# SIMSHIFT: A BENCHMARK FOR ADAPTING NEURAL SURROGATES TO DISTRIBUTION SHIFTS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Neural surrogates for Partial Differential Equations (PDEs) often suffer significant performance degradation when evaluated on unseen problem configurations, such as new initial conditions or structural dimensions. Meanwhile, Domain Adaptation (DA) techniques have been widely used in vision and language processing to generalize from limited information about unseen configurations. In this work, we address this gap through two focused contributions. First, we introduce SIMSHIFT, a novel benchmark dataset and evaluation suite composed of four industrial simulation tasks spanning diverse processes and physics: *hot rolling*, *sheet metal forming*, *electric motor design* and *heatsink design*. Second, we extend established DA methods to state-of-the-art neural surrogates and systematically evaluate them. These approaches use parametric descriptions and ground truth simulations from multiple source configurations, together with only parametric descriptions from target configurations. The goal is to accurately predict target simulations without access to ground truth simulation data. Extensive experiments on SIMSHIFT highlight the challenges of out-of-distribution neural surrogate modeling, demonstrate the potential of DA in simulation, and reveal open problems in achieving robust neural surrogates under distribution shifts in industrially relevant scenarios.

## 1 INTRODUCTION

PDE simulations are essential tools for understanding and predicting physical phenomena in engineering and science (Evans, 2010). Over recent years, machine learning has emerged as a novel and promising modeling option for complex systems (Brunton & Kutz, 2020), significantly accelerating and augmenting simulation workflows across diverse applications, including weather and climate forecasting (Pathak et al., 2022; Bodnar et al., 2025), material design (Merchant et al., 2023; Zeni et al., 2025) and protein folding (Abramson et al., 2024) to name a few.

In practice, however, models are often deployed outside of their training distribution. This *distribution shift* (Quionero-Candela et al., 2009; Wang et al., 2023) often leads to a significant performance degradation (Bonnet et al., 2022; Herde et al., 2024). A well known analogue is clinical microscopy: models trained with data collected at a few hospitals often fail when deployed at others because microscopes, staining protocols, and lighting conditions differ (Tellez et al., 2019; Koh et al., 2020). For neural surrogates an analogous "instrument shift" arises from new initial conditions, such as material parameters or mesh geometries not encountered during training. Robustness to distribution shifts is crucial for industrial adoption and deployment also because it is becoming a compliance requirement, as stated by Article 15 of the EU AI Act (European Union, 2024).

While methods for increasing out-of-distribution performance have been at the center of research for a long time (Ben-David et al., 2006; Shimodaira, 2000; Sugiyama et al., 2007b), to the best of our knowledge, no benchmark systematically investigates such methods on simulation tasks. Addressing this gap is particularly relevant in scientific and industrial settings, where generating ground truth simulation data is costly, limiting the diversity of training configurations. In contrast, parametric descriptions, such as material types or structural dimensions, are often readily available or easy to generate. This problem is known as *Unsupervised Domain Adaptation (UDA)* (Ben-David et al., 2010), where parametric (input) descriptions and full simulation outputs are available for each *source* configuration, while only input descriptions are provided for *target* configurations, without corresponding outputs.

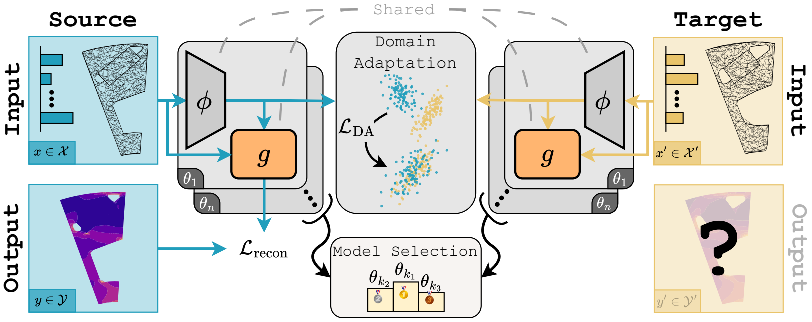


Figure 1: Schematic overview of the SIMSHIFT framework. In training, the model has access to inputs (e.g., parameters and meshes), corresponding outputs $(x, y)$ from the source domain (left, blue), and only inputs $x'$ from the target domain (right, yellow) are available. The neural operator $g$ and the conditioning network $\phi$ are shared across domains and jointly optimized. Two loss terms are used: $\mathcal{L}_{\text{recon}}$, computed on source labels, and $\mathcal{L}_{\text{DA}}$, which aligns source and target $\phi$ features. After training, unsupervised model selection strategies choose $\theta_{k1}$, which is expected to perform best on target domain.

To investigate the potential of UDA for neural surrogate modeling, we provide simulation data across a range of realistic tasks from industrial engineering design. We introduce a comprehensive benchmark that evaluates established UDA methods and neural surrogates. An overview of the framework is shown in Figure 1. Our contributions can be summarized as follows:

- We propose four practical datasets with flexible distribution shifts in *hot rolling*, *sheet metal forming*, *electric motor*, and *heatsink* design, based on realistic simulation setups.
- We present, to the best of our knowledge, the first joint study of established neural surrogate architectures and UDA on engineering simulations with unstructured meshes.
- We introduce *SIMSHIFT*, a modular benchmarking suite that complements our datasets with baseline models and algorithms. It allows easy integration of new simulations, machine learning methods, domain adaptation techniques, and model selection strategies.

## 2 RELATED WORK

**Unsupervised Domain Adaptation.** UDA research covers a wide spectrum of results from theoretical foundations (Ben-David et al., 2010; Zellinger et al., 2021a) to modern deep learning methods (Liu & Xue, 2021; Zellinger et al., 2019; Zhu et al., 2021; Long et al., 2018). A prominent class of methods, dubbed as *representation learning*, aims to map the data to a feature space, where source and target representations appear similar, while maintaining enough information for accurate prediction. To enforce feature similarity between domains, algorithms often employ statistical (Sun & Saenko, 2016; Gretton et al., 2006; Zhang et al., 2019; Shalit et al., 2017) or adversarial (Ganin et al., 2015; Tzeng et al., 2017) discrepancy measures. One crucial yet frequently overlooked factor in the success of UDA methods is model selection. Numerous studies underline the critical impact of hyperparameter choices on UDA algorithm performance, often overshadowing the adaptation method itself (Musgrave et al., 2021; Zellinger et al., 2021b; Dinu et al., 2023; Yang et al., 2024). Even more, since labeled data is unavailable in the target domain, standard validation approaches become infeasible. Thus, it is essential to jointly evaluate adaptation algorithms alongside their associated unsupervised model selection strategies. In this work, we focus on importance weighting strategies (Sugiyama et al., 2007a; You et al., 2019), which stand out by their general applicability, theoretical guarantees and high empirical performance.

**Benchmarks for Unsupervised Domain Adaptation.** Numerous benchmark datasets and evaluation protocols have been established for UDA methods across various machine learning domains, including computer vision (Venkateswara et al., 2017; Peng et al., 2018; Arjovsky et al., 2019), natural language processing (Blitzer et al., 2007), timeseries data (Ragab et al., 2022) and tabular data (Gardner et al., 2023). However, to the best of our knowledge, systematic UDA benchmarking for neural surrogates remains unexplored.

**Benchmarks for Neural Surrogates.** Recent years have seen a surge of surrogates belonging to the group of neural operators (see Appendix A), and benchmarks have grown alongside them. However, designing a robust and fair benchmark in the realm of PDEs is difficult and the current literature is not without shortcomings (Brandstetter, 2025). Many focus on solving PDEs on structured, regular grids (Gupta & Brandstetter, 2022; Takamoto et al., 2022; Ohana et al., 2024), which serve as valuable platforms for developing and testing new algorithms. However, these overlook the irregular meshes commonly used in large scale industrial simulations. In that direction, other benchmarks extend to Computational Fluid Dynamics (CFD) on irregular static meshes for airfoil simulations (Bonnet et al., 2022), aerodynamics for automotive (Elrefaie et al., 2024a;b), more academic fluid problems (Luo et al., 2023), and even particle based Smoothed Particle Hydrodynamics simulations (Toshev et al., 2023; 2024). Finally, and most closely related to our work, recent efforts have explored the application of Active Learning techniques (Cohn et al., 1996; Ren et al., 2021) to neural surrogates, introducing a benchmark specifically designed for scenarios where data is scarce (Musekamp et al., 2025). Despite these contributions, all current benchmarks often fall short when addressing a critical issue: the significant performance drop models exhibit under distribution shifts, i.e., when encountering simulation configurations beyond their training setting (Quionero-Candela et al., 2009).

## 3 DATASET PRESENTATION

Our datasets follow three design principles. (i) **Industry relevance:** They reflect practical, real-world simulation use-cases. The benchmark covers a diverse set of problems, including 2D as well as 3D cases. (ii) **Parametrized conditions:** The behavior of all simulations depends on the set of initial parameters only. (iii) **Steady-state scenarios:** We constrain them to time independent problems, being the standard use case in industry. Take for example design optimization tasks: most rely on either steady-state or time-averaged solutions rather than detailed transient dynamics. This is not just a modeling convenience, but reflects how simulation is integrated into design pipelines: numerical simulations are used to assess candidates by computing scalar objective values. This practice is well documented established various application areas, including thermal systems (Majumdar, 2021), aerodynamic shape optimization for aircrafts (Martins, 2022), wind turbine design (Martins, 2022), and car aerodynamics (Dumas, 2007). Additionally with this constraint we avoid additional complexities such as autoregressive error accumulation in neural surrogates (Lippe et al., 2023).

The datasets were generated using the commercial Finite Element Method (FEM) software *Abaqus*, the open-source simulation software *HOTINT* and the open-source CFD package *OpenFoam 9*. [1] An overview of each dataset together with its most important parameters and a custom metric, motivated by engineering practice, is presented in Sections 3.1 to 3.4. Additionally, we provide detailed descriptions of the respective numerical simulations in Appendix G. Since the behavior of each simulation task is entirely determined by its input parameters, we predefine source and target domains by partitioning the parameter space into distinct, non-overlapping regions. A detailed explanation of the domain splitting strategy is provided in Section 3.5. Table 1 summarizes key characteristics of each dataset, including physical dimensionality, mesh resolution, number of conditioning parameters, and total dataset size. All datasets are publicly hosted on Hugging Face[2].

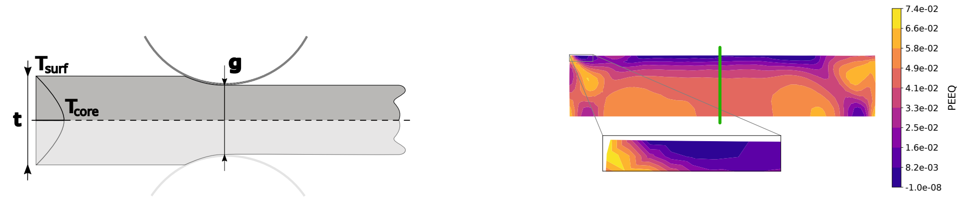
Table 1: Overview of the benchmark datasets. Heatsink meshes are subsampled to a fourth of their original size. Detailed descriptions of the parameter sampling ranges can be found in Appendix G.

| Dataset | Origin | Samples | Output channels | Avg. # nodes | Varied simulation parameters | Dim | Size (GB) |
|---|---|---|---|---|---|---|---|
| Rolling | Metallurgy | 4,750 | 10 | 576 | 4 | 2D | 0.5 |
| Forming | Manufacturing | 3,315 | 10 | 6,417 | 4 | 2D | 4.1 |
| Motor | Machinery | 3,196 | 26 | 9,052 | 15 | 2D | 13.4 |
| Heatsink | Electronics | 460 | 5 | 1,385,594 | 4 | 3D | 40.8 |

---

[1] Abaqus; HOTINT; OpenFoam 9.

[2] https://huggingface.co/datasets/simshift/SIMSHIFT_data



(a) Illustration of the simulation setup. The parameters correspond to those in Table 14. We use symmetry constraints and only simulate half of the slab.

(b) Metal slab after the process, showing PEEQ as a contour plot. The green line indicates the center cord, along which we measure the custom metric.

Figure 2: Overview of the *hot rolling* simulation scenario.

### 3.1 HOT ROLLING

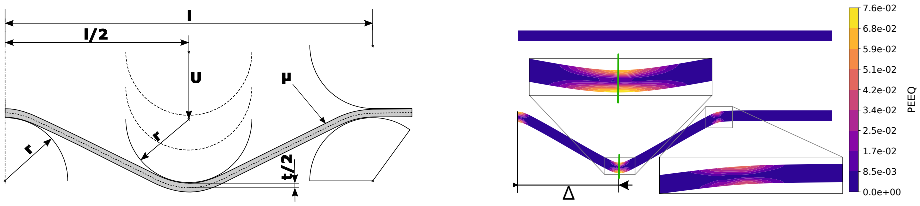
**Problem Description.** The *hot rolling* process plastically deforms a metal slab into a sheet metal product, as visualized in Figure 2. This complex thermo-mechanical operation involves coupled elasto-plastic deformation and heat transfer phenomena (Gupta, 2021; Galantucci & Tricarico, 1999; Jo et al., 2023). The Finite Element (FE) simulation models the progressive thickness reduction and thermal evolution of the material as it passes through a rolling gap, incorporating temperature-dependent material properties and contact between the slab and the rolls. Among the output fields, the key quantity is Equivalent Plastic Strain (PEEQ), representing the material's plastic deformation, visualized in Figure 2b. The custom metric measures the relative error of the PEEQ profile along the slab's vertical center cord (green line in Figure 2b).

**Input parameters** are the initial slab thickness $t$, temperature characteristics $T_{\text{core}}$ and $T_{\text{surf}}$ of the slab, as well as the geometry of the roll gap. To vary the slab deformation we define the thickness reduction as a percentage of the initial thickness: reduction $= \frac{t-g}{t}$, where $g$ is the rolling gap distance. Table 14 in Appendix G.1 shows a detailed overview of the parameter values together with their sampling ranges used to generate the dataset.

### 3.2 SHEET METAL FORMING

**Problem Description.** The *sheet metal forming* process is a critical manufacturing operation widely used across industries such as automotive and aerospace. FEM simulations are commonly employed to estimate critical quantities such as thinning, local plastic deformation and residual stress distribution (Tekkaya, 2000; Ablat & Qattawi, 2017; Folle et al., 2024). The simulation setup consists of a symmetrical workpiece supported at the ends and center, a holder and a punch that deforms the sheet by applying a displacement ($U$ in Figure 3a). The 2D simulation predicts the sheet's elasto-plastic deformation, providing quantities such as stress, elastic and plastic strain distributions (shown in Figure 3b). An essential engineering metric used in practice is the transverse stress (xx-component) distribution along the vertical center cord (green line in Figure 3).

**Input parameters** include the deformed sheet length $l$, the sheet thickness $t$, friction coefficient $\mu$ and the radii of the holder, punch, and supports $r$. Table 15 in Appendix G.2 provides the sampling ranges for data generation.



(a) Illustration of the simulation setup. The parameters correspond to those listed in Table 15.

(b) Material before (top) and after (bottom) the process, shown as PEEQ contours. $\Delta = l/2$

Figure 3: Overview of the *sheet metal forming* simulation scenario.

### 3.3 ELECTRIC MOTOR DESIGN

**Problem Description.** The *electric motor design* dataset encompasses a structural FEM simulation of a rotor in electric machinery, subjected to mechanical loading at burst speed. It is motivated by the conflicting design objectives in rotor development: while magnetic performance favors certain rotor topologies to optimize flux paths and torque generation, structural integrity requires designs capable of withstanding centrifugal loads without plastic deformation (Gerlach et al., 2021; Dorninger et al., 2021). The 2D simulation predicts stress and deformation responses due to assembly pressing forces and centrifugal loads, accounting for the rotor's topology, material properties, and rotation speed. The custom metric measures the relative error in Mises stress along the cord shown in green Figure 4.

**Input Parameters** together with their variations and a detailed technical drawing are omitted from the main body since this case is more complex than the preceding datasets. They are provided in Figure 31 and Table 17, both in Appendix G.3.

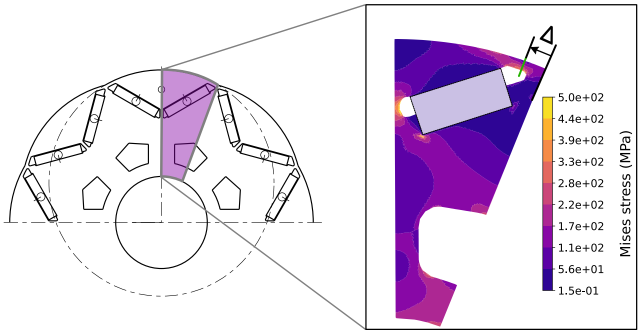


Figure 4: The *electric motor design* simulation scenario, with a schematic sketch of the motor (left) and zoomed-in detail from the simulated radial portion (right). Mises stress field contour plot is shown. The custom error metric is measured along the green line at $\Delta = \frac{t_{rsb1}}{2} + 1.1 * r_{r2}$.

### 3.4 HEATSINK DESIGN

**Problem Description.** The *heatsink design* dataset represents a CFD simulation focused on the thermal performance of heat sinks, commonly used in electronic cooling applications (Arularasan & Velraj, 2010; Rahman et al., 2024). It models the convective heat transfer from a heated base through an array of fins to the surrounding air. The simulation captures how geometric fin characteristics, specifically, the number, height, and thickness of fins, affect the overall heat dissipation, along with the temperature of the heat sink. Outputs include steady state temperature, velocity and pressure fields, enabling the assessment of design efficiency and thermal resistance under varying configurations. The main engineering metric measures the relative error in the temperature distribution along the dashed green line in Figure 5.

**Input Parameters** and their variations as well as an overview of the setup are provided in Appendix G.4.

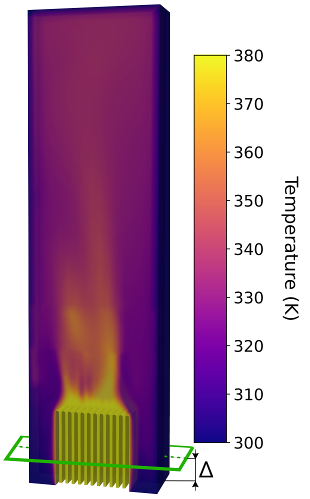


Figure 5: Slice of the *heatsink* 3D temperature field. Custom metric along dashed cord at $\Delta = 0.0025$.

### 3.5 DISTRIBUTION SHIFTS

SIMSHIFT's functionality allows for generating arbitrary n-dimensional parametric shifts for each problem, ensuring flexibility and extensibility. For benchmarking, each dataset includes three predefined distribution shifts: *easy*, *medium* and *hard*, which reflect increasing distributional distance in the respective input spaces (see Table 2 for parameter ranges). The source and target domains are constructed by shifting along the dominant input parameter of each simulation scenario, as suggested by domain experts.

To validate the design of our domain shifts we perform two analyses: (i) **Latent space inspection:** We train models across the full parameter ranges and perform a cluster analysis of their latent representations as the input conditions are varied. The resulting clusters consistently align with the parameters proposed by the experts, indicating that the chosen parameters dominate latent space variation (see visualizations in Figures 25 to 28, Appendix C). (ii) **Transfer difficulty validation:** Scalar parameter differences alone can be misleading regarding the actual shift difficulty experienced by models. We therefore provide the Proxy $\mathcal{A}$-Distance (PAD), which serves as an upper bound on the model's maximum transfer error. It works by bounding the $\mathcal{H}$-divergence, which in turn is an upper bound the maximum transfer error itself (for details see Bouvier et al. (2020), Johansson et al. (2019) and Zellinger et al. (2021b)). We estimate the PAD in the output spaces (ground truth simulation fields) using a PointNet Qi et al. (2017) mesh classifier. The resulting PAD values for each difficulty together with the domain defining parameter splits for all datasets are reported in Table 2.

The PAD values indicate a clear output-space distribution shift across all datasets. To illustrate this more concretely, consider the *hot rolling* dataset: in the *medium* difficulty setting, the range of PEEQ values in the source domain is $[0, 0.19]$, while in the target domain it extends to $[0, 0.28]$. Therefore part of the target solution field lies outside the support of the source field, demonstrating a genuine output-space shift in addition to the parametric input shift.

