutions (log-scaled average node count) for Source, SSA, and ETA, including standard deviation.

ETA consistently improves over the source model across all model and mesh sizes. More importantly, its performance remains stable as the mesh size increases. Even at the largest mesh, where the source and SSA errors increase substantially, ETA maintains a low RMSE. SSA, in contrast, fails to deliver comparable gains over the source baseline.

5.3 Neural Surrogates for Simulation: SIMSHIFT

SIMSHIFT (Setinek et al., 2025) spans 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 parameters such as initial conditions, material or geometry specifications that were used to generate the samples. Shifts happen in parametric space, as opposed to unstructured variations occurring in images. We perform all our experiments using the medium difficulty domain shift setup for all datasets. For a detailed description of the datasets, their creation, and the defined distribution shifts, we refer the reader to the SIMSHIFT publication (Setinek et al., 2025).

Table 1 summarizes the results across all datasets, comparing our method with SSA and Tent as established TTA baselines, as well as the unadapted source model (*Source*). Implementation details are provided in Sections C and D in the Appendix.

Across all settings, ETA consistently outperforms SSA and yields the strongest performance among all adaptation methods, establishing a new baseline for test-time adaptation in neural surrogate regression. While in some regimes the absolute gains over the source model are modest, they are achieved without sacrificing stability. In contrast, both SSA and, more prominently, Tent frequently degrade performance relative to the pretrained model, indicating a lack of robustness to these high-dimensional distribution shifts.

To improve the interpretability of our results, we quantify the discrepancy between the source and target domains directly in the output space. Namely, the potential transfer error is upper bounded by the $\mathcal{H}$-divergence which itself is upper bounded by the Proxy $\mathcal{A}$-distance (PAD) (for details see Johansson et al. (2019), Bouvier et al. (2020) and Zellinger et al. (2021)). Heatsink and Rolling hold the largest PAD values (1.767 and 1.765) and correspondingly show the largest performance gains,

Table 1: Comparison of baselines vs. TTA methods across all SIMSHIFT simulation datasets, averaged over batches and four pretrained models (± std). RMSE is field-normalized. Tent is unstable: * excludes exploding seeds, ** indicates no stable runs.

(a) Rolling

Model	RMSE (↓)	MAE (↓)	R^2 (↑)
Source	$1.139_{\pm 0.001}$	$1.005_{\pm 0.001}$	$0.563_{\pm 0.001}$
Tent	$2.426_{\pm 0.019}$	$2.099_{\pm 0.019}$	$-0.408_{\pm 0.010}$
SSA	$1.060_{\pm 0.020}$	$0.950_{\pm 0.018}$	$0.610_{\pm 0.010}$
ETA	$\mathbf{1.004_{\pm 0.007}}$	$\mathbf{0.883_{\pm 0.008}}$	$\mathbf{0.643_{\pm 0.002}}$

(b) Motor

Model	RMSE (↓)	MAE (↓)	R^2 (↑)
Source	$0.056_{\pm 0.004}$	$0.029_{\pm 0.002}$	$0.995_{\pm 0.001}$
Tent*	$0.886_{\pm 0.026}$	$0.568_{\pm 0.023}$	$0.073_{\pm 0.015}$
SSA	$0.121_{\pm 0.004}$	$0.071_{\pm 0.002}$	$0.980_{\pm 0.002}$
ETA	$\mathbf{0.053_{\pm 0.003}}$	$\mathbf{0.028_{\pm 0.001}}$	$\mathbf{0.996_{\pm 0.001}}$

(c) Forming

Model	RMSE (↓)	MAE (↓)	R^2 (↑)
Source	$0.072_{\pm 0.003}$	$0.031_{\pm 0.001}$	$0.996_{\pm 0.000}$
Tent*	$1.414_{\pm 0.002}$	$0.705_{\pm 0.001}$	$-0.066_{\pm 0.001}$
SSA	$0.182_{\pm 0.031}$	$0.083_{\pm 0.012}$	$0.976_{\pm 0.008}$
ETA	$\mathbf{0.068_{\pm 0.003}}$	$\mathbf{0.029_{\pm 0.002}}$	$\mathbf{0.996_{\pm 0.000}}$

(d) Heatsink

Model	RMSE (↓)	MAE (↓)	R^2 (↑)
Source	$0.637_{\pm 0.001}$	$0.424_{\pm 0.001}$	$0.479_{\pm 0.001}$
Tent	**	**	**
SSA	$0.661_{\pm 0.004}$	$0.402_{\pm 0.003}$	$0.458_{\pm 0.001}$
ETA	$\mathbf{0.615_{\pm 0.001}}$	$\mathbf{0.396_{\pm 0.002}}$	$0.469_{\pm 0.001}$

whereas Motor exhibits a significantly smaller PAD value (0.314), indicating a comparatively weak output-space shift (values taken from Table 2 in Setinek et al. (2025)). This is consistent with the moderate performance improvement observed for Motor.

5.4 Generative Design Optimization: EngiBench

Table 2: Comparison of current baselines with TTA methods for design optimization datasets. Reported results averaged over batches for two pretrained models (± std).

(a) Beams2D

Model	COMP (↓)	MAE (↓)	MMD (↓)
Source	$114.7_{\pm 1.854}$	$\mathbf{0.020_{\pm 0.003}}$	$\mathbf{0.036_{\pm 0.002}}$
SSA	$121.7_{\pm 2.326}$	$0.034_{\pm 0.004}$	$0.039_{\pm 0.003}$
ETA	$\mathbf{112.3_{\pm 1.290}}$	$0.025_{\pm 0.004}$	$0.037_{\pm 0.002}$

(b) HeatConduction2D

Model	COMP ($\times 10^{-3}$)	MAE (↓)	MMD (↓)
Source	$0.577_{\pm 0.561}$	$0.336_{\pm 0.057}$	$0.095_{\pm 0.000}$
SSA	$0.712_{\pm 0.615}$	$0.349_{\pm 0.057}$	$0.095_{\pm 0.000}$
ETA	$\mathbf{0.537_{\pm 0.491}}$	$\mathbf{0.334_{\pm 0.015}}$	$0.095_{\pm 0.000}$

We evaluate on two EngiBench design optimization tasks: *structural beam bending* and *2D heat conduction*. By default, these datasets do not include predefined source and target domains. As already described for the Plate 2D dataset, we follow the specified approach in Setinek et al. (2025). A detailed analysis of this process and corresponding figures can be found in Section E.

We report Mean Absolute Error (MAE), the Maximum Mean Discrepancy (MMD), and Compliance (COMP), a dataset-specific objective value calculated with a Finite Element Method (FEM) solver. For Beams2D, compliance is the inverse of stiffness, whereas for HeatConduction2D, it is the thermal compliance coefficient. All performance metrics reflect only feasible design solutions, as we exclude structural failure cases for analysis.

In Table 2, we compare our method against the *"unregularized"* pretrained model (*Source*) and SSA. Across both tasks, our approach typically matches or reduces errors relative to the unregularized model. Compared to our method, SSA shows unstable behavior on certain metrics, sometimes even deteriorating performance substantially. Such behavior is highly undesirable in TTA deployments and underlines the strong suit of our approach: its stability.

5.5 Ablations

Component Analysis We evaluate the contribution of E-optimal source selection and importance weighting by an incremental ablation study. Starting from the existing alignment strategy SSA, we isolate the effect of E-optimal source importance weighting from source selection. For both the SSA baseline and the source importance weighting, we use the original parame-

Table 3: RMSE scores of ETA with and without importance weighting and model selection. The first two rows were evaluated with $lr = 0.01$. Best scores are bolded.

$\hat{\mathcal{R}}_S(f_\theta)$	IWV	Rolling	Motor	Forming
		$1.060_{\pm 0.020}$	$0.121_{\pm 0.004}$	$0.182_{\pm 0.031}$
✓		$1.004_{\pm 0.008}$	$0.077_{\pm 0.030}$	$0.154_{\pm 0.037}$
	✓	$1.006_{\pm 0.010}$	$0.055_{\pm 0.012}$	$0.072_{\pm 0.012}$
✓	✓	$\mathbf{1.004_{\pm 0.007}}$	$\mathbf{0.053_{\pm 0.003}}$	$\mathbf{0.068_{\pm 0.003}}$
Source		$1.139_{\pm 0.001}$	$0.056_{\pm 0.004}$	$0.072_{\pm 0.003}$

ter values ($lr = 0.01$) from Adachi et al. (2025). Results on the SIMSHIFT benchmark in Table 3 show that each incremental addition improves performance over the previous configuration.

Random vs E-optimal samples To assess whether our selection strategy provides a meaningful signal, we compare E-optimal selection with random sample selection on the *SIMSHIFT* and *Plate 2D* datasets. E-optimal selection outperforms other sampling strategies in all datasets. Even though the results shown are only marginal, they guarantee stability with low variance, contrary to random selection of samples. Detailed table overview is shown in the Appendix (Section F).

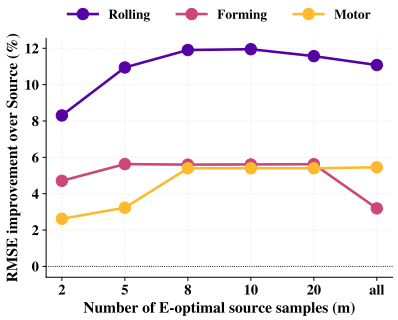

Figure 4: RMSE improvement vs. m E-optimal samples

m-Samples We show in Figure 4 that increasing the number of E-optimal samples improves performance only up to a specific sample size across the tested datasets. Using the complete source dataset can lead to a performance drop compared with the compact E-optimal subset. This indicates that in the high-dimensional simulation problem setting, successful adaptation depends on the quality rather than the quantity of the source samples.

Compute and Memory Compared to SSA, our method adds moderate computational overhead. On hot rolling, ETA increases runtime by approximately $1.34\times$, mainly due to forwarding the selected source samples and computing the source regularization term (see Table 4). The overhead depends on the number of source samples m, but remains moderate for the values used in our experiments (see Table 11). Our proposed learning rate sweeps can be executed in parallel and therefore do not add significant runtime overhead. Additionally, memory overhead is small: at $m = 8$, the additional memory amounts to less than 1% of full memory on most datasets and at most 2.6%.

Table 4: Runtime comparison on *hot rolling*. Mean $\pm$ std across 10 runs.

Method	Runtime (s)	Increase
SSA	$2.612_{\pm 1.748}$	-
ETA	$3.493_{\pm 0.624}$	0.881 ($\uparrow$ $1.34\times$)

6 Conclusion and Future Work

In this work, we take an initial step toward reliable test-time adaptation for neural surrogates and, more broadly, for high-dimensional multivariate regression. Our main methodological contribution is the use of E-optimal statistics within a unified framework to stabilize test-time adaptation. The proposed adjustments enable TTA to achieve consistent zero-shot performance improvements at moderate computational cost.

Limitations and future directions Despite the outlined gains, some results show only marginal improvements, a behavior commonly observed in TTA methods (Zhao et al., 2023b; Sójka et al., 2023). Additionally, our approach only evaluates on single-step distribution shifts, leaving open how the method behaves under non-stationary simulation tasks, where errors could accumulate. To mitigate some of the outlined limitations, we foresee two directions: incorporating physics-informed constraints and priors (Raissi et al., 2019; Cai et al., 2021), tailored and calibrated to the test cases, and uncertainty quantification to localize failure regions in fields where adaptation is necessary.

AI USE STATEMENT

In this work, we used generative AI tools for three tasks requiring disclosure: providing critical ingredients for proving mathematical claims, assisting in the writing of the proof, and implementing methods. Specifically, we used ChatGPT and Claude to suggest candidate proof steps and lemma for the results in Section 4; we checked each suggested step line by line, and wrote and verified the final proof as it appears in the paper. The authors formulated the mathematical claims and theorem statements themselves, not an LLM. We additionally used Codex to implement the baseline methods reported in Section 5; these implementations follow the specifications given in the original papers, and we reviewed the resulting code against those specifications and verified usability before using it. We implemented our proposed method. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or give feedback on our research methodology or experiments, clean or reformat datasets, or interpret results. Translation assistance and support for qualitative or thematic data analysis are not applicable to this work.

Additionally, we used generative AI tools for several tasks in the recommended-disclosure category: editing the manuscript to improve readability and clarity, obtaining feedback on the draft text and on how we position our contribution relative to related work, writing the code that produces the figures in this paper (including suggestions on chart type and styling), and general software engineering tasks such as refactoring and debugging. The research question, the overall approach, and the structure of the paper are entirely the authors' own and were not developed with AI assistance. We have reviewed all AI-assisted work: all AI-generated code was inspected, run, and tested by the authors, and every figure was checked against the underlying experimental data to confirm it is an accurate representation of our results; all AI-edited text was reviewed by the authors for factual accuracy and for unintentional overlap with existing work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Experimental details for all benchmark datasets, including the source/target splits and model architectures, are provided in Appendix Sections D and E. The full training pipeline, along with all chosen hyperparameters, is described in Appendix Sections C and D. The complete proof of Theorem 1 is provided in Appendix Section B. Code will be released upon publication.

REFERENCES

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.

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.

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.

Martin S. Alnæs, Anders Logg, Kristian B. Ølgaard, Marie E. Rognes, and Garth N. Wells. Unified form language: A domain-specific language for weak formulations of partial differential equations. *ACM Trans. Math. Softw.*, 40(2), March 2014. ISSN 0098-3500. doi: 10.1145/2566630. URL https://doi.org/10.1145/2566630.

Neil Ashton, Adam Clark, Liam Heidt, Christopher Ivey, Sanjeeb Bose, Rahul Agrawal, Konrad Goc, Rishi Ranade, Corey Adams, Peter Sharpe, Sheel Nidhan, Semit Akkurt, Daniel V. Leibovici, and Jean Kossaifi. Hiliftaeroml: High-fidelity computational fluid dynamics dataset for high-lift aircraft aerodynamics. *CoRR*, abs/2605.19565, 2026. doi: 10.48550/ARXIV.2605.19565. URL https://doi.org/10.48550/arXiv.2605.19565.