Beyond the predefined one-dimensional splits, we explore higher-dimensional distribution shifts. In Appendix H.2, we demonstrate that models, adaptation algorithms and model selection strategies exhibit consistent behavior under a two-dimensional shift in the *electric motor design* dataset.

Table 2: SIMSHIFT's predefined distribution shifts. We show the domain defining parameter and its respective ranges for all difficulty levels together with the corresponding PAD.

| **Dataset** | **Parameter** | **Difficulty** | **Source range** | **Target range** | **PAD** |
|---|---|---|---|---|---|
| Rolling | Reduction $r$ $(-)$ | easy | [0.01, 0.13) | [0.13, 0.15] | 1.063 |
| | | medium | [0.01, 0.115) | [0.115, 0.15] | 1.159 |
| | | hard | [0.01, 0.10) | [0.10, 0.15] | 1.210 |
| Forming | Thickness $t$ (mm) | easy | [2, 4.8) | [4.8, 5] | 0.860 |
| | | medium | [2, 4.3) | [4.3, 5] | 0.938 |
| | | hard | [2, 4.1) | [4.1, 5] | 1.030 |
| Electric Motor | Rotor slot diameter 3 $d_{r3}$ (mm) | easy | [100, 122) | [122, 126] | 0.762 |
| | | medium | [99, 120) | [120, 126] | 0.932 |
| | | hard | [99, 118) | [118, 126] | 0.955 |
| Heatsink | # fins | easy | [5, 13) | [13, 14] | 1.446 |
| | | medium | [5, 12) | [12, 15] | 1.683 |
| | | hard | [5, 11) | [11, 15] | 1.861 |

## 4 BENCHMARK SETUP

This section outlines the learning problem (Section 4.1), the UDA algorithms considered (Section 4.2), the unsupervised model selection strategies (Section 4.3), and the baseline models used (Section 4.4). Finally, we describe the experimental setup and evaluation metrics in Section 4.5.

### 4.1 LEARNING PROBLEM

Let $\mathcal{X}$ be an input space containing geometries and conditioning parameters (e.g., thickness and temperatures in Figure 2a) and $\mathcal{Y}$ be an output space containing ground truth solution fields, obtained from a numerical solver (e.g., PEEQ field in Figure 2b). Following (Ben-David et al., 2010), a *domain* is represented by a probability density function $p$ on $\mathcal{X} \times \mathcal{Y}$ (e.g., describing the probability of observing an input-output pair corresponding to the parameter range $r \in [0.01, 0.115)$ in Table 2). UDA has been formulated as follows: Given a source dataset $(x_1, y_1), ..., (x_n, y_n)$ drawn from a source domain $p_S$ together with an *unlabeled* target dataset $x'_1, ..., x'_m$ drawn from the ($\mathcal{X}$-marginal) of a target domain $p_T$, the problem is to find a model $f : \mathcal{X} \to \mathcal{Y}$ that has small expected risk on the target domain:

$$\mathbb{E}_{(x,y)\sim p_T}[\ell(f(x), y)]\,, \tag{1}$$

with $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ being some loss function. For example, consider the square loss $\ell(f(x), y) = (f(x)-y)^2$. In our setup $f(x) = g(x, \phi(x))$ is composed of a conditioning network $\phi$ and a surrogate $g$ (see Figure 1).

### 4.2 UNSUPERVISED DOMAIN ADAPTATION ALGORITHMS

Our UDA baseline algorithms are from the class of *domain-invariant representation learning* methods. These methods are strong baselines, in the sense that their performance typically lies within the standard deviation of the winning algorithms in large scale empirical evaluations (i.e., no significant outperformance is observed), see CMD, Deep CORAL and DANN in (Dinu et al., 2023, Tables 12–14), M3SDA in (Peng et al., 2019), MMDA and HoMM in (Ragab et al., 2022).

Following Johansson et al. (2019) and Zellinger et al. (2021b), we express the objective of domain-invariant learning using two learning models: a *representation* mapping $\phi \in \Phi \subset \{\phi : \mathcal{X} \to \mathcal{R}\}$, which in our case corresponds to the conditioning network that maps simulation parameters into some representation space $\mathcal{R} \subset \mathbb{R}^k$ and a *regressor* $g \in \mathcal{G} \subset \{g : \mathcal{X} \times \mathcal{R} \to \mathcal{Y}\}$, which is realized by a neural surrogate. The goal is to find a mapping $\phi$ under which the source representations $\phi(\mathbf{x}) := (\phi(x_1), \ldots, \phi(x_n))$ and the target representations $\phi(\mathbf{x}') := (\phi(x'_1), \ldots, \phi(x'_m))$ appear similar, and, at the same time, enough information is preserved for prediction by $g$, see (Quionero-Candela et al., 2009). This is realized by estimating objectives of the form

$$\min_{g\in\mathcal{G},\,\phi\in\Phi}\ \underbrace{\mathbb{E}_{(x,y)\sim p_S}\left[\ell\big(g(x,\phi(x)),y\big)\right]}_{\mathcal{L}_{\text{recon}}} + \lambda\cdot\underbrace{d\big(\phi(\mathbf{x}),\phi(\mathbf{x}')\big)}_{\mathcal{L}_{\text{DA}}}\ . \tag{2}$$

The training objective therefore consists of minimizing both terms: the supervised reconstruction loss $\mathcal{L}_{\text{recon}}$ and the domain adaptation loss $\mathcal{L}_{\text{DA}}$ as shown in Figure 1. A variety of UDA algorithms correspond to different implementations of the distance $d$. Good choices for $d$ in Equation (2) have been found to be the Wasserstein distance (Courty et al., 2017), the Maximum Mean Discrepancy (Baktashmotlagh et al., 2013), moment distances (Sun & Saenko, 2016; Zellinger et al., 2019), adversarially learned distances (Ganin et al., 2015) and other divergence measures (Johansson et al., 2019; Zhang et al., 2019). We outline the distance measures of all included algorithms in Appendix D. Furthermore, appropriately choosing the regularization parameter $\lambda$ is crucial for performance (Musgrave et al., 2021; Dinu et al., 2023; Yang et al., 2024), making model selection necessary.

### 4.3 UNSUPERVISED MODEL SELECTION STRATEGIES

Among all algorithm design choices in UDA, model selection has been repeatedly recognized as one of the most crucial (Musgrave et al., 2021; Yang et al., 2024), with sub-optimal choices potentially leading to *negative transfer* (Pan & Yang, 2010). However, classical approaches (e.g., validation set, cross-validation, information criterion) cannot be used due to missing labels and distribution shifts. It is therefore a natural benchmark requirement for UDA to provide also unified model selection strategies in addition to UDA algorithms.

In this work, we rely on Importance Weighted Validation (IWV) (Sugiyama et al., 2007a) and Deep Embedded Validation (DEV) (You et al., 2019) to overcome the two challenges: (i) distribution shift and (ii) missing target labels. These methods rely on the Radon-Nikodým derivative and the covariate shift assumption $p_S(y|x) = p_T(y|x)$ to obtain

$$\mathbb{E}_{(x,y)\sim p_T}[\ell(f(x), y)] = \mathbb{E}_{(x,y)\sim p_S}\left[\frac{p_T(x)\cancel{p_T(y|x)}}{p_S(x)\cancel{p_S(y|x)}}\ell(f(x), y)\right] = \mathbb{E}_{(x,y)\sim p_S}[\beta(x)\ell(f(x), y)]\,. \tag{3}$$

Equation (3) motivates to estimate the target error by a two step procedure: First, approaching challenge (i) by estimating the density ratio $\beta(x) = \frac{p_T(x)}{p_S(x)}$ from the input data only, and, approaching challenge (ii) by estimating target error by the weighted source error using *labeled* source data.

### 4.4 BASELINE MODELS

We provide a comprehensive range of machine learning methods, adapted to our conditioned simulation task, organized by their capacity to model interactions across different spatial scales:

*Global context models* such as PointNet (Qi et al., 2017) incorporate global information into local Multi-Layer Perceptrons (MLPs) by summarizing features of all input points by aggregation into a global representation, which is then shared among nodes. Recognizing the necessity of *local information* when dealing with complex meshes and structures, we include GraphSAGE (Hamilton et al., 2017), a proven Graph Neural Network (GNN) architecture (Scarselli et al., 2009; Battaglia et al., 2018) already used in other mesh based tasks (Pfaff et al., 2020; Bonnet et al., 2022). However, large scale applications of GNNs are challenging due to computational expense (Alkin et al., 2024a) and issues like oversmoothing (Rusch et al., 2023). Finally, to overcome these limitations, we employ *attention based models* (Vaswani et al., 2017). These models typically scale better with the number of points, and integrate both global and local information enabling stronger long-range interactions and greater expressivity. We include Transolver (Wu et al., 2024), a modern neural operator Transformer.

As an alternative categorization, baselines can also be classified by input-output pairings into *point-to-point* and *latent* approaches. The former explicitly encodes nodes, while the latter represents the underlying fields in a latent space and requires queries to retrieve nodes. While all previously mentioned models are *point-to-point*, we also include Universal Physics Transformer (UPT) (Alkin et al., 2024a; Fürst et al., 2025) and Geometry-Informed Neural Operator (GINO) Li et al. (2023b), as examples of latent field methods. Both methods are designed for large problems and offer favorable scaling on big meshes through latent field modeling. The main difference is that GINO latent space is constrained to a regular grid, where it operates in the *frequency* domain. UPT, in contrast, learns in a standard unconstrained latent domain. Both UPT and GINO are designed for large scale meshes, and therefore we benchmark them on the *heatsink design* dataset.

We provide detailed explanations of all implemented architectures in Appendix E. Our framework explicitly conditions neural operators on configuration parameters. We first embed them using a sinusoidal (sin–cos) encoding and a shallow MLP $\phi$ to produce a latent representation and then condition the neural operator $g$ by using either concatenation of the latent conditioning vector, FiLM (Perez et al., 2018) or DiT conditioning layers (Peebles & Xie, 2023). As an alternative, we also evaluate replacing $\phi$ with a geometric mesh encoder that derives the latent representation directly from the input geometry. On the *electric motor design* dataset, this variant performs worse (see Appendix H.1), supporting our design choice.

### 4.5 EXPERIMENTS AND EVALUATION

**Experimental Setup.** We benchmark four prominent UDA algorithms (Deep Coral (Sun & Saenko, 2016), CMD (Zellinger et al., 2019), DANN (Ganin et al., 2015) and DARE-GRAM (Nejjar et al., 2023)) in combination with the following four unsupervised model selection strategies: IWV (Sugiyama et al., 2007a), DEV (You et al., 2019), Source Best (SB) (selecting models based on source domain validation performance) and Target Best (TB) (selecting models based on target simulation data, which is not available in UDA but serves as a lower bound for perfect model selection).

For the baseline neural surrogate models, we evaluate PointNet, GraphSAGE, and Transolver on the *hot rolling*, *sheet metal forming*, and *electric motor design* datasets. Due to memory and runtime constraints on the large scale *heatsink design* dataset, we omit GraphSAGE and instead benchmark UPT and GINO alongside PointNet and Transolver.

**Experimental Scale.** We perform an extensive sweep over the critical UDA parameter $\lambda$ and average across four seeds, resulting in a total of $\mathbf{1{,}664}$ training runs (see Table 12). Details on architectures, hyperparameters, training setup and normalization, as well as a breakdown of training times are included in Appendices E and F.

**Evaluation Metrics.** For each dataset, we report the Normalized Root Mean Squared Error (NRMSE) averaged over all output fields, as well as the per field Root Mean Squared Error (RMSE) values (computed on denormalized data), the Euclidean error for deformation predictions and the custom error metrics described in Sections 3.1 to 3.4. Additionally we provide physics-based evaluation metrics for all datasets. These metrics are tailored to the underlying PDEs. Detailed metric definitions are provided in Appendix F.2.

## 5 BENCHMARKING RESULTS

Table 3 overviews our benchmarking results, showing the best UDA and selection combination per model. Across datasets and architectures, UDA applied together with unsupervised model selection generally leads to a target error reduction, measured by NRMSE averaged across all fields. However, when examining the dataset-specific custom metrics introduced in Sections 3.1 to 3.4, the individual fields, and the physics-based metrics, gains are not uniform, and some methods improve the global loss, while performance on particular metrics. This pattern suggests that standard methods are a good starting point, but specialized algorithms tailored to high-dimensional regression tasks are needed. Furthermore, the gap between the best UDA + selection configurations and the TB oracle (lower bound on error) indicates that current unsupervised model selection strategies also leave room for improvement. Despite the clear benefits of UDA, no single UDA algorithm or unsupervised selection strategy dominates across all datasets. In addition to this summary, we report full source and target metrics across architectures, algorithms, and selection strategies in Tables 4 to 10.

Finally, since the presented tables only report performance on the *medium* difficulty setting, we additionally visualize model behavior of the best performing combination (model + UDA algorithm + selection strategy) across all difficulty levels of the *hot rolling* dataset in Figure 6. It illustrates the increase in prediction error as the domain gap widens and highlights the consistent improvements achieved by applying UDA algorithms combined with unsupervised model selection strategies on the *easy* and *medium* settings.

For the *hard* setting, however, the shown unsupervised model selection algorithm fails to identify suitable models, as the mean error matches that of the unregularized baselines with the standard deviation even increasing. Nonetheless, the theoretical lower bound (TB) remains substantially

Table 3: Best performing combination of UDA algorithm and unsupervised model selection for each dataset (*medium* difficulty) and architecture. We also report an oracle with target best (TB) selection, which provides a lower bound on the selection error. Entries show the target domain (N)RMSE or physics-based metric. Promotion on the unregularized baseline are shown in parentheses, with improvement indicated as negative values and asterisks marking unstable unregularized baselines. For each dataset, the best configuration (green) is chosen by the lowest NRMSE across all fields (bold).

| **Dataset** | **Model** | **Best UDA Method + Model Selection** | **All Fields Normalized Avg (-)** | **Mises Stress (MPa)** | **Rel Custom Error (-)** | **VM Consistency (-)** |
|---|---|---|---|---|---|---|
| Rolling | GraphSAGE | DARE-GRAM + IWV | **0.192 (-0.172)** | **12.384 (-7.406)** | **0.142 (-0.092)** | **0.049 (+0.003)** |
| | PointNet | CMD + SB | 0.387 (-0.082) | 27.922 (+0.311) | 0.261 (-0.009) | 0.055 (-0.001) |
| | Transolver | CMD + SB | 0.781 (⋆) | 71.526 (⋆) | 0.507 (⋆) | 0.086 (⋆) |
| | Oracle (GraphSAGE) | DARE-GRAM + TB | 0.192 (-0.172) | 12.384 (-7.406) | 0.142 (-0.092) | 0.049 (+0.003) |
| Motor | GraphSAGE | DARE-GRAM + SB | 0.342 (-0.033) | 29.088 (-0.370) | 0.349 (-0.078) | 0.031 (-0.000) |
| | PointNet | Deep Coral + SB | 0.313 (-0.084) | 26.229 (-4.425) | 0.197 (-0.147) | 0.043 (-0.003) |
| | Transolver | Deep Coral + SB | **0.098 (-0.018)** | **7.269 (-0.729)** | **0.089 (-0.022)** | **0.016 (-0.003)** |
| | Oracle (Transolver) | Deep Coral + TB | 0.098 (-0.018) | 7.266 (-0.732) | 0.089 (-0.022) | 0.016 (-0.002) |

| **Dataset** | **Model** | **Best UDA Method + Model Selection** | **All Fields Normalized Avg (-)** | **Mises Stress (MPa)** | **Rel Custom Error (-)** | **Plastic Residual (-)** |
|---|---|---|---|---|---|---|
| Forming | GraphSAGE | DANN + IWV | 0.334 (-0.042) | 52.917 (+6.821) | 5.384 (+2.888) | 0.509 (+0.028) |
| | PointNet | Deep Coral + SB | 0.182 (-0.044) | 31.345 (-0.090) | 1.154 (+0.273) | **0.451 (-0.029)** |
| | Transolver | Deep Coral + DEV | **0.154 (-0.014)** | **24.427 (+1.457)** | **0.806 (+0.199)** | 0.581 (+0.098) |
| | Oracle (Transolver) | CMD + TB | 0.131 (-0.037) | 20.275 (-2.695) | 0.796 (+0.189) | 0.506 (+0.022) |

| **Dataset** | **Model** | **Best UDA Method + Model Selection** | **All Fields Normalized Avg (-)** | **Temperature (K)** | **Rel Custom Error (-)** | **BC Violation Velocity (m/s)** |
|---|---|---|---|---|---|---|
| Heatsink | PointNet | DARE-GRAM + SB | 0.371 (-0.197) | 12.343 (-8.783) | 0.015 (-0.035) | 0.121 (+0.011) |
| | Transolver | Deep Coral + DEV | **0.318 (-0.128)** | **9.081 (-0.639)** | **0.009 (-0.001)** | 0.117 (+0.039) |
| | UPT | Deep Coral + SB | 0.325 (-0.116) | 12.414 (-0.619) | 0.013 (-0.000) | 0.107 (+0.036) |
| | GINO | Deep CORAL + SB | 0.356 (-0.128) | 14.031 (+0.136) | 0.017 (+0.000) | **0.107 (+0.024)** |
| | Oracle (Transolver) | Deep Coral + TB | 0.310 (-0.135) | 8.718 (-1.002) | 0.009 (-0.001) | 0.117 (+0.039) |

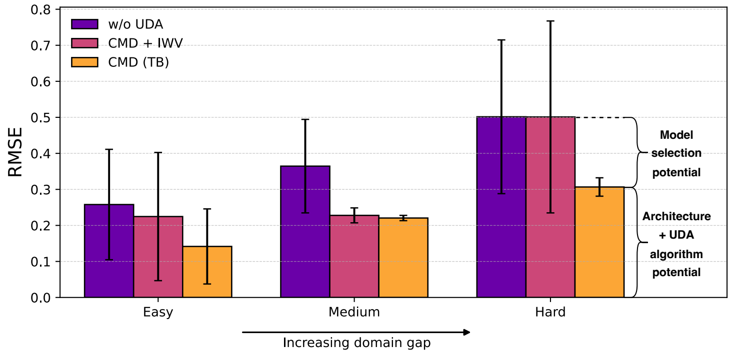


Figure 6: Target error scaling with increasing domain gap. We show the averaged RMSE across all (normalized) fields for the *easy*, *medium*, and *hard* gaps on the *hot rolling* task. We compare models without UDA, the best performing UDA method with unsupervised model selection (CMD + IWV), and the theoretical lower bound (TB). Error bars indicate the standard deviation across four seeds. Furthermore, we highlight potentials of architecture, algorithm and selection improvements on the *hard* task.

below the unregularized error. Figure 6 again highlights the two promising directions for further improvement of the presented baselines: (i) enhancement of neural surrogate architectures and UDA algorithms, and (ii) especially, improvement of unsupervised model selection strategies.

## 6 DISCUSSION

We presented SIMSHIFT, a collection of industry relevant datasets paired with a benchmarking library for comparing UDA algorithms, unsupervised model selection strategies and neural surrogates in real world scenarios. We adapted available techniques, applied them on physical simulation data and performed extensive experiments to evaluate their performance on the presented datasets. Our findings suggest that standard UDA training methods can improve performance of models in unseen parameter ranges in physical simulations, with improvement margins in line with those seen in UDA literature (Dinu et al., 2023; Ragab et al., 2022). Additionally, we find correct unsupervised model selection to be extremely important in downstream model performance on target domains, with it arguably having as much impact as the UDA training itself, which is also in agreement with other DA works (Musgrave et al., 2021).

**Limitations.** We acknowledge that our datasets are limited under two main aspects: (i) They only cover *steady-state* problems, which represent a large portion of industrial simulation tasks. However, an extension with *time-dependent* datasets could be valuable for certain application areas. (ii) They cover a wide range of mesh sizes, ranging from roughly $\mathcal{O}(10^2)$ up to $\mathcal{O}(10^6)$ nodes. Nevertheless, many industrial scenarios require substantially larger meshes. These limitations reflect design choices aimed at benchmarking clarity and computational feasibility and leave room for future extensions.