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

Igor A. Baratta, Joseph P. Dean, Jørgen S. Dokken, Michal Habera, Jack S. Hale, Chris N. Richardson, Marie E. Rognes, Matthew W. Scroggs, Nathan Sime, and Garth N. Wells. Dolfinx: The next generation fenics problem solving environment, December 2025. URL https://doi.org/10.5281/zenodo.18101307.

Mathilde Bateson, Herve Lombaert, and Ismail Ben Ayed. Test-time adaptation with shape moments for image segmentation. In Linwei Wang, Qi Dou, P. Thomas Fletcher, Stefanie Speidel, and Shuo Li (eds.), *Medical Image Computing and Computer Assisted Intervention - MICCAI 2022 - 25th International Conference, Singapore, September 18-22, 2022, Proceedings, Part IV*, volume 13434 of *Lecture Notes in Computer Science*, pp. 736–745. Springer, 2022. doi: 10.1007/978-3-031-16440-8_70. URL https://doi.org/10.1007/978-3-031-16440-8_70.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

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.

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.

Victor Bouvier, Philippe Very, Clément Chastagnol, Myriam Tami, and Céline Hudelot. Robust domain adaptation: Representations, weights and inductive bias. In Frank Hutter, Kristian Kersting, Jefrey Lijffijt, and Isabel Valera (eds.), *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020, Proceedings, Part I*, volume 12457 of *Lecture Notes in Computer Science*, pp. 353–377. Springer, 2020. doi: 10.1007/978-3-030-67658-2_21. URL https://doi.org/10.1007/978-3-030-67658-2_21.

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.

Zeshuai Deng, Zhuokun Chen, Shuaicheng Niu, Thomas H. Li, Bohan Zhuang, and Mingkui Tan. Efficient test-time adaptation for super-resolution with second-order degradation and reconstruction. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

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.

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.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 06–11 Aug 2017.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. doi: https://doi.org/10.1002/nme.2579. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2579.

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.

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.

Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *International conference on artificial neural networks*, pp. 87–94. Springer, 2001.

Markus Holzleitner, Sergei V Pereverzyev, and Werner Zellinger. Domain generalization by functional regression. *Numerical Functional Analysis and Optimization*, 45(3):259–281, 2024.

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, 2022.

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

Fredrik D Johansson, David Sontag, and Rajesh Ranganath. Support and invertibility in domain-invariant representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 527–536. PMLR, 2019.

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.

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.

J. Kiefer. Optimum experimental designs. *Journal of the Royal Statistical Society: Series B (Methodological)*, 21(2):272–304, 1959. doi: https://doi.org/10.1111/j.2517-6161.1959.tb00338.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1959.tb00338.x.

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.

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.

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

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

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.

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.

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, 2025. ISSN 1573-1405. doi: 10.1007/s11263-024-02181-w. URL http://dx.doi.org/10.1007/s11263-024-02181-w.

Guan-Ting Lin, Shang-Wen Li, and Hung-yi Lee. Listen, adapt, better WER: source-free single-utterance test-time adaptation for automatic speech recognition. In Hanseok Ko and John H. L. Hansen (eds.), *23rd Annual Conference of the International Speech Communication Association, Interspeech 2022, Incheon, Korea, September 18-22, 2022*, pp. 2198–2202. ISCA, 2022. doi: 10.21437/INTERSPEECH.2022-600. URL https://doi.org/10.21437/Interspeech.2022-600.

Huan Liu, Zijun Wu, Liangyan Li, Sadaf Salehkalaibar, Jun Chen, and Keyan Wang. Towards multi-domain single image dehazing via test-time training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 5821–5830. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00574. URL https://doi.org/10.1109/CVPR52688.2022.00574.

Huan Liu, Zhixiang Chi, Yuanhao Yu, Yang Wang, Jun Chen, and Jin Tang. Meta-auxiliary learning for future depth prediction in videos. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, pp. 5745–5754. IEEE, 2023. doi: 10.1109/WACV56688.2023.00571. URL https://doi.org/10.1109/WACV56688.2023.00571.

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.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International conference on machine learning*, pp. 10–18. PMLR, 2013.

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.

Jiayi Ni, Senqiao Yang, Ran Xu, Jiaming Liu, Xiaoqi Li, Wenyu Jiao, Zehui Chen, Yi Liu, and Shanghang Zhang. Distribution-aware continual test-time adaptation for semantic segmentation. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pp. 3044–3050. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10610045. URL https://doi.org/10.1109/ICRA57147.2024.10610045.

Ramin Nikzad-Langerodi, Werner Zellinger, Susanne Saminger-Platz, and Bernhard A. Moser. Domain adaptation for regression under beer–lambert's law. *Knowledge-Based Systems*, 210: 106447, 2020. doi: 10.1016/j.knosys.2020.106447.

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.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Seobin Park, Jinsu Yoo, Donghyeon Cho, Jiwon Kim, and Tae Hyun Kim. Fast adaptation to super-resolution networks via meta-learning. In *European conference on computer vision*, pp. 754–769. Springer, 2020.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023.

Friedrich Pukelsheim. *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics, 2006. doi: 10.1137/1.9780898719109. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898719109.

Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset Shift in Machine Learning*. MIT Press, 2008.

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.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells III, and Alejandro F. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, volume 9351 of *Lecture Notes in Computer Science*, pp. 234–241. Springer, 2015. doi: 10.1007/978-3-319-24574-4_28. URL https://doi.org/10.1007/978-3-319-24574-4_28.

Matthew W. Scroggs, Igor A. Baratta, Chris N. Richardson, and Garth N. Wells. Basix: a runtime finite element basis evaluation library. *Journal of Open Source Software*, 7(73):3982, 2022a. doi: 10.21105/joss.03982. URL https://doi.org/10.21105/joss.03982.

Matthew W. Scroggs, Jørgen S. Dokken, Chris N. Richardson, and Garth N. Wells. Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes. *ACM Trans. Math. Softw.*, 48(2), May 2022b. ISSN 0098-3500. doi: 10.1145/3524456. URL https://doi.org/10.1145/3524456.

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.

Burr Settles. Active learning literature survey. 2009.

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.

Damian Sójka, Sebastian Cygert, Bartłomiej Twardowski, and Tomasz Trzciński. Ar-tta: A simple method for real-world continual test-time adaptation. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 3483–3487. IEEE, 2023.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(35):985–1005, 2007. URL http://jmlr.org/papers/v8/sugiyama07a.html.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision (ECCV) Workshops*, pp. 443–450. Springer, 2016.

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.

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.

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.

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.

Jeya Maria Jose Valanarasu, Pengfei Guo, Vibashan VS, and Vishal M Patel. On-the-fly test-time adaptation for medical image segmentation. In *Medical Imaging with Deep Learning*, pp. 586–598. PMLR, 2024.