**Future Directions.** Motivated by our results, we identify several promising research directions: (i) Although we include a diverse and competitive set of UDA algorithms and unsupervised model selection techniques, a wide range of methods remain unexplored in the context of scientific ML. Examples include ensembling based adaptation (Cha et al., 2021), adversarial information bottleneck approaches (Luo et al., 2019; Song et al., 2020) or diffusion based methods (Peng et al., 2024; Liao et al., 2025). In addition, test-time adaptation methods (Wang et al., 2021; Adachi et al., 2025) could be designed and tested using our benchmark. (ii) SIMSHIFT currently evaluates standard UDA algorithms and does not integrate physics constraints (Karniadakis et al., 2021) into training. Our framework and datasets allows to include physics constraints, and we find the direction of a specific physics-inspired UDA method a very interesting and potentially fruitful gap in the current research.

## REPRODUCIBILITY STATEMENT

The first step towards reproducibility are the datasets. We provide a download link for all dataset with our predefined domain shifts and a high–level description of each dataset and the splitting strategy in Section 3 and Appendix C. To take transparency a step further, we present detailed descriptions of the respective numerical simulations together with the employed solvers, initial/boundary conditions and the configuration details in Appendix G. Concerning the machine learning parts of the benchmark, we provide detailed descriptions of the architectures, their configurations and training hyperparameters in Appendix F.1. Additionally, we provide an anonymized codebase with pinned dependencies, fixable seeds and a comprehensive `README.md` along with all necessary configuration files used in our benchmarking pipeline.

## REFERENCES

Muhammad Ali Ablat and Ala Qattawi. Numerical simulation of sheet metal forming: a review. *The international journal of advanced manufacturing technology*, 89:1235–1250, 2017.

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pp. 1–3, 2024.

Kazuki Adachi, Shin'ya Yamaguchi, Atsutoshi Kumagai, and Tomoki Hamagami. Test-time adaptation for regression by subspace alignment. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL `https://openreview.net/forum?id=SXtl7NRyE5`.

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., 2024a. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/2cd36d327f33d47b372d4711edd08de0-Paper-Conference.pdf`.

Benedikt Alkin, Tobias Kronlachner, Samuele Papa, Stefan Pirker, Thomas Lichtenegger, and Johannes Brandstetter. Neuraldem - real-time simulation of industrial particulate flows. *CoRR*, abs/2411.09678, 2024b. doi: 10.48550/ARXIV.2411.09678. URL `https://doi.org/10.48550/arXiv.2411.09678`.

Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *CoRR*, abs/1907.02893, 2019. URL `http://arxiv.org/abs/1907.02893`.

R. Arularasan and R. Velraj. Modeling and simulation of a parallel plate heat sink using computational fluid dynamics. *The International Journal of Advanced Manufacturing Technology*, 51(1):415–419, 2010. ISSN 1433-3015. doi: 10.1007/s00170-008-1867-9. URL `https://doi.org/10.1007/s00170-008-1867-9`.

Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 769–776, 2013.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

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

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, 2010. doi: 10.1007/S10994-009-5152-4. URL `https://doi.org/10.1007/s10994-009-5152-4`.

Maurits Bleeker, Matthias Dorfer, Tobias Kronlachner, Reinhard Sonnleitner, Benedikt Alkin, and Johannes Brandstetter. Neuralcfd: Deep learning on high-fidelity automotive aerodynamics simulations. *CoRR*, abs/2502.09692, 2025. doi: 10.48550/ARXIV.2502.09692. URL `https://doi.org/10.48550/arXiv.2502.09692`.

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In Annie Zaenen and Antal van den Bosch (eds.), *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 440–447, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `https://aclanthology.org/P07-1056/`.

Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, Megan Stanley, Anna Allen, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan A. Weyn, Haiyu Dong, Jayesh K. Gupta, Kit Thambiratnam, Alexander T. Archibald, Chun-Chieh Wu, Elizabeth Heider, Max Welling, Richard E. Turner, and Paris Perdikaris. A foundation model for the earth system. *Nat.*, 641(8065):1180–1187, 2025. doi: 10.1038/S41586-025-09005-Y. URL `https://doi.org/10.1038/s41586-025-09005-y`.

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`.

Johannes Brandstetter. Envisioning better benchmarks for machine learning pde solvers. *Nature Machine Intelligence*, 7:2–3, January 2025. doi: 10.1038/s42256-024-00962-z. URL `https://doi.org/10.1038/s42256-024-00962-z`. Published online: 13 December 2024.

Steven L. Brunton and J. Nathan Kutz. Machine learning for partial differential equations: Data-driven discovery, model reduction, and control. *Journal of Computational Dynamics*, 7(2):343–360, 2020. doi: 10.3934/jcd.2020015.

Timothy A Burress, Steven L Campbell, Chester Coomer, Curtis William Ayers, Andrew A Wereszczak, Joseph Philip Cunningham, Laura D Marlino, Larry Eugene Seiber, and Hua-Tay Lin. Evaluation of the 2010 toyota prius hybrid synergy drive system. Technical report, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States). Power . . . , 2011.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. SWAD: domain generalization by seeking flat minima. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 22405–22418, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/bcb41ccdc4363c6848a1d760f26c28a0-Abstract.html`.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. In *Advances in neural information processing systems*, volume 9, pp. 705–712, 1996.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017.

Marius-Constantin Dinu, Markus Holzleitner, Maximilian Beck, Hoan Duc Nguyen, Andrea Huber, Hamid Eghbal-zadeh, Bernhard Alois Moser, Sergei V. Pereverzyev, Sepp Hochreiter, and Werner Zellinger. Addressing parameter choice issues in unsupervised domain adaptation by aggregation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/forum?id=M95oDwJXayG`.

Alexander Dorninger, Simon Weitzhofer, Markus Schörgenhumer, Albert Sorgdrager, and Eike Janssen. Automated mechanical rotor design assessment based on 2d fea results. In *2021 11th International Electric Drives Production Conference (EDPC)*, pp. 1–8, 2021. doi: 10.1109/EDPC53547.2021.9684221.

Laurent Dumas. Chapter 1 cfd-based optimization for automotive aerodynamics. 2007. URL `https://api.semanticscholar.org/CorpusID:601424`.

Mohamed Elrefaie, Angela Dai, and Faez Ahmed. Drivaernet: A parametric car dataset for data-driven aerodynamic design and graph-based drag prediction. volume Volume 3A: 50th Design Automation Conference (DAC) of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. V03AT03A019. Curran Associates, Inc., 08 2024a. URL `https://doi.org/10.1115/DETC2024-143593`.

Mohamed Elrefaie, Florin Morar, Angela Dai, and Faez Ahmed. Drivaernet++: A large-scale multimodal car dataset with computational fluid dynamics simulations and deep learning benchmarks. 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. 499–536. Curran Associates, Inc., 2024b. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/013cf29a9e68e4411d0593040a8a1eb3-Paper-Datasets_and_Benchmarks_Track.pdf`.

European Union. Artificial intelligence act (eu regulation 2024/1689), article 15: Accuracy, robustness and cybersecurity. `http://data.europa.eu/eli/reg/2024/1689/oj`, 2024. OJ L, 2024/1689, 12 July 2024.

Lawrence C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, 2nd edition, 2010.

Luis Fernando Folle, Tiago Nunes Lima, Matheus Passos Sarmento Santos, Bruna Callegari, Bruno Caetano dos Santos Silva, Luiz Gustavo Souza Zamorano, and Rodrigo Santiago Coelho. A review on sheet metal forming behavior in high-strength steels and the use of numerical simulations. *Metals*, 14(12), 2024. ISSN 2075-4701. doi: 10.3390/met14121428. URL `https://www.mdpi.com/2075-4701/14/12/1428`.

Nicola Rares Franco, Andrea Manzoni, and Paolo Zunino. Mesh-informed neural networks for operator learning in finite element spaces. *Journal of Scientific Computing*, 97, 2022. URL `https://api.semanticscholar.org/CorpusID:258547333`.

Andreas Fürst, Florian Sestak, Artur P. Toshev, Benedikt Alkin, Nikolaus A. Adams, Andreas Mayr, Günter Klambauer, and Johannes Brandstetter. UPT++: Latent point set neural operators for modeling system state transitions. In *ICLR 2025 Workshop on Machine Learning Multiscale Processes*, 2025. URL `https://openreview.net/forum?id=u3sEGcYyqw`.

L.M. Galantucci and L. Tricarico. Thermo-mechanical simulation of a rolling process with an fem approach. *Journal of Materials Processing Technology*, 92-93:494–501, 1999. ISSN 0924-0136. doi: https://doi.org/10.1016/S0924-0136(99)00242-3. URL `https://www.sciencedirect.com/science/article/pii/S0924013699002423`.

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

Josh Gardner, Zoran Popovic, and Ludwig Schmidt. Benchmarking distribution shift in tabular data with tableshift. 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. URL `http://papers.nips.cc/paper_files/paper/2023/hash/a76a757ed479a1e6a5f8134bea492f83-Abstract-Datasets_and_Benchmarks.html`.

M.E. Gerlach, M. Zajonc, and B. Ponick. Mechanical stress and deformation in the rotors of high-speed pmsm and im. *Elektrotechnik & Informationstechnik*, 138(2):96–109, 2021. doi: 10.1007/s00502-021-00866-5. URL `https://doi.org/10.1007/s00502-021-00866-5`.

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`.

Jayesh K Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized pde modeling. *arXiv preprint arXiv:2209.15616*, 2022.

N.K. Gupta. *Steel Rolling: Principle, Process & Application*. CRC Press, 2021. ISBN 9781032022161. URL `https://books.google.at/books?id=wlEgzgEACAAJ`.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bezenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=JC1VKK3UXk`.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=fILj7WpI-g`.

Seo Yeon Jo, Seojun Hong, Heung Nam Han, and Myoung-Gyu Lee. Modeling and simulation of steel rolling with microstructure evolution: An overview. *steel research international*, 94(2):2200260, 2023. doi: https://doi.org/10.1002/srin.202200260. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/srin.202200260`.

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.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00314-5.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. *CoRR*, abs/2012.07421, 2020. URL `https://arxiv.org/abs/2012.07421`.

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`.

Donald R Lesuer. Experimental investigations of material models for ti-6al-4v titanium and 2024-t3 aluminum. *DOT/FAA/R-00/25*, 2000.

Zongyi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *CoRR*, abs/2010.08895, 2020a. URL `https://arxiv.org/abs/2010.08895`.

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, 2020b.

Zongyi Li, Nikola Borislavov Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Prakash Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Anima Anandkumar. Geometry-informed neural operator for large-scale 3d PDEs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL `https://openreview.net/forum?id=86dXbqT5Ua`.

Zongyi Li, Nikola Borislavov Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Prakash Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Anima Anandkumar. Geometry-informed neural operator for large-scale 3d pdes, 2023b. URL `https://arxiv.org/abs/2309.00583`.

Kang Liao, Zongsheng Yue, Zhouxia Wang, and Chen Change Loy. Denoising as adaptation: Noise-space domain adaptation for image restoration. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL `https://openreview.net/forum?id=jsBhmOCKYs`.

Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. Pde-refiner: Achieving accurate long rollouts with neural PDE solvers. 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. URL `http://papers.nips.cc/paper_files/paper/2023/hash/d529b943af3dba734f8a7d49efcb6d09-Abstract-Conference.html`.

Q. Liu and H. Xue. Adversarial spectral kernel matching for unsupervised time series domain adaptation. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 30, 2021.

M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

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`.

Xufei Lu, Xin Lin, Michele Chiumenti, Miguel Cervera, JunJie Li, Liang Ma, Lei Wei, Yunlong Hu, and Weidong Huang. Finite element analysis and experimental validation of the thermomechanical behavior in laser solid forming of ti-6al-4v. *Additive Manufacturing*, 21:30–40, 2018.

Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. Significance-aware information bottleneck for domain adaptive semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 6777–6786. IEEE, 2019. doi: 10.1109/ICCV.2019.00688. URL `https://doi.org/10.1109/ICCV.2019.00688`.

Yining Luo, Yingfa Chen, and Zhen Zhang. Cfdbench: A comprehensive benchmark for machine learning methods in fluid dynamics. *CoRR*, abs/2310.05963, 2023. doi: 10.48550/ARXIV.2310.05963. URL `https://doi.org/10.48550/arXiv.2310.05963`.

P. Majumdar. *Simulation of Thermal Systems*, chapter 9, pp. 528–561. John Wiley Sons, Ltd, 2021. ISBN 9781118956922. doi: https://doi.org/10.1002/9781118956922.ch9. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118956922.ch9`.

Joaquim R.R.A. Martins. Aerodynamic design optimization: Challenges and perspectives. *Computers Fluids*, 239:105391, 2022. ISSN 0045-7930. doi: https://doi.org/10.1016/j.compfluid.2022.105391. URL `https://www.sciencedirect.com/science/article/pii/S0045793022000615`.

Florian Menter, M. Kuntz, and RB Langtry. Ten years of industrial experience with the sst turbulence model. *Heat and Mass Transfer*, 4, 01 2003.

Amil Merchant, Simon L. Batzner, Samuel S. Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nat.*, 624(7990):80–85, 2023. doi: 10.1038/S41586-023-06735-9.

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`.

Kevin Musgrave, Serge J. Belongie, and Ser-Nam Lim. Unsupervised domain adaptation: A reality check. *CoRR*, abs/2111.15672, 2021. URL `https://arxiv.org/abs/2111.15672`.

Ismail Nejjar, Qin Wang, and Olga Fink. DARE-GRAM : Unsupervised domain adaptation regression by aligning inverse gram matrices. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 11744–11754. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01130. URL `https://doi.org/10.1109/CVPR52729.2023.01130`.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam H. Parker, Miles Cranmer, and Shirley Ho. The well: a large-scale collection of diverse physics simulations for machine learning. 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. 44989–45037. Curran Associates, Inc., 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/4f9a5acd91ac76569f2fe291b1f4772b-Paper-Datasets_and_Benchmarks_Track.pdf`.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *CoRR*, abs/2202.11214, 2022. URL `https://arxiv.org/abs/2202.11214`.

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`.

Duo Peng, Qiuhong Ke, Arulmurugan Ambikapathi, Yasin Yazici, Yinjie Lei, and Jun Liu. Unsupervised domain adaptation via domain-adaptive diffusion. *IEEE Trans. Image Process.*, 33: 4245–4260, 2024. doi: 10.1109/TIP.2024.3424985. URL `https://doi.org/10.1109/TIP.2024.3424985`.

Xingchao Peng, Ben Usman, Kuniaki Saito, Neela Kaushik, Judy Hoffman, and Kate Saenko. Syn2real: A new benchmark forsynthetic-to-real visual domain adaptation. *CoRR*, abs/1806.09755, 2018.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pp. 3942–3951, 2018. URL `https://arxiv.org/abs/1709.07871`.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. *CoRR*, abs/2010.03409, 2020. URL `http://dblp.uni-trier.de/db/journals/corr/corr2010.html#abs-2010-03409`.

Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 77–85. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.16. URL `https://doi.org/10.1109/CVPR.2017.16`.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009. ISBN 0262170051.

Mohamed Ragab, Emadeldeen Eldele, Wee Ling Tan, Chuan-Sheng Foo, Zhenghua Chen, Min Wu, Chee Keong Kwoh, and Xiaoli Li. ADATIME: A benchmarking suite for domain adaptation on time series data. *CoRR*, abs/2203.08321, 2022. doi: 10.48550/ARXIV.2203.08321. URL `https://doi.org/10.48550/arXiv.2203.08321`.

Md Atiqur Rahman, S. M. Mozammil Hasnain, Prabhu Paramasivam, and Abinet Gosaye Ayanie. Advancing thermal management in electronics: a review of innovative heat sink designs and optimization techniques. *RSC Adv.*, 14:31291–31319, 2024. doi: 10.1039/D4RA05845C. URL `http://dx.doi.org/10.1039/D4RA05845C`.

Zheng Ren, Yongxin Yang, Bingbing Chen, Yaqing Li, Chengzhong Xu, Timothy M Hospedales, and Tao Wang. A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54(9):1–36, 2021.

T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the International Conference on Machine Learning*, pp. 3076–3085. PMLR, 2017.

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`.

Yuxuan Song, Lantao Yu, Zhangjie Cao, Zhiming Zhou, Jian Shen, Shuo Shao, Weinan Zhang, and Yong Yu. Improving unsupervised domain adaptation with variational information bottleneck. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang (eds.), *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pp. 1499–1506. IOS Press, 2020. doi: 10.3233/FAIA200257. URL `https://doi.org/10.3233/FAIA200257`.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *J. Mach. Learn. Res.*, 8:985–1005, 2007a. doi: 10.5555/1314498.1390324. URL `https://dl.acm.org/doi/10.5555/1314498.1390324`.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis (eds.), *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pp. 1433–1440. Curran Associates, Inc., 2007b. URL `https://proceedings.neurips.cc/paper/2007/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract.html`.

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

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 1596–1611. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/0a9747136d411fb83f0cf81820d44afb-Paper-Datasets_and_Benchmarks.pdf`.

A.Erman Tekkaya. State-of-the-art of simulation of sheet metal forming. *Journal of Materials Processing Technology*, 103(1):14–22, 2000. ISSN 0924-0136. doi: https://doi.org/10.1016/S0924-0136(00)00413-1. URL `https://www.sciencedirect.com/science/article/pii/S0924013600004131`.

David Tellez, Geert Litjens, Péter Bándi, Wouter Bulten, John-Melle Bokhorst, Francesco Ciompi, and Jeroen van der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical Image Analysis*, 58: 101544, 2019. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2019.101544. URL `https://www.sciencedirect.com/science/article/pii/S1361841519300799`.

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.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7167–7176, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. *CoRR*, abs/1706.07522, 2017. URL `http://arxiv.org/abs/1706.07522`.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=uXl3bZLkr3c`.

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

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.

Jianfei Yang, Hanjie Qian, Yuecong Xu, Kai Wang, and Lihua Xie. Can we evaluate domain adaptation models without target-domain labels? In *International Conference on Learning Representations*, 2024.

Kaichao You, Ximei Wang, Mingsheng Long, and Michael I. Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7124–7133. PMLR, 2019. URL `http://proceedings.mlr.press/v97/you19a.html`.

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`.

Werner Zellinger, Bernhard A Moser, and Susanne Saminger-Platz. On generalization in moment-based domain adaptation. *Annals of Mathematics and Artificial Intelligence*, 89(3):333–369, 2021a.

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 Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 20798–20811, 2021b. URL `https://proceedings.neurips.cc/paper/2021/hash/ae0909a324fb2530e205e52d40266418-Abstract.html`.

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, et al. A generative model for inorganic materials design. *Nature*, pp. 1–3, 2025.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *Proceedings of the International Conference on Machine Learning*, pp. 7404–7413, 2019.

Y. Zhu, F. Zhuang, J. Wang, G. Ke, J. Chen, J. Bian, H. Xiong, and Q. He. Deep subdomain adaptation network for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 32(4):1713–1722, 2021.

# LLM USAGE DISCLOSURE

In general, LLM tools were used to refine writing in parts of the paper. DeepSeek-R1 and GPT-5 were additionally used to make visualizations prettier, speed up the development of plotting functions, and dump experimental results neatly into latex tabled tables. Beyond that, they were not used to a significant degree in other parts of the code, as neither Copilot nor Cursor are used by the main author. AI assistants were strictly editors and decorators, i.e. they were not involved in ideation, reordering ideas, or at any higher or lower conceptual level.