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.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

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.

Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *International Conference on Learning Representations (ICLR)*, 2017.

Werner Zellinger, Natalia Shepeleva, Marius-Constantin Dinu, Hamid Eghbal-zadeh, Hoan Duc Nguyen, Bernhard Nessler, Sergei V. Pereverzyev, and Bernhard Alois Moser. The balancing principle for parameter choice in distance-regularized domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 20798–20811, 2021.

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.

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

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

Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. On pitfalls of test-time adaptation. In *Proceedings of the 40th International Conference on Machine Learning*, 2023b.

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 (SSA) is a TTA method for one-dimensional regression (Adachi et al., 2025). It consists of two steps: *feature alignment* and *significant-subspace alignment*. First, source statistics such as the mean $\boldsymbol{\mu}_S$ and covariance $\boldsymbol{\Sigma}_S$ are computed after source training. A significant subspace is then constructed by selecting the K leading eigenpairs of $\boldsymbol{\Sigma}_S$. Each principal direction $\mathbf{v}_S^{(k)}$ is weighted according to its influence on the regression output:

$$\alpha^{(k)} = 1 + \left| \mathbf{w}^\top \mathbf{v}_S^{(k)} \right|,$$

where $\alpha^{(k)} \geq 1$ emphasizes directions that strongly affect the regression output.

At test time, the target features are projected onto the selected source principal directions. The mean and variance of the projected target features, $\tilde{\mu}_T^{(k)}$ and $(\tilde{\sigma}_T^{(k)})^2$, are then aligned with the corresponding source statistics 0 and $\lambda_S^{(k)}$. The resulting adaptation objective is a weighted symmetric Kullback–Leibler divergence.

B Proof of Theorem 1

Theorem 1 (Extended version) For every $A \in \mathcal{A}_m$ and $T \in \mathcal{T}$, define the regularized covariance matrices $S_A := \Sigma_A + \rho I$, $S_T := \Sigma_T + \rho I$, $\rho > 0$. Let $\lambda_A := \lambda_{\min}(S_A)$. Assume that there exist constants $R, M < \infty$ such that

$$\|\mu_T - \mu_A\|_2 \leq R, \qquad \|S_A\|_* \leq M \qquad \text{for every } A \in \mathcal{A}_m \text{ and } T \in \mathcal{T}. \tag{10}$$

Then,

$$|\mathcal{L}_{\mathrm{TTA}}(A, T; \theta) - \mathcal{L}_{\mathrm{TTA}}(A, T'; \theta)| \leq 2\sqrt{C_A \, d_{\mathcal{T}}(T, T')}, \tag{11}$$

where

$$C_A := \max\left\{ \frac{1}{4}\left(\frac{1}{\lambda_A} + \frac{M + R^2}{\rho^2} \right),\ \frac{R}{2}\left(\frac{1}{\lambda_A} + \frac{1}{\rho} \right) \right\}.$$

.

Consequently, $\mathcal{L}_{\mathrm{TTA}}(A, \cdot; \theta)$ is stable in the sense of Eq. (6).

Proof. For fixed A and θ, $\widehat{R}_A(\theta)$ is independent of the target batch. Hence,

$$|\mathcal{L}_{\mathrm{TTA}}(A, T; \theta) - \mathcal{L}_{\mathrm{TTA}}(A, T'; \theta)| = 2\left| \sqrt{J_A(T; \phi)} - \sqrt{J_A(T'; \phi)} \right| \tag{12}$$

$$\leq 2\sqrt{|J_A(T; \phi) - J_A(T'; \phi)|}. \tag{13}$$

Therefore, it suffices to establish the Lipschitz bound

$$|J_A(T; \phi) - J_A(T'; \phi)| \leq C_A \, d_T(T, T').$$

To simplify notation, we drop ϕ in the following.

For two m-dimensional Gaussian distributions, the symmetric KL divergence in Eq. (3) can be written as

$$J_A(T) = \frac{1}{4}\Big[\operatorname{tr}(S_A^{-1} S_T) + \operatorname{tr}(S_T^{-1} S_A) - 2m + (\mu_T - \mu_A)^\top (S_A^{-1} + S_T^{-1})(\mu_T - \mu_A) \Big]. \tag{14}$$

Fix $T, T' \in \mathcal{T}$ and define $\Delta_\mu := \mu_{T'} - \mu_T$ and $\Delta_\Sigma := \Sigma_{T'} - \Sigma_T$. Consider the line segment between their Gaussian parameters:

$$\mu_t := \mu_T + t\Delta_\mu, \qquad S_t := S_T + t\Delta_\Sigma, \qquad t \in [0, 1].$$

Since $S_T = \Sigma_T + \rho I$ and $S_{T'} = \Sigma_{T'} + \rho I$, the covariance path may equivalently be written as

$$S_t = (1 - t) S_T + t S_{T'}.$$

Since Σ_T and $\Sigma_{T'}$ are positive semidefinite, we have $S_T \succeq \rho I$ and $S_{T'} \succeq \rho I$. Consequently,

$$S_t \succeq \rho I \qquad \text{and} \qquad \|S_t^{-1}\|_2 \leq \frac{1}{\rho}$$

for all $t \in [0,1]$.

Define $\delta_t := \mu_t - \mu_A$. Using the convexity of the Euclidean norm and Eq. (10),

$$\begin{aligned}\|\delta_t\|_2 &= \|(1-t)(\mu_T - \mu_A) + t(\mu_{T'} - \mu_A)\|_2 \\ &\leq (1-t)\|\mu_T - \mu_A\|_2 + t\|\mu_{T'} - \mu_A\|_2 \leq R.\end{aligned}$$

Now define the scalar function

$$g(t) := \frac{1}{4}\Big[\operatorname{tr}(S_A^{-1}S_t) + \operatorname{tr}(S_t^{-1}S_A) - 2m + \delta_t^\top (S_A^{-1} + S_t^{-1})\delta_t\Big]. \tag{15}$$

By construction, $g(0) = J_A(T)$ and $g(1) = J_A(T')$. Along the path, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}S_t = \Delta_\Sigma, \qquad \frac{\mathrm{d}}{\mathrm{d}t}\delta_t = \Delta_\mu.$$

The derivative of the inverse satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}S_t^{-1} = -S_t^{-1}\Delta_\Sigma S_t^{-1}.$$

Differentiating Eq. (15) therefore gives

$$4g'(t) = \operatorname{tr}(S_A^{-1}\Delta_\Sigma) - \operatorname{tr}\big(S_t^{-1}\Delta_\Sigma S_t^{-1}S_A\big) + 2\Delta_\mu^\top(S_A^{-1} + S_t^{-1})\delta_t - \delta_t^\top S_t^{-1}\Delta_\Sigma S_t^{-1}\delta_t. \tag{16}$$

We next bound the covariance-dependent terms. By spectral–nuclear norm duality,

$$\big|\operatorname{tr}(S_A^{-1}\Delta_\Sigma)\big| \leq \|S_A^{-1}\|_2\|\Delta_\Sigma\|_* = \frac{1}{\alpha_A}\|\Delta_\Sigma\|_*. \tag{17}$$