# A ON NEURAL OPERATORS

One prominent approach in neural surrogate modeling for PDEs is operator learning (Kovachki et al., 2021; Li et al., 2020a; Lu et al., 2021; Alkin et al., 2024a; Li et al., 2020b). In this setting, an operator maps input functions, such as boundary or initial conditions, to the corresponding solution of the PDE. During training, neural operators typically learn from input-output pairs of discretized functions (Kovachki et al., 2021; Li et al., 2020a; Lu et al., 2021; Alkin et al., 2024a). While some methods expect regular, grid based inputs (Li et al., 2020a), others can be applied to any kind of data structure (Alkin et al., 2024a; Li et al., 2020b; 2023b). One notable property is *discretization invariance*, which, along with the ability to handle irregular data, enables generalization across different resolutions and mesh geometries. This is a highly desirable property for industrial simulations (Pfaff et al., 2020; Alkin et al., 2024a; Fürst et al., 2025; Li et al., 2023a; Franco et al., 2022), where non-uniform meshes are the standard due to the computational and modeling advantages. In this work, we focus on domain adaptation rather than benchmarking discretization invariance, and include neural surrogates that may not satisfy this property, such as (Hamilton et al., 2017). Such models have been leveraged in several large scale industrial contexts, including CFD for automotive (Bleeker et al., 2025) or Discrete Element Method (DEM) simulations for industrial processes (Alkin et al., 2024b).

# B DETAILED RESULTS

Complementing the summary in Table 3 of the main paper, the following sections present detailed results for each dataset. For every dataset, we present a complete empirical evaluation of our benchmark that compares the performance for all combinations of models, UDA algorithms and model selection strategies across all output fields.

While these quantitative metrics offer a high level summary of model performance, industry practitioners often need a more fine grained picture to assess the neural surrogate's capabilities under distribution shifts. To address this, we include additional analyses and visualizations alongside the quantitative results. First, we provide error distribution histograms to better illustrate the difficulty of the domain shift occurring in each dataset. Additionally, we present fringe and scatter plots comparing model predictions with the respective ground truth numerical solutions.

## B.1 HOT ROLLING

Table 4 presents the complete benchmarking results for the *hot rolling* dataset.

To gain more insights, we conduct additional analyses on the best performing model, selected based on having the lowest average normalized target domain error across all fields. Figure 7 shows the error distribution of this model and clearly highlights the substantial distribution shift between the source and target domain of the *hot rolling* dataset. Errors in the target domain are noticeably larger, almost up to an order of magnitude higher than those observed in the source domain.

To further illustrate the model's performance, we analyze two representative samples, one from the source and one from the target domain. Since the most critical field for downstream applications is PEEQ, we restrict the following analysis on this scalar field only.

Table 5 presents a summary of the absolute PEEQ prediction errors for the selected source and target samples. Additionally, Figure 8 and Figure 9 visualize the ground truth, predictions, and absolute errors for these samples using fringe plots.

Table 4: RMSE (mean ± std over 4 seeds) on the *hot rolling* dataset at *medium* difficulty. Values are target domain errors (lower is better). Bold marks the overall best model + UDA algorithm + model selection combination. For each architecture, the unregularized baseline row is shaded beige, whereas the best UDA + selection within that architecture is underlined and shaded green. Asterisks denote unstable runs ($\geq 10\times$ the column median).

| Model | DA Algorithm | Model Selection | All Fields Normalized Avg (-) | | Deformation (mm) | | Logarithmic Strain ($\times10^{-2}$) | | Equivalent Plastic Strain ($\times10^{-2}$) | | Mises Stress (MPa) | | Relative Custom Error (-) | | VM Consistency (-) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT |
| **GraphSAGE** | - | - | 0.016(±0.000) | 0.365(±0.130) | 0.525(±0.023) | 5.715(±1.567) | 0.018(±0.000) | 0.997(±0.377) | 0.033(±0.000) | 2.113(±0.789) | 1.972(±0.024) | 19.790(±7.186) | 0.041(±0.002) | 0.234(±0.081) | 0.059(±0.000) | 0.046(±0.002) |
| | DANN | DEV | 0.014(±0.000) | 1.171(±0.058) | 0.599(±0.051) | 17.369(±0.802) | 0.019(±0.001) | 3.451(±0.176) | 0.035(±0.001) | 7.296(±0.405) | 2.076(±0.055) | 110.648(±8.295) | 0.047(±0.003) | 0.730(±0.052) | 0.058(±0.001) | 0.080(±0.023) |
| | DANN | IWV | 0.014(±0.000) | 0.289(±0.147) | 0.561(±0.032) | 5.359(±1.848) | 0.018(±0.000) | 0.792(±0.186) | 0.033(±0.001) | 1.622(±0.306) | 1.992(±0.037) | 24.471(±22.423) | 0.042(±0.002) | 0.183(±0.045) | 0.059(±0.001) | 0.068(±0.042) |
| | DANN | SB | 0.014(±0.000) | 0.692(±0.511) | 0.573(±0.043) | 11.090(±7.161) | 0.018(±0.000) | 2.120(±1.506) | 0.034(±0.001) | 4.510(±3.201) | 1.991(±0.045) | 60.352(±51.358) | 0.043(±0.004) | 0.466(±0.318) | 0.058(±0.001) | 0.055(±0.009) |
| | DANN | TB | 0.014(±0.000) | 0.230(±0.041) | 0.604(±0.010) | 4.640(±0.593) | 0.018(±0.001) | 0.740(±0.134) | 0.034(±0.001) | 1.549(±0.275) | 2.017(±0.047) | 14.867(±3.085) | 0.042(±0.001) | 0.170(±0.026) | 0.059(±0.001) | 0.048(±0.002) |
| | CMD | DEV | 0.015(±0.001) | 1.328(±0.056) | 0.613(±0.045) | 17.690(±1.402) | 0.019(±0.002) | 3.584(±0.244) | 0.036(±0.004) | 7.358(±0.555) | 2.143(±0.175) | 130.494(±23.592) | 0.045(±0.005) | 0.779(±0.049) | 0.059(±0.001) | 0.241(±0.172) |
| | CMD | IWV | 0.014(±0.000) | 0.270(±0.082) | 0.581(±0.026) | 5.227(±1.179) | 0.018(±0.000) | 0.843(±0.195) | 0.033(±0.001) | 1.759(±0.381) | 1.979(±0.038) | 18.785(±8.449) | 0.043(±0.003) | 0.194(±0.043) | 0.058(±0.000) | 0.057(±0.020) |
| | CMD | SB | 0.014(±0.000) | 0.786(±0.535) | 0.571(±0.040) | 12.160(±7.174) | 0.018(±0.000) | 2.403(±1.541) | 0.033(±0.000) | 5.068(±3.248) | 1.974(±0.034) | 68.509(±55.017) | 0.041(±0.001) | 0.526(±0.327) | 0.058(±0.000) | 0.079(±0.045) |
| | CMD | TB | 0.014(±0.000) | 0.221(±0.007) | 0.583(±0.033) | 4.607(±0.261) | 0.018(±0.000) | 0.711(±0.014) | 0.033(±0.000) | 1.507(±0.045) | 1.992(±0.021) | 14.288(±1.040) | 0.044(±0.001) | 0.165(±0.003) | 0.058(±0.000) | 0.048(±0.001) |
| | DARE-GRAM | DEV | 0.014(±0.000) | 1.022(±0.168) | 0.538(±0.018) | 15.744(±2.302) | 0.019(±0.001) | 3.109(±0.500) | 0.035(±0.001) | 6.573(±1.071) | 2.065(±0.035) | 86.703(±21.972) | 0.045(±0.002) | 0.673(±0.116) | 0.059(±0.001) | 0.156(±0.051) |
| | **DARE-GRAM** | **IWV** | 0.014(±0.000) | **0.192(±0.029)** | 0.555(±0.026) | 4.082(±0.597) | 0.018(±0.000) | 0.616(±0.103) | 0.033(±0.000) | 1.297(±0.221) | 2.005(±0.018) | 12.384(±1.900) | 0.042(±0.000) | 0.142(±0.022) | 0.058(±0.001) | 0.049(±0.001) |
| | DARE-GRAM | SB | 0.014(±0.000) | 0.237(±0.044) | 0.477(±0.090) | 4.535(±0.809) | 0.018(±0.000) | 0.773(±0.139) | 0.033(±0.000) | 1.624(±0.307) | 1.985(±0.018) | 14.838(±3.624) | 0.041(±0.002) | 0.179(±0.030) | 0.059(±0.000) | 0.047(±0.002) |
| | DARE-GRAM | TB | 0.014(±0.000) | 0.192(±0.029) | 0.555(±0.026) | 4.082(±0.597) | 0.018(±0.000) | 0.616(±0.103) | 0.033(±0.000) | 1.297(±0.221) | 2.005(±0.018) | 12.384(±1.900) | 0.042(±0.000) | 0.142(±0.022) | 0.058(±0.001) | 0.049(±0.001) |
| | Deep Coral | DEV | 0.014(±0.000) | 0.723(±0.302) | 0.560(±0.058) | 11.013(±4.292) | 0.018(±0.000) | 2.202(±0.860) | 0.034(±0.000) | 4.601(±1.848) | 2.002(±0.016) | 61.162(±33.535) | 0.042(±0.002) | 0.490(±0.182) | 0.058(±0.001) | 0.073(±0.019) |
| | Deep Coral | IWV | 0.014(±0.000) | 0.273(±0.068) | 0.583(±0.054) | 5.137(±0.652) | 0.018(±0.000) | 0.841(±0.153) | 0.033(±0.000) | 1.774(±0.304) | 1.980(±0.014) | 19.965(±8.986) | 0.040(±0.001) | 0.203(±0.043) | 0.058(±0.001) | 0.052(±0.010) |
| | Deep Coral | SB | 0.014(±0.000) | 0.511(±0.420) | 0.548(±0.031) | 8.679(±5.518) | 0.018(±0.000) | 1.597(±1.222) | 0.033(±0.000) | 3.385(±2.597) | 1.970(±0.010) | 41.196(±43.492) | 0.041(±0.000) | 0.358(±0.260) | 0.058(±0.000) | 0.056(±0.017) |
| | Deep Coral | TB | 0.014(±0.000) | 0.212(±0.012) | 0.590(±0.045) | 4.547(±0.361) | 0.018(±0.000) | 0.679(±0.050) | 0.033(±0.001) | 1.427(±0.095) | 1.992(±0.028) | 13.829(±0.609) | 0.040(±0.002) | 0.158(±0.014) | 0.059(±0.001) | 0.049(±0.002) |
| PointNet | - | - | 0.023(±0.001) | 0.469(±0.055) | 2.240(±0.001) | 11.474(±0.290) | 0.026(±0.001) | 1.225(±0.165) | 0.051(±0.002) | 2.519(±0.385) | 2.860(±0.138) | 27.611(±5.693) | 0.057(±0.002) | 0.270(±0.046) | 0.063(±0.001) | 0.056(±0.011) |
| | DANN | DEV | 0.020(±0.001) | 1.126(±0.047) | 2.237(±0.002) | 18.207(±0.312) | 0.027(±0.001) | 3.377(±0.105) | 0.053(±0.004) | 7.181(±0.218) | 2.955(±0.180) | 106.747(±6.049) | 0.060(±0.003) | 0.727(±0.025) | 0.064(±0.002) | 0.088(±0.020) |
| | DANN | IWV | 0.020(±0.001) | 0.984(±0.424) | 2.241(±0.009) | 17.007(±3.626) | 0.028(±0.002) | 2.959(±1.236) | 0.054(±0.004) | 6.270(±2.634) | 2.967(±0.193) | 91.530(±44.149) | 0.064(±0.005) | 0.631(±0.265) | 0.063(±0.001) | 0.099(±0.034) |
| | DANN | SB | 0.019(±0.001) | 0.951(±0.347) | 2.239(±0.004) | 16.497(±3.351) | 0.027(±0.001) | 2.906(±1.015) | 0.052(±0.003) | 6.165(±2.173) | 2.886(±0.135) | 85.396(±36.953) | 0.063(±0.002) | 0.633(±0.213) | 0.063(±0.000) | 0.105(±0.049) |
| | DANN | TB | 0.020(±0.001) | 0.336(±0.054) | 2.239(±0.002) | 11.137(±0.329) | 0.027(±0.001) | 1.092(±0.206) | 0.052(±0.002) | 2.312(±0.421) | 2.988(±0.138) | 22.461(±4.074) | 0.060(±0.004) | 0.240(±0.047) | 0.064(±0.001) | 0.053(±0.005) |
| | CMD | DEV | 0.020(±0.001) | 1.164(±0.107) | 2.240(±0.002) | 18.411(±1.132) | 0.028(±0.001) | 3.591(±0.305) | 0.054(±0.003) | 7.643(±0.591) | 2.996(±0.178) | 102.004(±14.284) | 0.068(±0.009) | 0.790(±0.066) | 0.061(±0.001) | 0.079(±0.019) |
| | CMD | IWV | 0.020(±0.001) | 1.232(±0.036) | 2.240(±0.002) | 19.099(±0.332) | 0.028(±0.001) | 3.775(±0.107) | 0.055(±0.002) | 8.015(±0.237) | 3.053(±0.149) | 110.674(±4.420) | 0.072(±0.007) | 0.836(±0.039) | 0.061(±0.001) | 0.077(±0.021) |
| | CMD | SB | 0.019(±0.001) | 0.387(±0.059) | 2.241(±0.002) | 11.327(±0.574) | 0.027(±0.001) | 1.201(±0.297) | 0.051(±0.001) | 2.511(±0.699) | 2.852(±0.079) | 27.922(±6.676) | 0.060(±0.004) | 0.261(±0.084) | 0.061(±0.002) | 0.055(±0.013) |
| | CMD | TB | 0.019(±0.001) | 0.353(±0.078) | 2.240(±0.002) | 11.231(±0.508) | 0.026(±0.000) | 1.147(±0.284) | 0.051(±0.001) | 2.402(±0.634) | 2.843(±0.083) | 23.881(±4.994) | 0.058(±0.003) | 0.249(±0.077) | 0.061(±0.001) | 0.049(±0.002) |
| | DARE-GRAM | DEV | 0.021(±0.001) | 1.217(±0.034) | 2.240(±0.001) | 19.001(±0.305) | 0.029(±0.002) | 3.706(±0.123) | 0.058(±0.004) | 7.872(±0.254) | 3.205(±0.202) | 111.781(±4.997) | 0.076(±0.012) | 0.820(±0.035) | 0.062(±0.002) | 0.084(±0.007) |
| | DARE-GRAM | IWV | 0.019(±0.001) | 0.881(±0.440) | 2.240(±0.001) | 14.879(±4.056) | 0.027(±0.001) | 2.067(±0.830) | 0.051(±0.002) | 4.147(±1.607) | 2.878(±0.116) | 98.547(±75.723) | 0.059(±0.006) | 0.482(±0.249) | 0.062(±0.001) | 0.204(±0.083) |
| | DARE-GRAM | SB | 0.019(±0.001) | 0.773(±0.482) | 2.241(±0.001) | 14.018(±4.369) | 0.026(±0.001) | 1.933(±0.592) | 0.051(±0.001) | 3.948(±0.859) | 2.837(±0.073) | 84.934(±85.428) | 0.058(±0.003) | 0.479(±0.201) | 0.062(±0.001) | 0.117(±0.124) |
| | DARE-GRAM | TB | 0.019(±0.001) | 0.318(±0.068) | 2.238(±0.001) | 11.167(±0.357) | 0.026(±0.001) | 1.002(±0.267) | 0.050(±0.002) | 2.085(±0.571) | 2.870(±0.094) | 22.504(±3.875) | 0.060(±0.003) | 0.225(±0.061) | 0.062(±0.001) | 0.057(±0.008) |
| | Deep Coral | DEV | 0.020(±0.001) | 1.029(±0.106) | 2.239(±0.003) | 17.061(±1.272) | 0.028(±0.001) | 3.064(±0.375) | 0.056(±0.003) | 6.479(±0.889) | 3.027(±0.119) | 95.928(±11.663) | 0.068(±0.010) | 0.678(±0.087) | 0.061(±0.001) | 0.094(±0.008) |
| | Deep Coral | IWV | 0.020(±0.001) | 1.048(±0.167) | 2.238(±0.003) | 17.395(±1.880) | 0.028(±0.001) | 3.077(±0.508) | 0.055(±0.002) | 6.461(±1.120) | 2.984(±0.108) | 100.276(±20.956) | 0.067(±0.004) | 0.661(±0.117) | 0.061(±0.001) | 0.108(±0.010) |
| | Deep Coral | SB | 0.019(±0.000) | 0.977(±0.158) | 2.243(±0.002) | 16.764(±1.497) | 0.027(±0.001) | 2.947(±0.409) | 0.052(±0.001) | 6.257(±0.856) | 2.866(±0.081) | 88.933(±21.502) | 0.060(±0.004) | 0.651(±0.079) | 0.060(±0.001) | 0.094(±0.005) |
| | Deep Coral | TB | 0.019(±0.000) | 0.346(±0.078) | 2.239(±0.003) | 11.099(±0.287) | 0.027(±0.001) | 1.100(±0.270) | 0.051(±0.001) | 2.304(±0.618) | 2.857(±0.089) | 24.024(±6.005) | 0.058(±0.003) | 0.236(±0.061) | 0.063(±0.001) | 0.050(±0.005) |
| Transolver | - | - | 0.028(±0.001) | $\star$ | 0.580(±0.035) | $\star$ | 0.035(±0.001) | $\star$ | 0.070(±0.002) | $\star$ | 3.534(±0.088) | $\star$ | 0.077(±0.003) | $\star$ | 0.060(±0.005) | 0.650(±0.562) |
| | DANN | DEV | 0.024(±0.001) | 1.316(±0.047) | 0.566(±0.029) | 19.257(±0.628) | 0.037(±0.002) | 3.826(±0.120) | 0.074(±0.005) | 8.116(±0.248) | 3.534(±0.134) | 135.380(±7.612) | 0.084(±0.011) | 0.819(±0.030) | 0.062(±0.002) | 0.081(±0.007) |
| | DANN | IWV | 0.023(±0.001) | $\star$ | 0.563(±0.032) | $\star$ | 0.036(±0.002) | $\star$ | 0.072(±0.004) | $\star$ | 3.505(±0.111) | $\star$ | 0.080(±0.006) | $\star$ | 0.060(±0.004) | $\star$ |
| | DANN | SB | 0.023(±0.000) | $\star$ | 0.557(±0.026) | $\star$ | 0.035(±0.000) | $\star$ | 0.068(±0.001) | $\star$ | 3.413(±0.037) | $\star$ | 0.078(±0.003) | $\star$ | 0.065(±0.002) | $\star$ |
| | DANN | TB | 0.023(±0.001) | 1.248(±0.044) | 0.581(±0.052) | 18.346(±0.511) | 0.036(±0.002) | 3.634(±0.123) | 0.072(±0.004) | 7.702(±0.270) | 3.477(±0.130) | 126.738(±6.268) | 0.081(±0.012) | 0.772(±0.028) | 0.062(±0.005) | 0.083(±0.008) |
| | CMD | DEV | 0.024(±0.001) | 1.122(±0.311) | 0.615(±0.033) | 16.889(±3.999) | 0.037(±0.001) | 3.344(±0.794) | 0.073(±0.002) | 7.127(±1.619) | 3.573(±0.097) | 107.289(±44.373) | 0.085(±0.008) | 0.716(±0.151) | 0.059(±0.004) | 0.119(±0.041) |
| | CMD | IWV | 0.024(±0.001) | 2.631(±3.517) | 0.598(±0.040) | 78.602(±130.758) | 0.037(±0.002) | 4.817(±4.473) | 0.073(±0.003) | 12.726(±14.421) | 3.598(±0.107) | 351.111(±537.157) | 0.083(±0.007) | 1.370(±1.618) | 0.062(±0.004) | 0.502(±0.846) |
| | CMD | SB | 0.023(±0.000) | 0.781(±0.350) | 0.587(±0.024) | 11.988(±4.742) | 0.036(±0.001) | 2.352(±0.967) | 0.071(±0.002) | 4.980(±2.072) | 3.556(±0.090) | 71.526(±40.616) | 0.081(±0.006) | 0.507(±0.201) | 0.062(±0.004) | 0.086(±0.013) |
| | CMD | TB | 0.024(±0.001) | 0.567(±0.137) | 0.615(±0.046) | 9.350(±2.012) | 0.037(±0.002) | 1.798(±0.445) | 0.074(±0.003) | 3.834(±1.010) | 3.593(±0.154) | 41.987(±11.589) | 0.082(±0.004) | 0.405(±0.100) | 0.058(±0.004) | 0.071(±0.023) |
| | DARE-GRAM | DEV | 0.023(±0.001) | 3.153(±2.545) | 0.558(±0.028) | 51.238(±44.025) | 0.036(±0.001) | 8.082(±6.894) | 0.071(±0.003) | 15.250(±12.194) | 3.462(±0.093) | 360.475(±324.276) | 0.078(±0.006) | 1.193(±0.663) | 0.063(±0.002) | 0.232(±0.185) |
| | DARE-GRAM | IWV | 0.023(±0.001) | $\star$ | 0.554(±0.020) | $\star$ | 0.036(±0.002) | $\star$ | 0.070(±0.004) | $\star$ | 3.431(±0.071) | $\star$ | 0.078(±0.005) | $\star$ | 0.059(±0.005) | 0.666(±0.660) |
| | DARE-GRAM | SB | 0.022(±0.000) | $\star$ | 0.545(±0.020) | $\star$ | 0.034(±0.001) | 22.287(±33.097) | 0.067(±0.002) | 46.255(±69.242) | 3.385(±0.049) | $\star$ | 0.073(±0.005) | 3.331(±4.667) | 0.060(±0.005) | $\star$ |
| | DARE-GRAM | TB | 0.023(±0.001) | 0.827(±0.112) | 0.556(±0.010) | 12.936(±1.844) | 0.035(±0.002) | 2.619(±0.376) | 0.070(±0.004) | 5.681(±0.772) | 3.415(±0.082) | 62.083(±11.908) | 0.077(±0.004) | 0.570(±0.069) | 0.063(±0.002) | 0.095(±0.020) |
| | Deep Coral | DEV | 0.023(±0.001) | 3.242(±4.897) | 0.596(±0.014) | 92.226(±160.351) | 0.035(±0.002) | 9.242(±13.641) | 0.070(±0.003) | 22.735(±34.969) | 3.475(±0.171) | 254.700(±375.199) | 0.079(±0.005) | 1.587(±2.089) | 0.060(±0.002) | 0.376(±0.598) |
| | Deep Coral | IWV | 0.023(±0.001) | $\star$ | 0.580(±0.023) | $\star$ | 0.035(±0.002) | $\star$ | 0.069(±0.005) | $\star$ | 3.439(±0.114) | $\star$ | 0.080(±0.005) | 4.597(±4.748) | 0.062(±0.003) | 0.603(±0.492) |
| | Deep Coral | SB | 0.023(±0.001) | 3.611(±4.679) | 0.582(±0.017) | 95.093(±158.462) | 0.034(±0.002) | 10.329(±12.991) | 0.068(±0.004) | 25.707(±33.261) | 3.402(±0.079) | 270.011(±365.767) | 0.076(±0.005) | 1.931(±1.940) | 0.064(±0.003) | 0.813(±0.768) |
| | Deep Coral | TB | 0.023(±0.000) | 0.656(±0.188) | 0.589(±0.021) | 10.202(±2.895) | 0.037(±0.001) | 1.985(±0.589) | 0.072(±0.003) | 4.247(±1.285) | 3.518(±0.053) | 53.782(±18.391) | 0.077(±0.005) | 0.427(±0.119) | 0.061(±0.004) | 0.078(±0.014) |

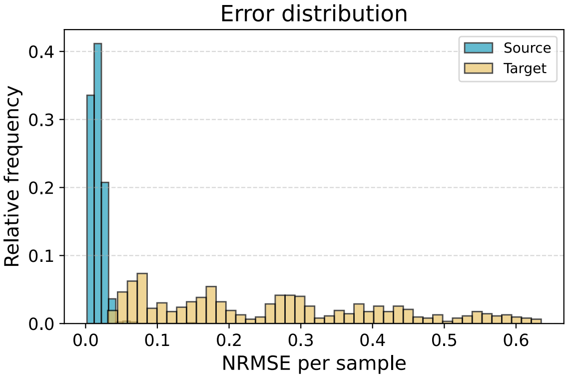


Figure 7: Distribution of NRMSE (averaged across all fields) for the test sets of the source (blue) and target (yellow) domains in the *hot rolling* dataset. Bar height indicates the relative frequency of samples within each bin.

Table 5: Absolute error of PEEQ predictions for representative samples from the source and target domain of the *hot rolling* dataset. Lowest value per metric is bold.

| Metric | Source | Target |
|---|---|---|
| Mean | **2.07e-04** | 1.46e-02 |
| Std | **1.87e-04** | 2.73e-03 |
| Median | **1.66e-04** | 1.49e-02 |
| $Q_{01}$ | **7.45e-09** | 6.82e-03 |
| $Q_{25}$ | **6.38e-05** | 1.37e-02 |
| $Q_{75}$ | **2.99e-04** | 1.58e-02 |
| $Q_{99}$ | **7.61e-04** | 2.14e-02 |

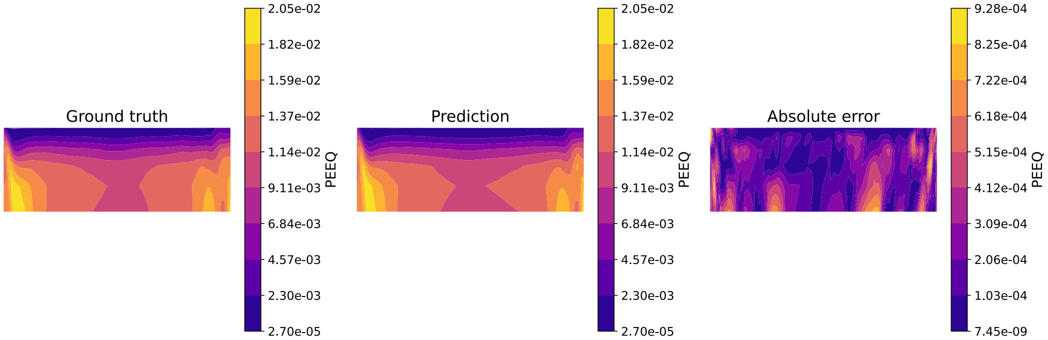


Figure 8: Fringe plot of the *hot rolling* dataset (representative source sample). Shown is the ground truth (left) and predicted (middle) PEEQ, as well as the absolute error (right).

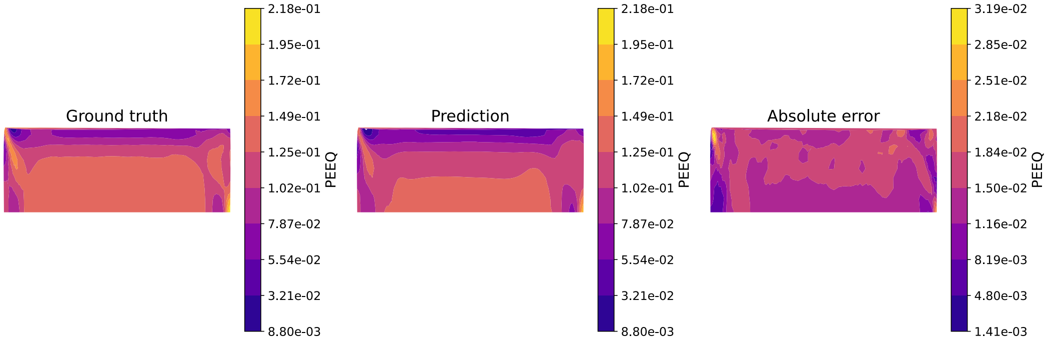


Figure 9: Fringe plot of the *hot rolling* dataset (representative target sample). Shown is the ground truth (left) and predicted (middle) PEEQ, as well as the absolute error (right).

### B.2 SHEET METAL FORMING

In contrast to the substantial shift observed in the hot rolling dataset, the distribution shift in the *sheet metal forming* dataset is moderate. Table 6 presents the detailed performance across all models, algorithms, and selections for this dataset.

Table 6: RMSE (mean ± std over 4 seeds) on the *sheet metal forming* dataset at *medium* difficulty. Values are target domain errors (lower is better). Bold marks the overall best model + UDA algorithm + model selection combination. For each architecture, the unregularized baseline row is shaded beige, whereas the best UDA + selection within that architecture is underlined and shaded green.