For the second term, cyclicity of the trace gives

$$\operatorname{tr}\big(S_t^{-1}\Delta_\Sigma S_t^{-1}S_A\big) = \operatorname{tr}\big(S_t^{-1}S_A S_t^{-1}\Delta_\Sigma\big)\,.$$

Hence, again by spectral–nuclear norm duality,

$$\begin{aligned}\big|\operatorname{tr}\big(S_t^{-1}\Delta_\Sigma S_t^{-1}S_A\big)\big| &\leq \|S_t^{-1}S_A S_t^{-1}\|_2\|\Delta_\Sigma\|_* \\ &\leq \|S_t^{-1}\|_2^2\|S_A\|_2\|\Delta_\Sigma\|_* \\ &\leq \frac{M}{\rho^2}\|\Delta_\Sigma\|_*,\end{aligned} \tag{18}$$

where we used

$$\|S_t^{-1}\|_2 \leq \frac{1}{\rho} \qquad \text{and} \qquad \|S_A\|_2 \leq \|S_A\|_* \leq M.$$

For the final covariance-dependent term, set $B_t := S_t^{-1}\Delta_\Sigma S_t^{-1}$. Then, $\delta_t^\top B_t\delta_t = \operatorname{tr}\big(\delta_t^\top B_t\delta_t\big)$. Using cyclicity of the trace,

$$\operatorname{tr}\big(\delta_t^\top B_t\delta_t\big) = \operatorname{tr}\big(\delta_t\delta_t^\top B_t\big) = \operatorname{tr}\big(S_t^{-1}\delta_t\delta_t^\top S_t^{-1}\Delta_\Sigma\big)\,.$$

Therefore,

$$\begin{aligned}\big|\delta_t^\top S_t^{-1}\Delta_\Sigma S_t^{-1}\delta_t\big| &= \big|\operatorname{tr}\big(S_t^{-1}\delta_t\delta_t^\top S_t^{-1}\Delta_\Sigma\big)\big| \\ &\leq \big\|S_t^{-1}\delta_t\delta_t^\top S_t^{-1}\big\|_2\,\|\Delta_\Sigma\|_* \\ &= \|S_t^{-1}\delta_t\|_2^2\|\Delta_\Sigma\|_* \\ &\leq \|S_t^{-1}\|_2^2\|\delta_t\|_2^2\|\Delta_\Sigma\|_* \\ &\leq \frac{R^2}{\rho^2}\|\Delta_\Sigma\|_*.\end{aligned} \tag{19}$$

The mean-dependent term satisfies

$$\begin{aligned}\left|2\Delta_\mu^\top(S_A^{-1}+S_t^{-1})\delta_t\right| &\le 2\|\Delta_\mu\|_2\left(\|S_A^{-1}\|_2+\|S_t^{-1}\|_2\right)\|\delta_t\|_2\\ &\le 2R\left(\frac{1}{\lambda_A}+\frac{1}{\rho}\right)\|\Delta_\mu\|_2. \end{aligned} \tag{20}$$

Combining Eqs. (17)–(20) with Eq. (16) yields, uniformly for $t \in [0,1]$,

$$|g'(t)| \le \frac{1}{4}\left(\frac{1}{\lambda_A}+\frac{M+R^2}{\rho^2}\right)\|\Delta_\Sigma\|_* + \frac{R}{2}\left(\frac{1}{\lambda_A}+\frac{1}{\rho}\right)\|\Delta_\mu\|_2. \tag{21}$$

By the fundamental theorem of calculus,

$$|J_A(T')-J_A(T)| = |g(1)-g(0)| = \left|\int_0^1 g'(t)\,\mathrm{d}t\right| \le \int_0^1 |g'(t)|\,\mathrm{d}t.$$

Since the bound in Eq. (21) is independent of t, it follows that

$$|J_A(T')-J_A(T)| \le c_{\Sigma,A}\|\Sigma_T-\Sigma_{T'}\|_* + c_{\mu,A}\|\mu_T-\mu_{T'}\|_2,$$

where

$$c_{\Sigma,A} := \frac{1}{4}\left(\frac{1}{\lambda_A}+\frac{M+R^2}{\rho^2}\right), \qquad c_{\mu,A} := \frac{R}{2}\left(\frac{1}{\lambda_A}+\frac{1}{\rho}\right).$$

Let $C_A := \max\{c_{\Sigma,A}, c_{\mu,A}\}$. Then, by the definition of $d_{\mathcal{T}}$,

$$|J_A(T')-J_A(T)| \le C_A\left(\|\Sigma_T-\Sigma_{T'}\|_* + \|\mu_T-\mu_{T'}\|_2\right) = C_A\, d_{\mathcal{T}}(T,T').$$

Combining this with Eq. (13) yields

$$|\mathcal{L}_{\mathrm{TTA}}(A,T;\theta) - \mathcal{L}_{\mathrm{TTA}}(A,T';\theta)| \le 2\sqrt{C_A\, d_T(T,T')}.$$

Thus, $d_T(T_k,T) \to 0$ implies

$$\mathcal{L}_{\mathrm{TTA}}(A,T_k;\theta) \to \mathcal{L}_{\mathrm{TTA}}(A,T;\theta),$$

which proves stability.

□

C TTA TRAINING

Base Models and Feature Representations All TTA experiments start from a pretrained source-domain surrogate. The Plate 2D and SIMSHIFT experiments use Transolver-based models, whereas the EngiBench experiments use conditional diffusion models. In our specific setup, task-dependent parameters, such as thickness or temperature, are encoded through a conditioner network. The resulting conditioning output is passed to the base model, which, in our case, is a Transolver or diffusion model. We extract features from the main body's output and define the split between the representation learner and the predictor.

The exact location of this split depends on the dataset. For SIMSHIFT and Plate 2D, the network is split before the decoder, such that the conditioner and the Transolver body together constitute the representation learner g^ϕ, while the decoder acts as the predictor h^ψ. For EngiBench, the split is applied after the conditioner, meaning that the conditioner serves as the representation learner g^ϕ and the remaining diffusion network functions as the predictor h^ψ.

Prediction-Aware Importance Weights. Let $\mathbf{z} \in \mathbb{R}^n$ denote the feature representation at the selected split. We define W as the stored PyTorch weight tensor of the architecture-specific linear layer associated with this split. For output-decoder, CNN, and MLP splits, this is the final linear prediction layer. For example, for the Transolver decoder,

$$W = \texttt{decoder.mlp[-1].weight} \in \mathbb{R}^{d\times n}, \tag{22}$$

which is the exact linear map from the pre-decoder representation to the raw nodewise output. Accordingly, the prediction-aware importance of a source principal direction $\mathbf{v}_S^{(k)}$ is computed from $\|W\mathbf{v}_S^{(k)}\|_2$. No Jacobian or fitted linear approximation is used.

When deformation prediction is enabled, $d_{\text{out}} = d_{\text{field}} + d_{\text{deformation}}$; therefore, the norm includes both the predicted physical fields and the displacement-coordinate channels. The nodewise latent features are averaged per sample when computing the PCA statistics, whereas W remains the shared nodewise decoder weight.

For conditioner splits, W is the stored weight of the final linear layer of the conditioning encoder. In this case, $\|W\mathbf{v}_S^{(k)}\|_2$ measures the influence of the principal direction on the conditioning embedding, rather than the sensitivity of the complete nonlinear predictor.

Source Statistics and Sample Selection For all TTA experiments, source samples are selected from the full source training dataset using Algorithm 1, and the corresponding set is stored for adaptation and model selection. These samples are additionally used to compute the source statistics μ_S and σ_S.

The minimum-eigenvalue criterion is degenerate while the selected information matrix is rank deficient. We therefore use a two-stage implementation. First, the centered source-feature matrix $\mathbf{Z} = g^\phi(\mathbf{x}_S) - \text{mean}\big(g^\phi(\mathbf{x}_S)\big) \in \mathbb{R}^{N_S \times n}$ is projected onto

$$r = \min(r_{95\%}, m, \text{rank}(\mathbf{Z})) \tag{23}$$

principal components, where $r_{95\%}$ is the smallest number of components explaining at least 95% of the variance. For notational consistency with Algorithm 1, we continue to denote the resulting projected matrix in $\mathbb{R}^{N_S \times r}$ by $\mathbf{Z}$.

We then apply column-pivoted QR to $\mathbf{Z}^\top$. Its first r pivots initialize the selected-index set $\mathcal{I}$, while $\mathcal{U} = \{1, \ldots, N_S\} \setminus \mathcal{I}$ contains the remaining indices. Starting from these sets, the remaining samples are selected using the minimum-eigenvalue updates in lines 6–10 of Algorithm 1 until $|\mathcal{I}| = m$. If $r = m$, the complete sample budget is consumed by the pivoted-QR initialization, and no minimum-eigenvalue updates are required.

At test time, the target features are projected onto the precomputed source principal directions. For each direction k, the target mean and variance, $(\tilde{\mu}_T^{(k)}, (\tilde{\sigma}_T^{(k)})^2)$, are computed and aligned with the corresponding source statistics $(\tilde{\mu}_S^{(k)}, (\tilde{\sigma}_S^{(k)})^2)$. Following Adachi et al. (2025), we minimize the weighted channel-wise symmetric KL divergence at each model update. Updates are performed on domain batches obtained by grouping target samples according to the physical parameter defining the distribution shift. Examples include Plate 2D angles, forming thickness, rolling reduction, motor geometry, heatsink fin count, beam minimum radius, and heat-conduction volume fraction. Discrete parameters are grouped by their observed values, while continuous parameters are divided into a configurable number of bins. Each resulting group forms a single target batch. The model receives exactly one adaptation update on that batch, is evaluated on the same batch, and is then reset before evaluating another learning rate or domain group. This prevents updates from one target domain from leaking into another.

Adaptation is restricted to layer normalization (Ba et al., 2016) parameters: for EngiBench, only the layer normalization layers of the conditioner are updated, whereas for SIMSHIFT and Plate 2D, layer normalization parameters of both the Transolver and the conditioner are adapted. All remaining parameters are kept fixed.

For parameter tuning, we compute the latent density ratio after a single forward pass through the test-time-adapted model. To estimate this ratio, the latent source and target mean and covariance are computed and stored prior to model adaptation. After adapting the model with a candidate learning rate, the density ratio is evaluated on the selected labeled source samples and used to weight their prediction losses. Since very-high-dimensional settings are prone to a lot of noise in the covariance estimation (Zhang et al., 2023), we decided to perform a dimension reduction to improve robustness. This enables reliable covariance estimation using the E-optimally selected source samples. The resulting density-ratio-weighted source loss provides an estimate of target risk and is used solely for model selection.

Using the estimate in Eq. (9), we perform model selection via line search over the TTA learning rate, evaluating performance after each adaptation step and stopping once further updates no longer improve

the objective. The search over learning rates (lr) is terminated based on performance measured on the E-optimally selected source samples. We use the Root Mean Squared Error (RMSE) for the SIMSHIFT and Plate 2D datasets and the COMP metric for EngiBench. We set the hyperparameter search for the learning rates to $[0.01, 0.005, 0.001, 0.0005, 0.0001]$.

To ensure robustness, we use 4 different models for the SIMSHIFT benchmark and Plate 2D dataset, 2 for the structural beam bending dataset, and 2 for the 2D heat conduction dataset.

Baselines For model comparison, we evaluate existing TTA methods commonly used in both regression and classification tasks. For SSA and Tent, we follow the procedures described in their respective method sections (Wang et al., 2021; Adachi et al., 2025). In the implementation of SSA, the top-K eigenvalues need to be identified to compute statistics only based on a sparse set of information. We do this for each dataset. For Tent, an additional modification is required for the SIMSHIFT dataset: since entropy minimization is applied by minimizing predictive uncertainty, we train a model that explicitly predicts both mean and variance.

D Experimental Setup

In the following paragraphs, we detail the experimental setup, including the selected models and our training and testing strategy.

D.1 Model Architectures & Pretraining

We employ different model architectures to evaluate our TTA method. The models are based on the architectures provided in the benchmark datasets Setinek et al. (2025) and Felten et al. (2025), implemented in PyTorch, and designed for conditional regression or optimization tasks. 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 (Wu et al., 2024) 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 (Peebles & Xie, 2023). We use a latent dimension of 128, a slice base of 32, and four attention layers. This results in a model with 0.57M parameters. We additionally employ a larger model with 56, 128, and 8 layers for the more complex dataset, leading to 4.07M parameters. We provide a detailed overview in Table 5 for both datasets (Plate 2D and SIMSHIFT).

Table 5: Hyperparameters for the Transolver models used on the SIMSHIFT and plate datasets.

Hyperparameter	**SIMSHIFT**	**Plates2D**	**Class Argument Name**
Base channels	128	128	`transolver_base`
Latent channels	8	8	`latent_channels`
Transformer layers	4	4	`num_layers`
Attention heads	4	4	`num_heads`
Physics-attention slices	32	32	`slice_base`
Conditioning mode	DiT	DiT	`conditioning_mode`
Node-feature dimension	0	4	`node_feature_dim`
Dropout	0.1	0.1	`dropout_prob`
Attention dropout	0.1	0.1	`attn_dropout_prob`

Diffusion Model. As a diffusion model, we employ a conditional U-Net (Ronneberger et al., 2015) from Hugging Face's `diffusers` library[2].

The model works as a denoiser, taking a noisy field and a conditioning vector from the conditioning network described above and producing a noise prediction. We summarize all hyperparameters of our diffusion model in Table 6. To train the model, we use the standard Denoising Diffusion Probabilistic Models (DDPM) objective of noise prediction ("ϵ-prediction") with 100 diffusion steps and a `squaredcos_cap_v2` beta scheduler.

Table 6: Hyperparameters for our conditional diffusion U-Net. This setup leads to a model size of 17.5M parameters.