| Model | DA Algorithm | Model Selection | All Fields Normalized Avg (-) | | Deformation (mm) | | Logarithmic Strain ($\times 10^{-2}$) | | Equivalent Plastic Strain ($\times 10^{-2}$) | | Mises Stress (MPa) | | Rel Custom Error (-) | | VM Consistency (-) | | Violation Elastic ($\times 10^{-2}$) | | Residual Plastic (-) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT |
| GraphSAGE | - | - | 0.070(±0.002) | 0.376(±0.028) | 1.411(±0.070) | 1.939(±0.530) | 0.024(±0.001) | 0.156(±0.014) | 0.043(±0.001) | 0.272(±0.026) | 11.022(±0.324) | 46.097(±4.911) | 0.238(±0.013) | 2.496(±0.088) | 0.063(±0.002) | 0.076(±0.009) | 0.018(±0.004) | 0.277(±0.152) | 0.508(±0.044) | 0.481(±0.043) |
| | DANN | DEV | 0.056(±0.004) | ⋆ | 1.347(±0.045) | 16.199(±21.097) | 0.023(±0.001) | 0.965(±1.238) | 0.042(±0.003) | ⋆ | 10.597(±0.564) | 406.576(±403.134) | 0.237(±0.006) | 10.849(±9.217) | 0.062(±0.002) | 0.286(±0.154) | 0.009(±0.003) | 0.372(±0.241) | 0.545(±0.049) | 0.945(±0.459) |
| | DANN | IWV | 0.056(±0.003) | 0.334(±0.036) | 1.400(±0.075) | 2.136(±0.182) | 0.023(±0.002) | 0.164(±0.021) | 0.042(±0.003) | 0.290(±0.039) | 10.676(±0.347) | 52.917(±8.385) | 0.237(±0.009) | 5.384(±4.228) | 0.063(±0.001) | 0.091(±0.024) | 0.017(±0.008) | 0.110(±0.038) | 0.532(±0.025) | 0.509(±0.041) |
| | DANN | SB | 0.055(±0.002) | 1.139(±0.411) | 1.404(±0.035) | 7.810(±6.066) | 0.022(±0.001) | 0.467(±0.147) | 0.040(±0.001) | 0.921(±0.452) | 10.732(±0.406) | 186.098(±37.057) | 0.236(±0.005) | 5.603(±1.282) | 0.063(±0.002) | 0.418(±0.077) | 0.014(±0.004) | 0.736(±1.097) | 0.540(±0.035) | 0.703(±0.019) |
| | DANN | TB | 0.057(±0.003) | 0.323(±0.025) | 1.416(±0.055) | 2.021(±0.156) | 0.023(±0.001) | 0.156(±0.010) | 0.042(±0.003) | 0.265(±0.004) | 10.728(±0.218) | 49.234(±5.606) | 0.239(±0.013) | 3.580(±0.384) | 0.063(±0.001) | 0.092(±0.023) | 0.013(±0.002) | 0.150(±0.071) | 0.520(±0.023) | 0.474(±0.021) |
| | CMD | DEV | 0.055(±0.002) | 0.857(±0.475) | 1.355(±0.058) | 6.409(±3.878) | 0.022(±0.001) | 0.380(±0.177) | 0.041(±0.002) | 0.645(±0.290) | 10.590(±0.343) | 145.233(±79.730) | 0.236(±0.002) | 7.008(±3.495) | 0.063(±0.001) | 0.315(±0.266) | 0.021(±0.013) | 0.209(±0.380) | 0.549(±0.044) | 0.652(±0.079) |
| | CMD | IWV | 0.055(±0.001) | 0.407(±0.124) | 1.326(±0.031) | 2.455(±1.014) | 0.022(±0.001) | 0.201(±0.057) | 0.041(±0.001) | 0.358(±0.112) | 10.730(±0.065) | 61.685(±20.293) | 0.235(±0.008) | 4.179(±1.820) | 0.063(±0.001) | 0.088(±0.022) | 0.013(±0.010) | 0.148(±0.069) | 0.537(±0.030) | 0.543(±0.038) |
| | CMD | SB | 0.055(±0.001) | 0.569(±0.306) | 1.433(±0.024) | 4.708(±4.280) | 0.022(±0.000) | 0.290(±0.160) | 0.040(±0.000) | 0.497(±0.273) | 10.550(±0.163) | 99.069(±58.190) | 0.230(±0.008) | 5.015(±3.239) | 0.062(±0.001) | 0.164(±0.144) | 0.010(±0.002) | 0.086(±0.082) | 0.509(±0.038) | 0.537(±0.092) |
| | CMD | TB | 0.057(±0.001) | 0.289(±0.036) | 1.345(±0.059) | 2.028(±0.798) | 0.023(±0.000) | 0.139(±0.017) | 0.042(±0.001) | 0.243(±0.028) | 10.828(±0.169) | 43.746(±5.836) | 0.234(±0.008) | 3.005(±0.236) | 0.063(±0.001) | 0.078(±0.029) | 0.017(±0.014) | 0.273(±0.284) | 0.543(±0.019) | 0.513(±0.045) |
| | DARE-GRAM | DEV | 0.055(±0.001) | 0.947(±0.578) | 1.396(±0.129) | 9.932(±11.711) | 0.022(±0.001) | 0.483(±0.316) | 0.041(±0.002) | 0.997(±0.726) | 10.652(±0.254) | 112.696(±49.940) | 0.233(±0.015) | 6.555(±2.805) | 0.062(±0.001) | 0.127(±0.064) | 0.027(±0.019) | 0.508(±0.429) | 0.576(±0.020) | 0.645(±0.118) |
| | DARE-GRAM | IWV | 0.055(±0.002) | 0.397(±0.107) | 1.335(±0.030) | 3.158(±2.365) | 0.022(±0.001) | 0.204(±0.067) | 0.040(±0.002) | 0.374(±0.145) | 10.497(±0.351) | 63.645(±21.570) | 0.226(±0.009) | 9.885(±8.520) | 0.062(±0.002) | 0.122(±0.068) | 0.008(±0.010) | 0.199(±0.141) | 0.544(±0.028) | 0.591(±0.049) |
| | DARE-GRAM | SB | 0.055(±0.001) | 1.077(±0.456) | 1.324(±0.021) | 10.927(±10.949) | 0.022(±0.001) | 0.527(±0.268) | 0.040(±0.001) | 1.081(±0.632) | 10.592(±0.216) | 139.111(±44.888) | 0.229(±0.007) | 5.995(±2.638) | 0.063(±0.001) | 0.147(±0.057) | 0.025(±0.019) | 1.102(±1.311) | 0.574(±0.021) | 0.667(±0.112) |
| | DARE-GRAM | TB | 0.057(±0.001) | 0.294(±0.017) | 1.342(±0.007) | 1.815(±0.448) | 0.023(±0.001) | 0.141(±0.011) | 0.042(±0.001) | 0.245(±0.018) | 10.661(±0.310) | 45.427(±3.991) | 0.231(±0.002) | 2.909(±0.572) | 0.063(±0.002) | 0.074(±0.010) | 0.016(±0.010) | 0.254(±0.150) | 0.541(±0.019) | 0.460(±0.068) |
| | Deep Coral | DEV | 0.054(±0.001) | 0.411(±0.103) | 1.347(±0.048) | 3.347(±2.232) | 0.021(±0.001) | 0.185(±0.031) | 0.039(±0.001) | 0.330(±0.064) | 10.355(±0.455) | 65.861(±28.277) | 0.228(±0.012) | ⋆ | 0.062(±0.002) | 0.088(±0.016) | 0.014(±0.009) | 0.432(±0.235) | 0.544(±0.047) | 0.544(±0.012) |
| | Deep Coral | IWV | 0.055(±0.003) | 1.396(±2.091) | 1.415(±0.063) | 6.244(±7.609) | 0.022(±0.001) | 0.478(±0.622) | 0.041(±0.002) | 0.803(±0.994) | 10.668(±0.377) | 86.669(±76.339) | 0.232(±0.015) | 4.353(±1.012) | 0.062(±0.002) | 0.092(±0.032) | 0.016(±0.008) | 0.321(±0.234) | 0.549(±0.035) | 0.590(±0.104) |
| | Deep Coral | SB | 0.056(±0.002) | 0.364(±0.105) | 1.392(±0.071) | 2.386(±0.735) | 0.022(±0.001) | 0.177(±0.055) | 0.041(±0.001) | 0.310(±0.090) | 10.744(±0.189) | 52.764(±11.554) | 0.240(±0.007) | 3.685(±0.837) | 0.063(±0.001) | 0.090(±0.024) | 0.017(±0.020) | 0.360(±0.245) | 0.533(±0.018) | 0.522(±0.046) |
| | Deep Coral | TB | 0.056(±0.003) | 0.287(±0.011) | 1.395(±0.068) | 1.825(±0.369) | 0.023(±0.001) | 0.137(±0.007) | 0.041(±0.002) | 0.242(±0.008) | 10.781(±0.333) | 44.161(±3.225) | 0.239(±0.015) | 2.905(±0.494) | 0.062(±0.001) | 0.071(±0.004) | 0.012(±0.007) | 0.123(±0.104) | 0.552(±0.021) | 0.521(±0.042) |
| PointNet | - | - | 0.077(±0.011) | 0.226(±0.047) | 2.012(±0.149) | 2.556(±0.948) | 0.024(±0.004) | 0.087(±0.022) | 0.045(±0.007) | 0.160(±0.039) | 11.357(±2.106) | 31.435(±6.317) | 0.238(±0.067) | 0.881(±0.135) | 0.098(±0.001) | 0.105(±0.015) | 0.044(±0.021) | 0.254(±0.144) | 0.514(±0.041) | 0.481(±0.055) |
| | DANN | DEV | 0.069(±0.007) | 1.164(±1.954) | 2.255(±0.031) | 5.817(±7.136) | 0.025(±0.003) | 0.694(±1.204) | 0.048(±0.005) | 1.499(±2.667) | 12.483(±1.748) | 122.815(±182.772) | 0.326(±0.016) | 5.403(±8.164) | 0.098(±0.002) | 0.263(±0.306) | 0.041(±0.013) | 0.873(±1.160) | 0.540(±0.004) | 0.605(±0.236) |
| | DANN | IWV | 0.071(±0.006) | 0.269(±0.177) | 2.254(±0.020) | 4.586(±5.092) | 0.026(±0.002) | 0.135(±0.082) | 0.049(±0.004) | 0.243(±0.150) | 12.935(±1.330) | 49.068(±41.241) | 0.330(±0.011) | 1.009(±0.370) | 0.098(±0.001) | 0.161(±0.113) | 0.053(±0.040) | 0.290(±0.208) | 0.558(±0.046) | 0.583(±0.110) |
| | DANN | SB | 0.067(±0.005) | 0.359(±0.153) | 2.250(±0.022) | 5.573(±4.577) | 0.025(±0.002) | 0.181(±0.076) | 0.047(±0.004) | 0.328(±0.138) | 12.090(±1.186) | 63.622(±34.927) | 0.327(±0.012) | 1.721(±0.875) | 0.097(±0.001) | 0.175(±0.105) | 0.058(±0.024) | 0.850(±0.902) | 0.528(±0.044) | 0.580(±0.113) |
| | DANN | TB | 0.076(±0.004) | 0.166(±0.008) | 2.270(±0.037) | 2.089(±0.144) | 0.028(±0.001) | 0.084(±0.010) | 0.053(±0.002) | 0.149(±0.016) | 14.069(±1.203) | 24.299(±2.097) | 0.336(±0.008) | 1.015(±0.194) | 0.098(±0.002) | 0.097(±0.009) | 0.096(±0.042) | 0.270(±0.195) | 0.530(±0.019) | 0.437(±0.042) |
| | CMD | DEV | 0.067(±0.005) | 0.275(±0.125) | 2.275(±0.065) | 2.987(±0.781) | 0.025(±0.002) | 0.134(±0.058) | 0.047(±0.004) | 0.240(±0.105) | 11.932(±0.837) | 53.092(±34.512) | 0.312(±0.016) | 1.416(±0.828) | 0.098(±0.001) | 0.159(±0.101) | 0.106(±0.045) | 0.264(±0.115) | 0.571(±0.043) | 0.548(±0.079) |
| | CMD | IWV | 0.072(±0.003) | 0.244(±0.146) | 2.253(±0.039) | 2.660(±0.994) | 0.027(±0.001) | 0.117(±0.070) | 0.050(±0.002) | 0.213(±0.125) | 13.159(±0.877) | 47.263(±38.465) | 0.314(±0.011) | 1.587(±0.689) | 0.099(±0.000) | 0.150(±0.108) | 0.093(±0.023) | 0.264(±0.156) | 0.575(±0.038) | 0.575(±0.061) |
| | CMD | SB | 0.060(±0.006) | 0.252(±0.066) | 1.988(±0.069) | 3.698(±1.484) | 0.022(±0.002) | 0.124(±0.029) | 0.042(±0.005) | 0.221(±0.049) | 10.166(±1.459) | 38.406(±13.599) | 0.253(±0.050) | 1.185(±0.297) | 0.097(±0.001) | 0.109(±0.018) | 0.060(±0.038) | 0.855(±1.125) | 0.504(±0.035) | 0.494(±0.074) |
| | CMD | TB | 0.069(±0.006) | 0.173(±0.013) | 2.099(±0.124) | 2.114(±0.141) | 0.026(±0.003) | 0.089(±0.011) | 0.049(±0.005) | 0.158(±0.019) | 12.260(±0.750) | 25.184(±1.660) | 0.269(±0.046) | 1.186(±0.308) | 0.099(±0.002) | 0.097(±0.005) | 0.065(±0.043) | 0.320(±0.055) | 0.518(±0.025) | 0.512(±0.053) |
| | DARE-GRAM | DEV | 0.067(±0.006) | 0.191(±0.025) | 2.280(±0.068) | 2.412(±0.492) | 0.025(±0.003) | 0.092(±0.016) | 0.046(±0.005) | 0.166(±0.027) | 12.033(±1.429) | 29.955(±2.932) | 0.325(±0.020) | 1.228(±0.425) | 0.097(±0.002) | 0.095(±0.006) | 0.038(±0.018) | 0.275(±0.180) | 0.545(±0.028) | 0.495(±0.062) |
| | DARE-GRAM | IWV | 0.071(±0.004) | 0.202(±0.062) | 2.259(±0.055) | 2.355(±0.491) | 0.026(±0.002) | 0.094(±0.032) | 0.050(±0.003) | 0.171(±0.058) | 13.040(±0.723) | 31.728(±6.262) | 0.329(±0.016) | 1.356(±0.420) | 0.098(±0.001) | 0.101(±0.012) | 0.037(±0.016) | 0.393(±0.361) | 0.544(±0.023) | 0.520(±0.048) |
| | DARE-GRAM | SB | 0.067(±0.004) | 0.224(±0.073) | 2.280(±0.041) | 2.959(±0.991) | 0.025(±0.002) | 0.110(±0.031) | 0.046(±0.004) | 0.204(±0.064) | 12.074(±1.154) | 35.062(±14.369) | 0.313(±0.009) | 1.260(±0.786) | 0.098(±0.002) | 0.112(±0.025) | 0.096(±0.065) | 0.575(±0.572) | 0.571(±0.032) | 0.548(±0.110) |
| | DARE-GRAM | TB | 0.074(±0.003) | 0.162(±0.012) | 2.253(±0.045) | 2.021(±0.103) | 0.028(±0.002) | 0.082(±0.012) | 0.052(±0.003) | 0.148(±0.021) | 13.647(±0.489) | 23.888(±2.240) | 0.325(±0.021) | 1.012(±0.174) | 0.098(±0.001) | 0.095(±0.007) | 0.091(±0.055) | 0.347(±0.241) | 0.544(±0.009) | 0.502(±0.076) |
| | Deep Coral | DEV | 0.065(±0.007) | 0.318(±0.236) | 2.194(±0.183) | 2.814(±1.229) | 0.024(±0.003) | 0.168(±0.133) | 0.045(±0.005) | 0.275(±0.176) | 11.634(±1.615) | 46.719(±31.265) | 0.287(±0.074) | 1.710(±0.426) | 0.098(±0.001) | 0.114(±0.021) | 0.084(±0.037) | 0.602(±0.372) | 0.532(±0.033) | 0.554(±0.083) |
| | Deep Coral | IWV | 0.064(±0.006) | 0.190(±0.027) | 2.196(±0.185) | 2.324(±0.411) | 0.024(±0.002) | 0.092(±0.013) | 0.044(±0.005) | 0.166(±0.022) | 11.283(±1.392) | 32.908(±5.779) | 0.286(±0.073) | 1.425(±0.388) | 0.097(±0.000) | 0.106(±0.015) | 0.068(±0.064) | 0.323(±0.118) | 0.554(±0.041) | 0.545(±0.056) |
| | Deep Coral | SB | 0.060(±0.009) | 0.182(±0.021) | 2.042(±0.185) | 2.555(±0.422) | 0.022(±0.004) | 0.084(±0.011) | 0.042(±0.008) | 0.150(±0.023) | 10.156(±2.001) | 31.345(±5.362) | 0.242(±0.079) | 1.154(±0.368) | 0.097(±0.001) | 0.096(±0.011) | 0.036(±0.025) | 0.261(±0.249) | 0.504(±0.041) | 0.451(±0.027) |
| | Deep Coral | TB | 0.069(±0.014) | 0.158(±0.006) | 2.129(±0.184) | 2.004(±0.051) | 0.026(±0.006) | 0.078(±0.005) | 0.049(±0.011) | 0.140(±0.009) | 12.320(±3.129) | 22.942(±1.429) | 0.283(±0.080) | 0.914(±0.144) | 0.099(±0.003) | 0.092(±0.007) | 0.044(±0.025) | 0.122(±0.114) | 0.533(±0.052) | 0.497(±0.033) |
| **Transolver** | - | - | 0.069(±0.002) | 0.168(±0.029) | 1.169(±0.012) | 1.185(±0.292) | 0.022(±0.001) | 0.069(±0.015) | 0.041(±0.001) | 0.126(±0.029) | 12.854(±0.469) | 22.970(±4.818) | 0.216(±0.010) | 0.608(±0.135) | 0.059(±0.001) | 0.060(±0.015) | 0.003(±0.001) | 0.010(±0.008) | 0.538(±0.014) | 0.483(±0.016) |
| | DANN | DEV | 0.056(±0.002) | 0.206(±0.051) | 1.209(±0.062) | 2.627(±1.500) | 0.021(±0.001) | 0.103(±0.022) | 0.040(±0.001) | 0.187(±0.038) | 12.255(±0.526) | 36.754(±15.109) | 0.211(±0.007) | 0.969(±0.298) | 0.060(±0.001) | 0.096(±0.054) | 0.001(±0.000) | 0.032(±0.041) | 0.538(±0.014) | 0.508(±0.027) |
| | DANN | IWV | 0.056(±0.003) | 0.165(±0.026) | 1.190(±0.049) | 1.472(±0.540) | 0.021(±0.001) | 0.081(±0.011) | 0.040(±0.002) | 0.150(±0.022) | 12.208(±0.551) | 26.724(±7.000) | 0.211(±0.009) | 0.920(±0.261) | 0.059(±0.000) | 0.065(±0.014) | 0.002(±0.001) | 0.059(±0.047) | 0.531(±0.009) | 0.444(±0.060) |
| | DANN | SB | 0.056(±0.002) | 0.162(±0.016) | 1.226(±0.029) | 1.490(±0.329) | 0.021(±0.001) | 0.080(±0.005) | 0.039(±0.001) | 0.147(±0.010) | 12.123(±0.323) | 26.489(±5.455) | 0.207(±0.007) | 0.888(±0.179) | 0.060(±0.001) | 0.068(±0.010) | 0.003(±0.002) | 0.055(±0.043) | 0.536(±0.012) | 0.474(±0.078) |
| | DANN | TB | 0.058(±0.002) | 0.133(±0.016) | 1.247(±0.056) | 1.201(±0.273) | 0.022(±0.001) | 0.063(±0.013) | 0.041(±0.001) | 0.117(±0.025) | 12.546(±0.653) | 21.246(±1.888) | 0.208(±0.009) | 0.617(±0.103) | 0.060(±0.000) | 0.058(±0.007) | 0.001(±0.001) | 0.012(±0.018) | 0.539(±0.014) | 0.466(±0.030) |
| | CMD | DEV | 0.058(±0.002) | 0.277(±0.132) | 1.223(±0.059) | 3.851(±3.289) | 0.022(±0.001) | 0.138(±0.064) | 0.041(±0.001) | 0.249(±0.113) | 12.778(±0.921) | 49.824(±31.899) | 0.207(±0.004) | 3.052(±4.046) | 0.060(±0.001) | 0.156(±0.137) | 0.003(±0.005) | 0.128(±0.101) | 0.545(±0.004) | 0.584(±0.116) |
| | CMD | IWV | 0.056(±0.001) | 0.202(±0.098) | 1.206(±0.032) | 2.076(±1.683) | 0.021(±0.000) | 0.105(±0.055) | 0.040(±0.001) | 0.187(±0.093) | 12.360(±0.581) | 30.106(±14.483) | 0.203(±0.003) | 2.845(±4.182) | 0.059(±0.000) | 0.081(±0.042) | 0.002(±0.001) | 0.098(±0.077) | 0.531(±0.013) | 0.572(±0.034) |
| | CMD | SB | 0.056(±0.002) | 0.276(±0.170) | 1.209(±0.064) | 3.714(±3.317) | 0.021(±0.001) | 0.141(±0.087) | 0.040(±0.001) | 0.251(±0.155) | 12.275(±0.422) | 46.610(±33.500) | 0.208(±0.004) | 1.422(±0.811) | 0.060(±0.001) | 0.113(±0.068) | 0.002(±0.002) | 0.022(±0.034) | 0.541(±0.019) | 0.538(±0.117) |
| | CMD | TB | 0.061(±0.001) | 0.131(±0.008) | 1.260(±0.043) | 1.021(±0.223) | 0.023(±0.001) | 0.065(±0.005) | 0.044(±0.001) | 0.117(±0.007) | 13.482(±0.436) | 20.275(±1.756) | 0.211(±0.011) | 0.796(±0.133) | 0.060(±0.001) | 0.053(±0.005) | 0.003(±0.002) | 0.015(±0.009) | 0.556(±0.010) | 0.506(±0.044) |
| | DARE-GRAM | DEV | 0.056(±0.002) | 0.175(±0.020) | 1.165(±0.011) | 1.814(±0.418) | 0.021(±0.001) | 0.087(±0.010) | 0.040(±0.001) | 0.160(±0.018) | 12.237(±0.765) | 29.784(±3.926) | 0.213(±0.007) | 1.047(±0.395) | 0.060(±0.001) | 0.078(±0.012) | 0.003(±0.002) | 0.044(±0.035) | 0.538(±0.016) | 0.549(±0.023) |
| | DARE-GRAM | IWV | 0.056(±0.001) | 0.167(±0.020) | 1.196(±0.021) | 1.625(±0.286) | 0.021(±0.000) | 0.084(±0.011) | 0.040(±0.001) | 0.153(±0.018) | 12.259(±0.352) | 25.986(±3.087) | 0.208(±0.007) | 0.874(±0.190) | 0.060(±0.001) | 0.065(±0.006) | 0.002(±0.001) | 0.042(±0.053) | 0.548(±0.014) | 0.505(±0.046) |
| | DARE-GRAM | SB | 0.055(±0.001) | 0.181(±0.014) | 1.193(±0.018) | 1.847(±0.355) | 0.021(±0.000) | 0.092(±0.009) | 0.039(±0.001) | 0.169(±0.017) | 12.012(±0.513) | 29.238(±2.975) | 0.207(±0.005) | 1.057(±0.411) | 0.060(±0.000) | 0.074(±0.012) | 0.003(±0.002) | 0.051(±0.046) | 0.543(±0.012) | 0.501(±0.038) |
| | DARE-GRAM | TB | 0.059(±0.001) | 0.135(±0.020) | 1.243(±0.099) | 1.016(±0.192) | 0.022(±0.001) | 0.069(±0.013) | 0.042(±0.001) | 0.125(±0.022) | 13.140(±0.255) | 19.899(±2.610) | 0.204(±0.003) | 0.704(±0.124) | 0.060(±0.002) | 0.052(±0.006) | 0.003(±0.003) | 0.036(±0.041) | 0.541(±0.011) | 0.542(±0.026) |
| | **Deep Coral** | **DEV** | 0.058(±0.001) | **0.154(±0.001)** | 1.238(±0.027) | 1.317(±0.159) | 0.022(±0.000) | 0.077(±0.003) | 0.041(±0.001) | 0.141(±0.006) | 12.894(±0.186) | 24.427(±1.420) | 0.209(±0.008) | 0.806(±0.155) | 0.059(±0.001) | 0.062(±0.002) | 0.001(±0.001) | 0.023(±0.019) | 0.551(±0.015) | 0.581(±0.021) |
| | Deep Coral | IWV | 0.057(±0.001) | 0.255(±0.205) | 1.215(±0.022) | 2.937(±3.128) | 0.021(±0.000) | 0.129(±0.109) | 0.040(±0.001) | 0.234(±0.195) | 12.609(±0.274) | 43.610(±38.032) | 0.209(±0.008) | 1.342(±0.973) | 0.059(±0.001) | 0.115(±0.112) | 0.002(±0.001) | 0.046(±0.071) | 0.553(±0.013) | 0.602(±0.072) |
| | Deep Coral | SB | 0.057(±0.001) | 0.263(±0.201) | 1.197(±0.016) | 3.264(±2.939) | 0.021(±0.000) | 0.135(±0.105) | 0.040(±0.001) | 0.243(±0.189) | 12.495(±0.186) | 44.304(±37.610) | 0.209(±0.009) | 1.610(±0.952) | 0.060(±0.000) | 0.123(±0.107) | 0.002(±0.001) | 0.052(±0.067) | 0.552(±0.010) | 0.564(±0.090) |
| | Deep Coral | TB | 0.059(±0.001) | 0.138(±0.015) | 1.224(±0.016) | 0.952(±0.031) | 0.022(±0.000) | 0.068(±0.012) | 0.042(±0.001) | 0.124(±0.023) | 12.946(±0.503) | 22.035(±2.227) | 0.205(±0.008) | 0.797(±0.260) | 0.060(±0.001) | 0.053(±0.002) | 0.003(±0.002) | 0.029(±0.018) | 0.531(±0.010) | 0.516(±0.011) |

To further illustrate model behavior under distribution shift, we examine the best performing model, selected by lowest normalized average target domain error. The error distribution (Figure 10) shows a moderate distribution shift between the source and target domain with some outliers in the target domain.

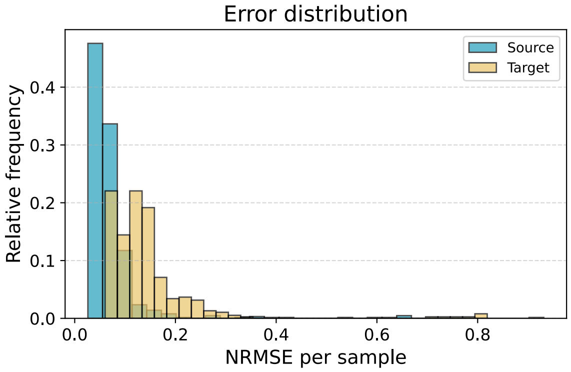


Figure 10: Distribution of NRMSE (averaged across all fields) for the test sets of the source (blue) and target (yellow) domains in the *sheet metal forming* dataset. Bar height indicates the relative frequency of samples within each bin.

To better understand the model's predictive behavior in this setting, we analyze best- and worst-case examples in each domain, again focusing on the critical PEEQ field. Table 7 provides a statistical summary of the absolute PEEQ prediction errors across the selected cases. Fringe plots in Figures 11 to 14 provide a visual understanding of model accuracy. These visualizations emphasize that while the best prediction in the target domain remains reasonably accurate, others (e.g., the worst case sample) exhibit notable discrepancies in the localized regions around the bends that we are most interested in.

Table 7: Absolute error of PEEQ predictions for the best and worst samples from the source and target domain of the *sheet metal forming* dataset. Lowest value per metric is bold.

| **Metric** | **Source** | | **Target** | |
|---|---|---|---|---|
| | **Best** | **Worst** | **Best** | **Worst** |
| Mean | **5.47e-05** | 1.68e-04 | 1.72e-04 | 1.86e-03 |
| Std | **1.35e-04** | 3.80e-04 | 4.52e-04 | 5.97e-03 |
| Median | **1.96e-05** | 4.35e-05 | 5.58e-05 | 2.85e-04 |
| $Q_{01}$ | **2.43e-07** | 6.52e-07 | 1.03e-06 | 6.66e-06 |
| $Q_{25}$ | **7.49e-06** | 1.84e-05 | 2.60e-05 | 1.53e-04 |
| $Q_{75}$ | **4.05e-05** | 1.05e-04 | 9.10e-05 | 4.67e-04 |
| $Q_{99}$ | **7.37e-04** | 1.96e-03 | 2.56e-03 | 3.40e-02 |

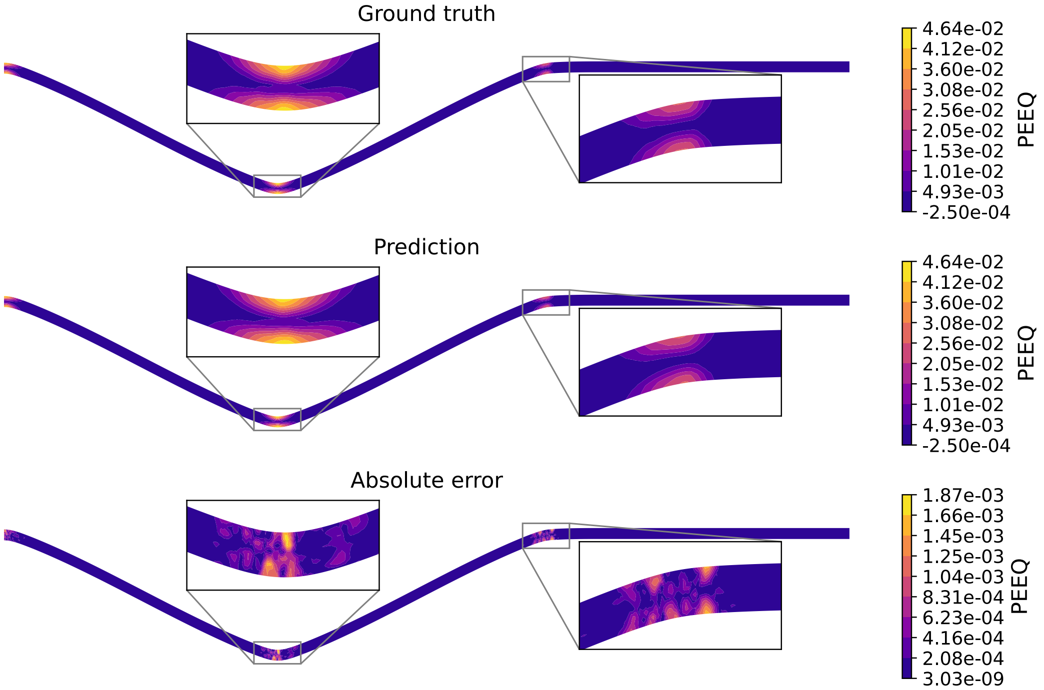


Figure 11: Fringe plot of the *sheet metal forming* dataset (best source sample). Shown is the ground truth (top) and predicted (middle) PEEQ, aswell as the absolute error (bottom).