Hyperparameter	**Value**	**HF Class Argument Name**
Block channels (low→high)	$[32, 64, 128, 256]$	`block_out_channels`
Layers per block	2	`layers_per_block`
Transformer layers / block	1	`transformer_layers_per_block`
Cross-attention dim	64	`cross_attention_dim`
Only cross-attention	`True`	`only_cross_attention=True`
Normalization groups	16	`norm_num_groups`
Activation	SiLU	`act_fn`

Pretraining setup. All unregularized baseline (*"Source"*) models are pretrained using the following setup: We use an initial learning rate of 10^{-3} with a cosine decay scheduler and weight decay of 10^{-5}. Training runs for up to 500, 1500, 3000 and 3000 epochs on *Beams2D*, *HeatConduction2D*, *Plates2D* and *SIMSHIFT*, respectively, with early stopping if the validation loss does not improve for 500 epochs. We enable gradient clipping and maintain an Exponential Moving Average (EMA) of the model parameters with decay 0.95. Automatic Mixed Precision (AMP) is enabled only for the large-scale *heatsink* dataset; for all others we train in `float32`. Batch size is 64 for EngiBench baselines and 16 for SIMSHIFT baselines.

D.2 Datasets

Plate 2D

This dataset mimics solid mechanics applications under varying domain geometries and boundary conditions. We simulate a perforated plate clamped on one end and apply a varying load on the other. The system can be described using the equations of static linear elasticity. Without body forces, they read as follows:

$$\begin{aligned} -\nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) &= \mathbf{0} && \text{in } \Omega \\ \boldsymbol{\sigma}(\mathbf{u}) &= \lambda \operatorname{tr}(\boldsymbol{\varepsilon}(\mathbf{u}))\, \mathbf{I} + 2\mu\, \boldsymbol{\varepsilon}(\mathbf{u}) \\ \boldsymbol{\varepsilon}(\mathbf{u}) &= \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\top}\right), \end{aligned} \tag{24}$$

where $\mathbf{u}$ is the displacement field. The Cauchy stress tensor $\boldsymbol{\sigma}$, related to the displacement through Hooke's Law for isotropic materials, and $\boldsymbol{\varepsilon}$ is the linearized strain tensor. In our setup, we employ the plane-stress formulation, where the Lamé parameters, λ and μ are related to Young's modulus E and Poisson's ratio ν via

$$\mu = \frac{E}{2(1+\nu)}, \qquad \lambda = \frac{2\mu\nu}{1-\nu}. \tag{25}$$

The system is subject to the following boundary conditions:

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_D && \text{on } \Gamma_D \\ \boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n} &= \mathbf{t} && \text{on } \Gamma_N, \end{aligned} \tag{26}$$

[2] `UNet2DConditionModel`

where Γ_D and Γ_N are the Dirichlet and Neumann boundaries respectively, $\mathbf{n}$ is the outward-pointing unit normal and $\mathbf{t}$ is the prescribed surface traction.

In our setup, we model standard structural steel using $E = 210\,\text{GPa}$ and $\nu = 0.3$. To vary the geometry and the boundary conditions, we randomly place differently sized holes into our domain and change the magnitude as well as the angle of the edge traction. Figure 5 visualizes our domain and Table 7 describes the sampled boundary conditions and geometric parameters.

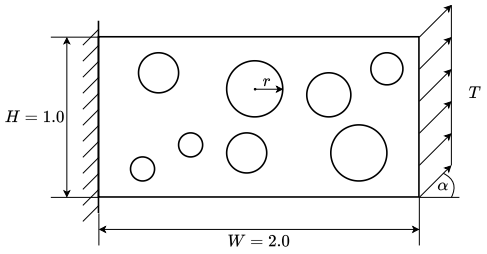

Figure 5: Perforated steel plate, clamped on the left and traction on the right boundary.

Table 7: Sampling ranges for boundary conditions and geometry parameters for the linear elasticity experiment.

Parameter	**Symbol**	**Range**
Traction magnitude	T	10^5–10^7 Pa
Traction angle	α	0–2π rad
Number of holes	–	1–20
Hole radii	r	0.04–0.30 m

To generate the dataset, we cast Eqs. (24) and (26) into a weak form and solve it computationally using the FEniCSx finite-element framework (Baratta et al., 2025; Alnæs et al., 2014; Scroggs et al., 2022a;b). To generate the necessary meshes, we use `gmsh` (Geuzaine & Remacle, 2009) via a parametric meshing script, enabling us to create meshes with varying node counts.

Mimicking common engineering scenarios, the learning task is the prediction of the von Mises stress. Under the plane-stress assumption it is computed as

$$\sigma_{\text{vM}} = \sqrt{\sigma_{11}^2 - \sigma_{11}\sigma_{22} + \sigma_{22}^2 + 3\,\sigma_{12}^2}, \tag{27}$$

a scalar equivalent stress that summarizes the multiaxial stress state and serves as the standard yield criterion for materials such as structural steel.

Section E outlines the source-target domain splitting routine applied for this dataset.

E Distribution Shifts

EngiBench Figures 6 to 7 show t-SNE visualizations of the networks' latent spaces for models trained across the full range of condition variables. For *structural beam bending* (Figure 6), `volfrac` and `rmin` cause the clearest structure in latent space. We therefore chose to split the source and target domain depending on the value range of `rmin`. For *2D heat conduction* (Figure 7), `volume` and `length` exhibit comparable influence on the latent space distribution. Following the same protocol, we split along `volume`. The resulting source and target ranges and sizes for both datasets can be found in Table 8.

Table 8: Defined distribution shifts (source and target domains) for each dataset.

Dataset	Parameter	Description	Source range (no. samples)	Target range (no. samples)
Beams2D	rmin	Minimum feature length of beam members.	[1.5, 3.25) (3087)	[3.25, 4] (353)
HeatConduction2D	volume	Volume limits on the material distributions.	[0.3, 0.465) (231)	[0.465, 0.6] (39)

Plate 2D To construct the domain shift for this dataset, we train a model across the whole simulation parameter ranges and inspect the learned embedding manifolds. We display the influence of the simulation parameters on the model's latent representation in Figure 8. The embedding shows no clear structure with respect to the number of holes, whereas both the magnitude and angle of the applied traction induce visible groupings. We therefore split our domains along ranges of the traction angle. We construct the domain split along the absolute traction angle $|\alpha|$, with the source domain comprising all samples with $|\alpha| \in [0, 0.6\pi]$ and the target domain those with $|\alpha| \in [0.74\pi, 0.87\pi]$. All models evaluated in the TTA experiments were pretrained exclusively on the source subset and