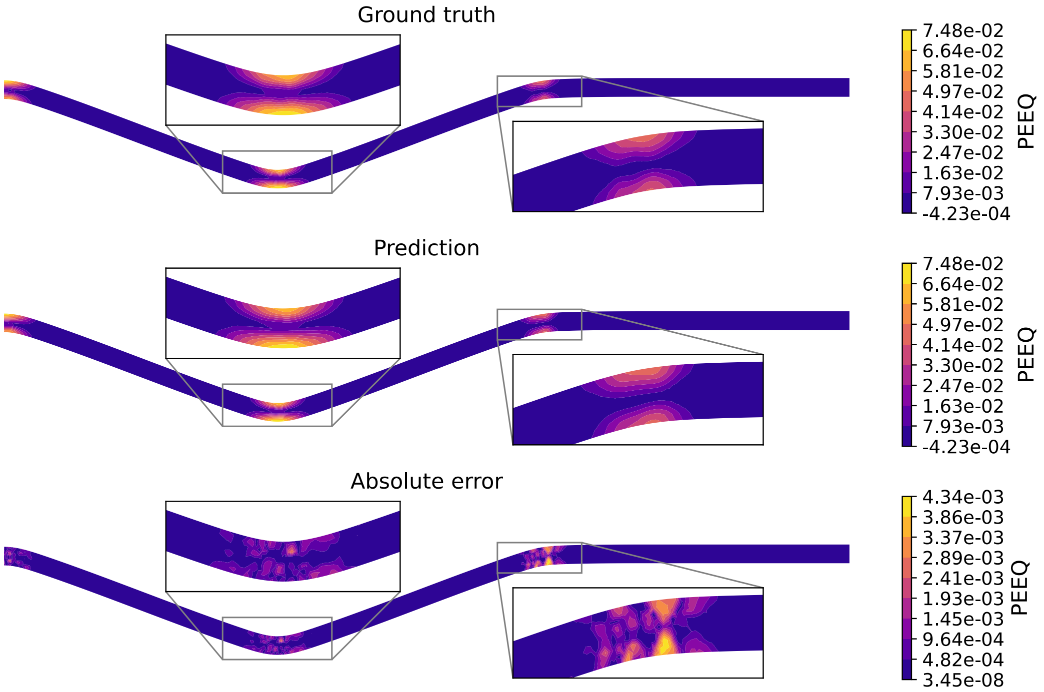


Figure 12: Fringe plot of the *sheet metal forming* dataset (worst source sample). Shown is the ground truth (top) and predicted (middle) PEEQ, as well as the absolute error (bottom).

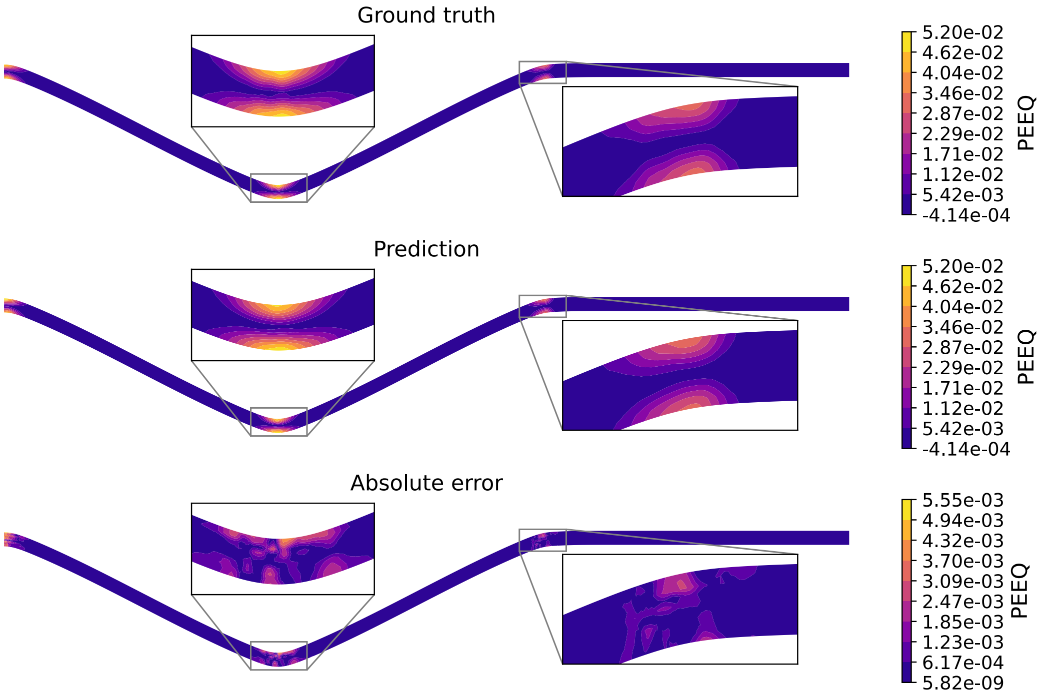


Figure 13: Fringe plot of the *sheet metal forming* dataset (best target sample). Shown is the ground truth (top) and predicted (middle) PEEQ, as well as the absolute error (bottom).

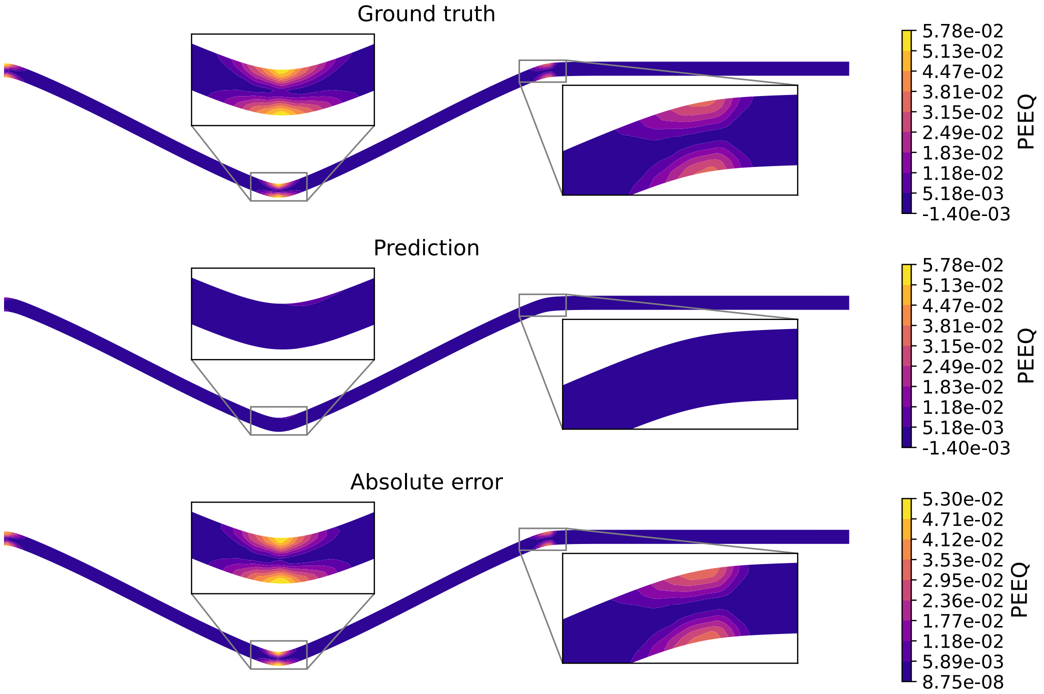


Figure 14: Fringe plot of the *sheet metal forming* dataset (worst target sample). Shown is the ground truth (top) and predicted (middle) PEEQ, as well as the absolute error (bottom).

### B.3 ELECTRIC MOTOR DESIGN

Table 8 presents the complete benchmarking results for the *electric motor design* dataset. For this dataset the relative degradation in model performance in the target domain is in general smaller than in the previous two presented above.

Table 8: RMSE (mean ± std over 4 seeds) on the *electric motor design* dataset at *medium* difficulty. Values are target domain errors (lower is better). Bold marks the overall best model + UDA algorithm + model selection combination. For each architecture, the unregularized baseline row is shaded beige, whereas the best UDA + selection within that architecture is underlined and shaded green.

| Model | DA Algorithm | Model Selection | All Fields Normalized Avg (-) | | Deformation (m) | | Logarithmic Strain ($\times 10^{-2}$) | | Principal Strain ($\times 10^{-2}$) | | Cauchy Stress (MPa) | | Mises Stress (MPa) | | Total Strain ($\times 10^{-2}$) | | Rel Custom Error (-) | | VM Consistency (-) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT |
| GraphSAGE | - | - | 0.317(±0.004) | 0.375(±0.006) | 0.002(±0.001) | 0.001(±0.000) | 0.008(±0.000) | 0.010(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 10.806(±0.180) | 12.796(±0.214) | 24.128(±0.736) | 29.458(±0.882) | 0.007(±0.000) | 0.009(±0.000) | 0.420(±0.056) | 0.427(±0.056) | 0.028(±0.001) | 0.031(±0.002) |
| | DANN | DEV | 0.313(±0.023) | 0.443(±0.070) | 0.003(±0.000) | 0.003(±0.001) | 0.008(±0.001) | 0.012(±0.002) | 0.008(±0.001) | 0.012(±0.002) | 11.445(±0.778) | 16.390(±2.734) | 25.642(±1.912) | 38.919(±7.104) | 0.008(±0.001) | 0.011(±0.002) | 0.494(±0.112) | 0.373(±0.056) | 0.035(±0.009) | 0.037(±0.009) |
| | DANN | IWV | 0.290(±0.002) | 0.350(±0.005) | 0.002(±0.000) | 0.002(±0.001) | 0.008(±0.000) | 0.010(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 10.674(±0.078) | 12.877(±0.190) | 23.841(±0.275) | 29.805(±0.503) | 0.007(±0.000) | 0.009(±0.000) | 0.428(±0.026) | 0.327(±0.014) | 0.028(±0.001) | 0.031(±0.001) |
| | DANN | SB | 0.293(±0.001) | 0.347(±0.006) | 0.003(±0.001) | 0.002(±0.001) | 0.008(±0.000) | 0.010(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 10.779(±0.025) | 12.755(±0.221) | 24.195(±0.163) | 29.318(±0.913) | 0.007(±0.000) | 0.009(±0.000) | 0.460(±0.013) | 0.341(±0.017) | 0.031(±0.003) | 0.032(±0.001) |
| | DANN | TB | 0.297(±0.004) | 0.343(±0.006) | 0.002(±0.000) | 0.002(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 10.952(±0.141) | 12.635(±0.247) | 24.660(±0.404) | 29.001(±0.820) | 0.007(±0.000) | 0.009(±0.000) | 0.480(±0.029) | 0.355(±0.035) | 0.031(±0.002) | 0.032(±0.001) |
| | CMD | DEV | 0.298(±0.013) | 0.405(±0.056) | 0.002(±0.000) | 0.002(±0.001) | 0.008(±0.000) | 0.011(±0.002) | 0.008(±0.000) | 0.011(±0.002) | 10.942(±0.492) | 15.031(±2.175) | 24.644(±1.326) | 36.013(±6.184) | 0.007(±0.000) | 0.010(±0.002) | 0.473(±0.046) | 0.353(±0.031) | 0.030(±0.003) | 0.030(±0.003) |
| | CMD | IWV | 0.304(±0.014) | 0.352(±0.007) | 0.002(±0.000) | 0.001(±0.000) | 0.008(±0.000) | 0.010(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 11.174(±0.557) | 12.951(±0.267) | 24.667(±0.760) | 29.811(±1.062) | 0.008(±0.000) | 0.009(±0.000) | 0.432(±0.031) | 0.329(±0.008) | 0.029(±0.000) | 0.031(±0.000) |
| | CMD | SB | 0.293(±0.010) | 0.344(±0.005) | 0.002(±0.000) | 0.001(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 10.746(±0.302) | 12.632(±0.222) | 23.942(±0.597) | 28.918(±0.552) | 0.007(±0.000) | 0.009(±0.000) | 0.414(±0.014) | 0.326(±0.019) | 0.028(±0.001) | 0.030(±0.001) |
| | CMD | TB | 0.295(±0.009) | 0.340(±0.005) | 0.002(±0.001) | 0.002(±0.001) | 0.008(±0.000) | 0.009(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 10.804(±0.258) | 12.510(±0.188) | 24.024(±0.739) | 28.500(±0.660) | 0.007(±0.000) | 0.008(±0.000) | 0.431(±0.045) | 0.336(±0.031) | 0.029(±0.002) | 0.031(±0.002) |
| | DARE-GRAM | DEV | 0.296(±0.018) | 0.356(±0.015) | 0.002(±0.000) | 0.002(±0.001) | 0.008(±0.000) | 0.010(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 10.878(±0.604) | 13.086(±0.487) | 24.301(±1.480) | 30.123(±1.025) | 0.007(±0.000) | 0.009(±0.000) | 0.470(±0.085) | 0.355(±0.071) | 0.031(±0.006) | 0.034(±0.007) |
| | DARE-GRAM | IWV | 0.288(±0.003) | 0.346(±0.004) | 0.002(±0.000) | 0.002(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 0.007(±0.000) | 0.009(±0.000) | 10.575(±0.094) | 12.708(±0.174) | 23.562(±0.198) | 29.144(±0.514) | 0.007(±0.000) | 0.009(±0.000) | 0.432(±0.003) | 0.325(±0.013) | 0.028(±0.000) | 0.031(±0.000) |
| | DARE-GRAM | SB | 0.291(±0.006) | 0.342(±0.006) | 0.002(±0.000) | 0.002(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 10.699(±0.245) | 12.582(±0.225) | 23.944(±0.595) | 29.088(±0.589) | 0.007(±0.000) | 0.009(±0.000) | 0.466(±0.023) | 0.349(±0.023) | 0.029(±0.001) | 0.031(±0.001) |
| | DARE-GRAM | TB | 0.289(±0.005) | 0.342(±0.006) | 0.002(±0.000) | 0.002(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 10.637(±0.187) | 12.583(±0.227) | 23.830(±0.496) | 29.114(±0.629) | 0.007(±0.000) | 0.009(±0.000) | 0.461(±0.025) | 0.350(±0.022) | 0.029(±0.001) | 0.031(±0.001) |
| | Deep Coral | DEV | 0.291(±0.005) | 0.358(±0.005) | 0.002(±0.000) | 0.002(±0.000) | 0.008(±0.000) | 0.010(±0.000) | 0.008(±0.000) | 0.010(±0.000) | 10.704(±0.177) | 13.188(±0.183) | 23.821(±0.477) | 30.940(±0.464) | 0.007(±0.000) | 0.009(±0.000) | 0.442(±0.018) | 0.333(±0.017) | 0.029(±0.002) | 0.031(±0.001) |
| | Deep Coral | IWV | 0.298(±0.010) | 0.352(±0.017) | 0.002(±0.001) | 0.002(±0.001) | 0.008(±0.000) | 0.010(±0.001) | 0.008(±0.000) | 0.009(±0.001) | 10.945(±0.381) | 12.924(±0.625) | 24.362(±1.045) | 29.779(±1.865) | 0.007(±0.000) | 0.009(±0.000) | 0.454(±0.069) | 0.350(±0.051) | 0.030(±0.002) | 0.032(±0.003) |
| | Deep Coral | SB | 0.288(±0.004) | 0.351(±0.008) | 0.003(±0.000) | 0.002(±0.001) | 0.008(±0.000) | 0.010(±0.000) | 0.007(±0.000) | 0.009(±0.000) | 10.562(±0.129) | 12.943(±0.313) | 23.450(±0.270) | 30.013(±1.011) | 0.007(±0.000) | 0.009(±0.000) | 0.424(±0.017) | 0.327(±0.017) | 0.028(±0.002) | 0.030(±0.001) |
| | Deep Coral | TB | 0.290(±0.003) | 0.338(±0.003) | 0.002(±0.001) | 0.002(±0.001) | 0.008(±0.000) | 0.009(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 10.698(±0.134) | 12.428(±0.119) | 23.983(±0.330) | 28.527(±0.407) | 0.007(±0.000) | 0.008(±0.000) | 0.454(±0.026) | 0.344(±0.021) | 0.029(±0.001) | 0.032(±0.001) |
| PointNet | - | - | 0.319(±0.050) | 0.396(±0.048) | 0.002(±0.001) | 0.002(±0.001) | 0.008(±0.001) | 0.010(±0.001) | 0.008(±0.001) | 0.010(±0.001) | 10.731(±1.626) | 13.417(±1.584) | 23.666(±3.655) | 30.654(±3.391) | 0.007(±0.001) | 0.009(±0.001) | 0.337(±0.128) | 0.344(±0.124) | 0.042(±0.005) | 0.046(±0.006) |
| | DANN | DEV | 0.287(±0.048) | 0.473(±0.075) | 0.001(±0.000) | 0.001(±0.001) | 0.008(±0.001) | 0.013(±0.002) | 0.007(±0.001) | 0.013(±0.002) | 10.412(±1.731) | 17.591(±3.004) | 22.771(±3.792) | 41.444(±7.989) | 0.007(±0.001) | 0.012(±0.002) | 0.324(±0.116) | 0.300(±0.086) | 0.042(±0.005) | 0.046(±0.005) |
| | DANN | IWV | 0.284(±0.050) | 0.443(±0.091) | 0.002(±0.001) | 0.001(±0.001) | 0.008(±0.001) | 0.012(±0.003) | 0.007(±0.001) | 0.012(±0.003) | 10.308(±1.757) | 16.452(±3.556) | 22.557(±3.813) | 38.760(±8.960) | 0.007(±0.001) | 0.011(±0.002) | 0.315(±0.121) | 0.287(±0.098) | 0.042(±0.005) | 0.045(±0.006) |
| | DANN | SB | 0.269(±0.037) | 0.434(±0.121) | 0.002(±0.001) | 0.002(±0.001) | 0.007(±0.001) | 0.012(±0.003) | 0.007(±0.001) | 0.012(±0.003) | 9.766(±1.273) | 16.083(±4.555) | 21.424(±2.685) | 37.862(±10.930) | 0.007(±0.001) | 0.011(±0.003) | 0.285(±0.092) | 0.258(±0.098) | 0.039(±0.003) | 0.043(±0.005) |
| | DANN | TB | 0.279(±0.053) | 0.343(±0.052) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.001) | 0.009(±0.001) | 0.007(±0.001) | 0.009(±0.001) | 10.135(±1.891) | 12.476(±1.841) | 22.144(±4.215) | 28.299(±3.982) | 0.007(±0.001) | 0.009(±0.001) | 0.309(±0.110) | 0.245(±0.091) | 0.041(±0.005) | 0.045(±0.005) |
| | CMD | DEV | 0.492(±0.439) | 0.395(±0.063) | 0.006(±0.009) | 0.002(±0.001) | 0.014(±0.013) | 0.011(±0.002) | 0.012(±0.010) | 0.011(±0.002) | 18.695(±17.506) | 14.655(±2.354) | 36.503(±29.917) | 34.151(±5.720) | 0.013(±0.012) | 0.010(±0.002) | 0.372(±0.172) | 0.262(±0.088) | 0.045(±0.007) | 0.045(±0.006) |
| | CMD | IWV | 0.470(±0.454) | 0.353(±0.073) | 0.006(±0.009) | 0.002(±0.001) | 0.013(±0.014) | 0.010(±0.002) | 0.011(±0.010) | 0.009(±0.002) | 17.825(±18.057) | 12.958(±2.781) | 34.334(±31.237) | 29.373(±6.774) | 0.012(±0.013) | 0.009(±0.002) | 0.343(±0.155) | 0.239(±0.060) | 0.043(±0.004) | 0.044(±0.004) |
| | CMD | SB | 0.471(±0.453) | 0.353(±0.073) | 0.006(±0.009) | 0.002(±0.001) | 0.014(±0.014) | 0.010(±0.002) | 0.011(±0.010) | 0.009(±0.002) | 17.886(±18.026) | 13.027(±2.803) | 34.519(±31.151) | 29.782(±6.876) | 0.012(±0.013) | 0.009(±0.002) | 0.341(±0.154) | 0.232(±0.050) | 0.043(±0.004) | 0.044(±0.003) |
| | CMD | TB | 0.252(±0.048) | 0.314(±0.059) | 0.002(±0.000) | 0.001(±0.000) | 0.007(±0.001) | 0.009(±0.002) | 0.006(±0.001) | 0.008(±0.002) | 9.125(±1.689) | 11.408(±2.097) | 19.744(±3.712) | 25.760(±5.145) | 0.006(±0.001) | 0.008(±0.001) | 0.269(±0.070) | 0.217(±0.055) | 0.040(±0.004) | 0.043(±0.003) |
| | DARE-GRAM | DEV | 0.252(±0.017) | 0.320(±0.017) | 0.002(±0.000) | 0.001(±0.000) | 0.007(±0.000) | 0.009(±0.000) | 0.006(±0.000) | 0.009(±0.000) | 9.115(±0.492) | 11.648(±0.547) | 19.810(±1.023) | 26.635(±1.545) | 0.006(±0.000) | 0.008(±0.000) | 0.248(±0.027) | 0.201(±0.027) | 0.039(±0.002) | 0.043(±0.003) |
| | DARE-GRAM | IWV | 0.252(±0.017) | 0.320(±0.017) | 0.002(±0.000) | 0.001(±0.000) | 0.007(±0.000) | 0.009(±0.000) | 0.006(±0.000) | 0.009(±0.000) | 9.115(±0.492) | 11.648(±0.547) | 19.810(±1.023) | 26.635(±1.545) | 0.006(±0.000) | 0.008(±0.000) | 0.248(±0.027) | 0.201(±0.027) | 0.039(±0.002) | 0.043(±0.003) |
| | DARE-GRAM | SB | 0.250(±0.020) | 0.314(±0.023) | 0.002(±0.001) | 0.001(±0.001) | 0.007(±0.000) | 0.009(±0.001) | 0.006(±0.000) | 0.008(±0.001) | 9.005(±0.630) | 11.419(±0.771) | 19.737(±1.135) | 26.229(±1.674) | 0.006(±0.000) | 0.008(±0.000) | 0.248(±0.027) | 0.193(±0.035) | 0.038(±0.003) | 0.042(±0.004) |
| | DARE-GRAM | TB | 0.285(±0.082) | 0.313(±0.023) | 0.003(±0.001) | 0.002(±0.001) | 0.008(±0.003) | 0.009(±0.001) | 0.007(±0.002) | 0.008(±0.001) | 10.473(±3.305) | 11.361(±0.742) | 22.230(±5.583) | 26.037(±1.607) | 0.007(±0.002) | 0.008(±0.000) | 0.244(±0.029) | 0.188(±0.034) | 0.038(±0.003) | 0.041(±0.004) |
| | Deep Coral | DEV | 0.259(±0.033) | 0.318(±0.012) | 0.002(±0.001) | 0.002(±0.001) | 0.007(±0.001) | 0.009(±0.000) | 0.007(±0.001) | 0.009(±0.000) | 9.334(±1.065) | 11.551(±0.391) | 20.385(±2.250) | 26.522(±0.984) | 0.006(±0.001) | 0.008(±0.000) | 0.265(±0.031) | 0.199(±0.025) | 0.039(±0.002) | 0.044(±0.003) |
| | Deep Coral | IWV | 0.259(±0.033) | 0.318(±0.012) | 0.002(±0.001) | 0.002(±0.001) | 0.007(±0.001) | 0.009(±0.000) | 0.007(±0.001) | 0.009(±0.000) | 9.334(±1.065) | 11.551(±0.391) | 20.385(±2.250) | 26.522(±0.984) | 0.006(±0.001) | 0.008(±0.000) | 0.265(±0.031) | 0.199(±0.025) | 0.039(±0.002) | 0.044(±0.003) |
| | Deep Coral | SB | 0.255(±0.035) | 0.313(±0.017) | 0.002(±0.001) | 0.002(±0.001) | 0.007(±0.001) | 0.009(±0.000) | 0.006(±0.001) | 0.008(±0.000) | 9.205(±1.131) | 11.382(±0.565) | 20.156(±2.363) | 26.229(±1.375) | 0.006(±0.001) | 0.008(±0.000) | 0.262(±0.033) | 0.197(±0.027) | 0.038(±0.002) | 0.043(±0.003) |
| | Deep Coral | TB | 0.255(±0.035) | 0.313(±0.017) | 0.002(±0.001) | 0.002(±0.001) | 0.007(±0.001) | 0.009(±0.000) | 0.006(±0.001) | 0.008(±0.000) | 9.205(±1.131) | 11.382(±0.565) | 20.156(±2.363) | 26.229(±1.375) | 0.006(±0.001) | 0.008(±0.000) | 0.262(±0.033) | 0.197(±0.027) | 0.038(±0.002) | 0.043(±0.003) |
| **Transolver** | - | - | 0.100(±0.011) | 0.116(±0.006) | 0.002(±0.001) | 0.001(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 3.318(±0.377) | 3.829(±0.201) | 6.699(±0.574) | 7.998(±0.432) | 0.002(±0.000) | 0.002(±0.000) | 0.108(±0.009) | 0.111(±0.008) | 0.016(±0.001) | 0.018(±0.001) |
| | DANN | DEV | 0.083(±0.003) | 0.103(±0.005) | 0.001(±0.000) | 0.001(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 2.959(±0.111) | 3.646(±0.160) | 6.068(±0.237) | 7.607(±0.344) | 0.002(±0.000) | 0.002(±0.000) | 0.101(±0.004) | 0.087(±0.004) | 0.014(±0.001) | 0.017(±0.001) |
| | DANN | IWV | 0.083(±0.001) | 0.104(±0.004) | 0.001(±0.000) | 0.001(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 2.978(±0.037) | 3.673(±0.110) | 6.143(±0.039) | 7.675(±0.263) | 0.002(±0.000) | 0.002(±0.000) | 0.099(±0.003) | 0.087(±0.003) | 0.014(±0.001) | 0.017(±0.001) |
| | DANN | SB | 0.082(±0.002) | 0.105(±0.009) | 0.001(±0.000) | 0.001(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 2.917(±0.078) | 3.751(±0.328) | 5.994(±0.187) | 7.863(±0.735) | 0.002(±0.000) | 0.002(±0.000) | 0.104(±0.005) | 0.088(±0.005) | 0.014(±0.000) | 0.017(±0.001) |
| | DANN | TB | 0.081(±0.002) | 0.100(±0.003) | 0.001(±0.000) | 0.001(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 2.900(±0.077) | 3.576(±0.102) | 5.958(±0.195) | 7.472(±0.210) | 0.002(±0.000) | 0.002(±0.000) | 0.103(±0.004) | 0.086(±0.002) | 0.014(±0.000) | 0.016(±0.000) |
| | CMD | DEV | 0.083(±0.002) | 0.107(±0.016) | 0.001(±0.000) | 0.001(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 2.976(±0.065) | 3.829(±0.590) | 6.093(±0.179) | 8.100(±1.540) | 0.002(±0.000) | 0.003(±0.000) | 0.105(±0.007) | 0.089(±0.004) | 0.014(±0.000) | 0.017(±0.002) |
| | CMD | IWV | 0.085(±0.002) | 0.103(±0.005) | 0.001(±0.000) | 0.001(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 3.038(±0.083) | 3.691(±0.183) | 6.230(±0.209) | 7.695(±0.407) | 0.002(±0.000) | 0.002(±0.000) | 0.104(±0.006) | 0.088(±0.004) | 0.015(±0.001) | 0.017(±0.001) |
| | CMD | SB | 0.161(±0.160) | 0.102(±0.008) | 0.002(±0.002) | 0.001(±0.000) | 0.004(±0.004) | 0.003(±0.000) | 0.004(±0.004) | 0.003(±0.000) | 5.217(±4.636) | 3.607(±0.224) | 13.116(±14.355) | 7.484(±0.441) | 0.004(±0.003) | 0.002(±0.000) | 0.154(±0.104) | 0.089(±0.002) | 0.015(±0.002) | 0.017(±0.003) |
| | CMD | TB | 0.082(±0.002) | 0.099(±0.003) | 0.001(±0.000) | 0.001(±0.000) | 0.002(±0.000) | 0.002(±0.000) | 0.002(±0.000) | 0.002(±0.000) | 2.924(±0.082) | 3.523(±0.110) | 5.981(±0.177) | 7.308(±0.229) | 0.002(±0.000) | 0.002(±0.000) | 0.105(±0.006) | 0.091(±0.004) | 0.014(±0.001) | 0.016(±0.001) |
| | DARE-GRAM | DEV | 0.085(±0.002) | 0.105(±0.006) | 0.001(±0.001) | 0.001(±0.001) | 0.002(±0.000) | 0.003(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 3.056(±0.077) | 3.743(±0.219) | 6.304(±0.178) | 7.834(±0.435) | 0.002(±0.000) | 0.002(±0.000) | 0.107(±0.005) | 0.092(±0.004) | 0.015(±0.001) | 0.017(±0.001) |
| | DARE-GRAM | IWV | 0.084(±0.002) | 0.102(±0.002) | 0.001(±0.001) | 0.001(±0.001) | 0.002(±0.000) | 0.003(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 3.016(±0.062) | 3.636(±0.081) | 6.209(±0.151) | 7.603(±0.153) | 0.002(±0.000) | 0.002(±0.000) | 0.105(±0.006) | 0.090(±0.004) | 0.014(±0.000) | 0.017(±0.001) |
| | DARE-GRAM | SB | 0.082(±0.001) | 0.106(±0.014) | 0.001(±0.000) | 0.002(±0.002) | 0.002(±0.000) | 0.003(±0.000) | 0.002(±0.000) | 0.003(±0.001) | 2.931(±0.022) | 3.776(±0.488) | 6.006(±0.070) | 8.577(±2.411) | 0.002(±0.000) | 0.002(±0.000) | 0.105(±0.004) | 0.095(±0.013) | 0.014(±0.001) | 0.017(±0.002) |
| | DARE-GRAM | TB | 0.083(±0.002) | 0.099(±0.002) | 0.001(±0.000) | 0.001(±0.000) | 0.002(±0.000) | 0.002(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 2.989(±0.063) | 3.543(±0.071) | 6.167(±0.164) | 7.389(±0.202) | 0.002(±0.000) | 0.002(±0.000) | 0.108(±0.003) | 0.090(±0.004) | 0.015(±0.000) | 0.017(±0.001) |
| | Deep Coral | DEV | 0.156(±0.145) | 0.099(±0.002) | 0.002(±0.002) | 0.001(±0.000) | 0.003(±0.002) | 0.002(±0.000) | 0.003(±0.003) | 0.002(±0.000) | 4.560(±3.191) | 3.517(±0.079) | 11.217(±10.266) | 7.308(±0.201) | 0.003(±0.001) | 0.002(±0.000) | 0.141(±0.066) | 0.090(±0.002) | 0.015(±0.002) | 0.016(±0.001) |
| | Deep Coral | IWV | 0.158(±0.144) | 0.102(±0.007) | 0.002(±0.002) | 0.001(±0.000) | 0.003(±0.002) | 0.003(±0.000) | 0.003(±0.003) | 0.003(±0.000) | 4.632(±3.145) | 3.629(±0.230) | 11.370(±10.167) | 7.600(±0.582) | 0.003(±0.001) | 0.002(±0.000) | 0.143(±0.065) | 0.090(±0.006) | 0.015(±0.001) | 0.017(±0.001) |
| | **Deep Coral** | **SB** | 0.082(±0.003) | **0.098(±0.003)** | 0.001(±0.000) | 0.001(±0.000) | 0.002(±0.000) | 0.002(±0.000) | 0.002(±0.000) | 0.002(±0.000) | 2.943(±0.106) | 3.497(±0.127) | 6.039(±0.234) | 7.269(±0.259) | 0.002(±0.000) | 0.002(±0.000) | 0.104(±0.005) | 0.089(±0.003) | 0.014(±0.000) | 0.016(±0.001) |
| | Deep Coral | TB | 0.082(±0.003) | 0.098(±0.003) | 0.001(±0.000) | 0.001(±0.000) | 0.002(±0.000) | 0.002(±0.000) | 0.002(±0.000) | 0.002(±0.000) | 2.944(±0.107) | 3.491(±0.120) | 6.049(±0.245) | 7.266(±0.256) | 0.002(±0.000) | 0.002(±0.000) | 0.104(±0.005) | 0.089(±0.003) | 0.014(±0.000) | 0.016(±0.001) |