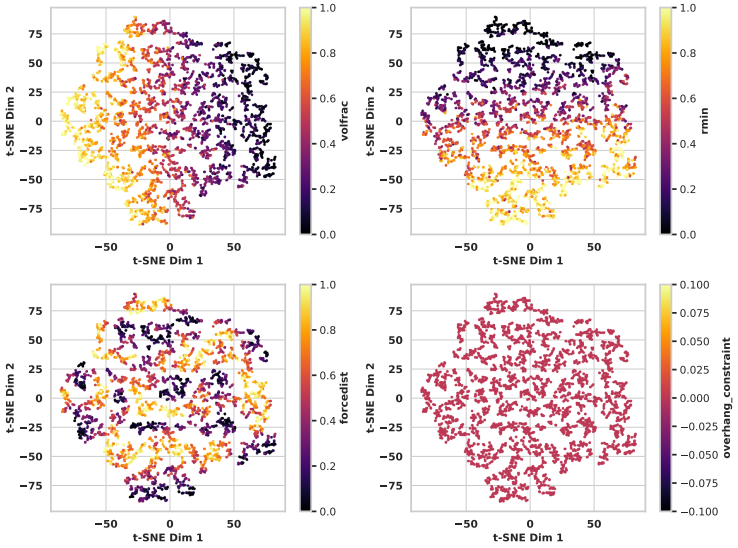

Figure 6: t-SNE visualization of the model's latent space on the *structural beam bending* dataset. While `overhang_constraint` and `forcedist` are either constant or exhibit almost a uniform distribution, `volfrac` and `rmin` exhibit a clear structure.

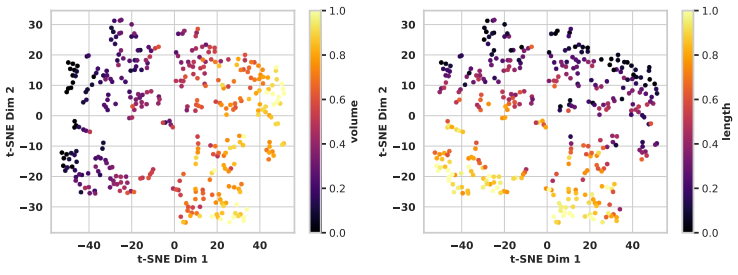

Figure 7: t-SNE visualization of the model's latent space on the *2D heat conduction* dataset. Both conditions (`volume` and `length`) exhibit a clear structure in the latent space.

had no exposure to target-domain samples or labels during pretraining. Figure 8 provides a post-hoc visualization of the resulting latent representations across the simulation parameters.

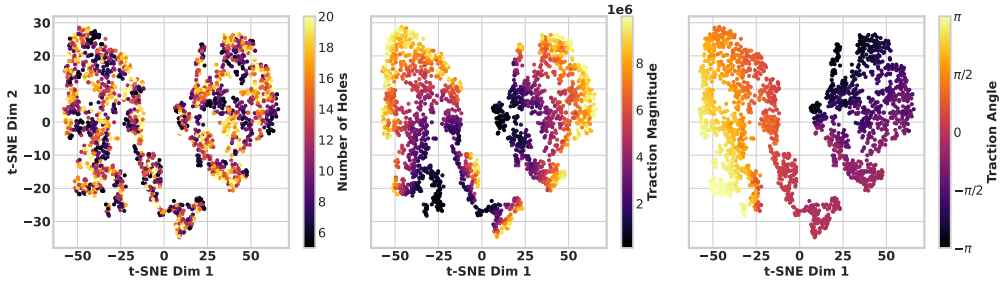

Figure 8: t-SNE visualization of the model's latent space on the *Plate 2D* dataset. While the number of holes exhibits a uniform distribution, the force angle and magnitude show clear structure in the latent space.

SIMSHIFT For a description of the distribution shifts, parameter ranges and latent space visualizations, we refer to Figure 4 and Appendix C.1 of Setinek et al. (2025).

F Additional Results

Mesh size results

Table 9: Performance comparison of different fidelity ranges and their scaled model size. Mean ± std for multiple model seeds.

Fidelity	Model Parameters	Method	RMSE	MAE	R^2
4.7k	211k	Source	0.309 ± 0.008	0.232 ± 0.006	0.798 ± 0.011
		SSA	0.323 ± 0.014	0.244 ± 0.012	0.781 ± 0.019
		ETA	**0.304 ± 0.010**	**0.228 ± 0.007**	**0.803 ± 0.012**
9k	384k	Source	0.307 ± 0.009	0.232 ± 0.006	0.798 ± 0.011
		SSA	0.306 ± 0.007	0.230 ± 0.008	0.799 ± 0.005
		ETA	**0.297 ± 0.009**	**0.224 ± 0.007**	**0.810 ± 0.012**
18.1k	622k	Source	0.306 ± 0.001	0.230 ± 0.001	0.800 ± 0.001
		SSA	0.312 ± 0.012	0.232 ± 0.011	0.793 ± 0.012
		ETA	**0.298 ± 0.003**	**0.225 ± 0.002**	**0.812 ± 0.007**
29.5k	1290k	Source	0.355 ± 0.024	0.271 ± 0.019	0.729 ± 0.045
		SSA	0.345 ± 0.014	0.264 ± 0.013	0.744 ± 0.023
		ETA	**0.294 ± 0.014**	**0.223 ± 0.001**	**0.806 ± 0.021**

Different sample selection overview The following tables combine the results for the different sample selection ablations on the examples of SIMSHIFT and Plate 2D.

Table 10: RMSE comparison of random and E-optimal sample selection in ETA, on the *rolling*, *forming*, *motor*, *heatsink* and *plates* (across fidelities) datasets.

Method	Rolling	Forming	Motor	Heatsink	Fidelity		
					Small	Medium	Large
Random	1.005 ± 0.010	0.069 ± 0.013	0.055 ± 0.005	0.621± 0.042	0.300 ± 0.009	0.298 ± 0.008	0.349 ± 0.022
ETA	**1.004 ± 0.007**	**0.068 ± 0.003**	**0.053 ± 0.003**	**0.615 ± 0.001**	0.300 ± 0.009	0.298 ± 0.003	**0.294 ± 0.014**

Total compute m sample selection Table 4 isolates the per-step cost against SSA, while Table 11 reports the cumulative cost over the full target set as m grows.

Table 11: End-to-end adaptation runtime over the full target set (hot rolling, forming) as a function of m. Increases are relative to $m = 2$.

Dataset	m	Runtime (s)	Absolute increase (s)	Percentage increase (%)
Rolling	2	4.58	–	–
	5	6.03	1.45	31.56
	8	6.78	2.20	47.96
	10	9.92	5.34	116.55
	all	199.50	194.92	4253.16
Forming	2	8.75	–	–
	5	9.25	0.50	5.72
	8	9.58	0.83	9.51
	10	12.05	3.30	37.77
	all	259.46	250.71	2866.11

Memory usage for $m = 8$

Table 12: Memory usage for $m = 8$ across datasets, including absolute memory consumption, full memory usage, and the fraction of full memory utilized

Dataset	Memory $m = 8$ (MB)	Full dataset (GB)	Percentage (%)
Rolling	0.58	3.34	0.02
Forming	1.76	17.78	0.01
Motor	4.64	15.61	0.03
Heatsink	4.08	542.63	0.00
Beams	0.46	0.20	0.23
Heat2D	0.93	0.04	2.57