To assess the effect of the domain shift on prediction accuracy in the *electric motor design* dataset further, Figure 15 shows the distribution of NRMSEs for the best performing model, selected by lowest average error in the target domain, in the source and target domain.

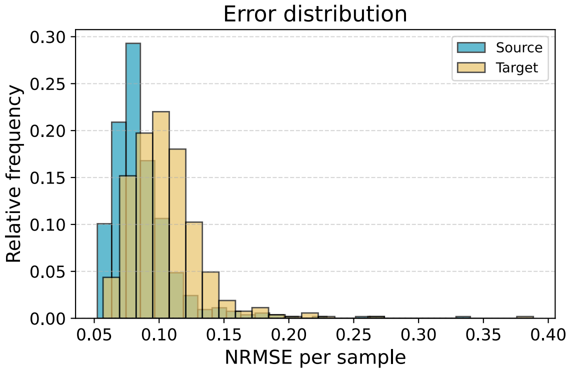


Figure 15: Distribution of NRMSE (averaged across all fields) for the test sets of the source (blue) and target (yellow) domains in the *electric motor design* dataset. Bar height indicates the relative frequency of samples within each bin.

In this task, the Mises stress is used as a scalar summary of the multi-axial stress state and is particularly interesting for downstream analysis and optimization. We therefore focus our closer inspection on this field.

Table 9 presents a comparison of absolute Mises stress errors for the best and worst samples from both the source and target test sets. The corresponding fringe plots are shown in Figures 16 to 19, comparing the ground truth and predicted fields alongside their absolute errors. They show that the best samples are predicted very well, whereas the worst sample of the source domain visually appears slightly worse than the one of the target domain. On average, however, it is still predicted more accurately than the worst sample of the target domain, as shown in Table 9.

Table 9: Absolute error (MPa) of Mises stress predictions for the best and worst samples from the source and target domain of the *electric motor design* dataset. Lowest value per metric is bold.

| **Metric** | **Source** | | **Target** | |
|---|---|---|---|---|
| | **Best** | **Worst** | **Best** | **Worst** |
| Mean | **2.00** | 20.50 | 2.67 | 23.21 |
| Std | **2.73** | 39.50 | 3.13 | 23.32 |
| Median | **1.26** | 13.63 | 1.68 | 13.09 |
| $Q_{01}$ | **0.02** | 0.18 | 0.03 | 0.11 |
| $Q_{25}$ | **0.60** | 5.08 | 0.75 | 2.87 |
| $Q_{75}$ | **2.24** | 24.05 | 3.50 | 41.97 |
| $Q_{99}$ | **13.43** | 140.17 | 15.05 | 78.28 |

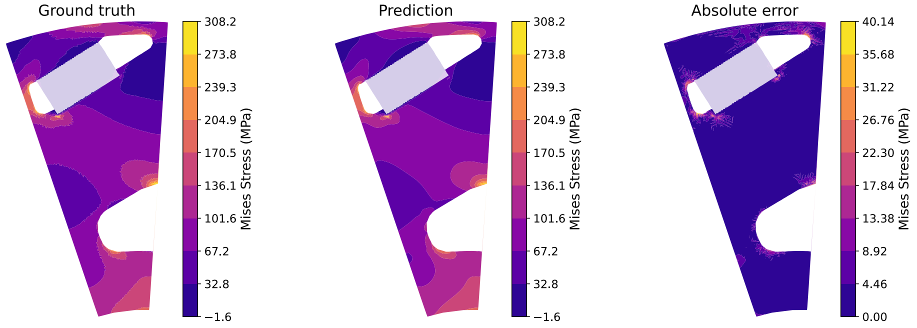


Figure 16: Fringe plot of the *electric motor design* dataset (best source sample). Shown is the ground truth (left) and predicted (middle) Mises stress, as well as the absolute error (right).

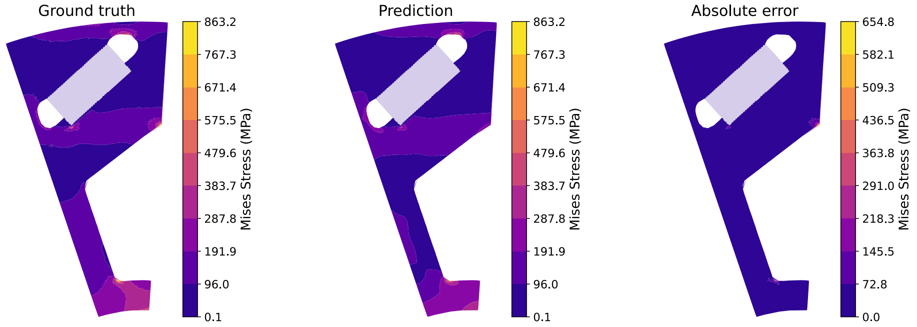


Figure 17: Fringe plot of the *electric motor design* dataset (worst source sample). Shown is the ground truth (left) and predicted (middle) Mises stress, as well as the absolute error (right).

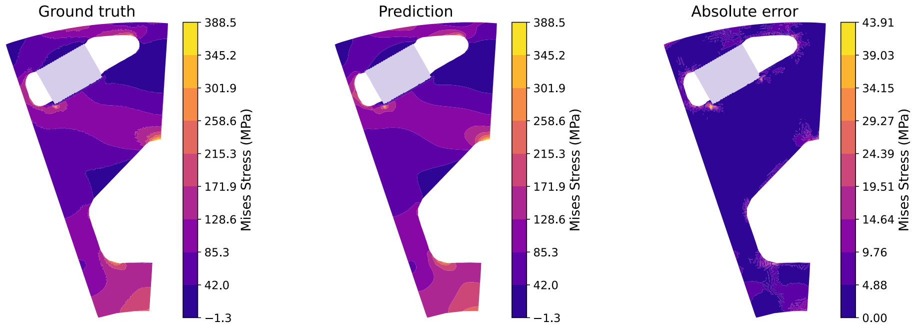


Figure 18: Fringe plot of the *electric motor design* dataset (best target sample). Shown is the ground truth (left) and predicted (middle) Mises stress, as well as the absolute error (right).

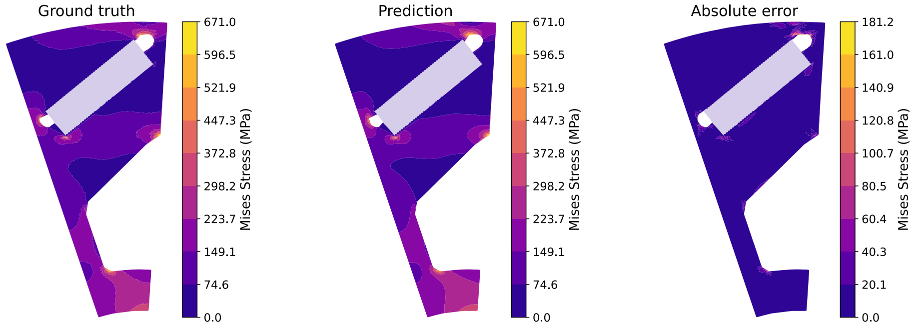


Figure 19: Fringe plot of the *electric motor design* dataset (worst target sample). Shown is the ground truth (left) and predicted (middle) Mises stress, as well as the absolute error (right).

### B.4 HEATSINK DESIGN

Table 10 presents the complete benchmarking results for the *heatsink design* dataset.

Table 10: RMSE (mean ± std over 4 seeds) on the *heatsink* dataset at *medium* difficulty. Values are target domain errors (lower is better). Bold marks the overall best model + UDA algorithm + model selection combination. For each architecture, the unregularized baseline row is shaded beige, whereas the best UDA + selection within that architecture is underlined and shaded green.

| Model | DA Algorithm | Model Selection | All Fields Normalized Avg (-) | | Temperature (K) | | Velocity (m/s) | | Pressure (kPa) | | Rel Custom Error (-) | | BC Violation Temperature (-) | | BC Violation Velocity (m/s) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT |
| PointNet | - | - | 0.525(±0.026) | 0.568(±0.030) | 15.581(±1.535) | 21.126(±2.365) | 0.054(±0.002) | 0.044(±0.000) | 385.602(±34.377) | 1879.390(±239.203) | 0.036(±0.006) | 0.050(±0.009) | 0.257(±0.026) | 0.265(±0.020) | 0.105(±0.007) | 0.110(±0.008) |
| | DANN | DEV | 0.339(±0.104) | 0.442(±0.050) | 12.078(±4.555) | 19.408(±3.391) | 0.043(±0.009) | 0.047(±0.007) | 814.988(±1032.338) | 1997.691(±359.872) | 0.026(±0.014) | 0.041(±0.012) | 0.176(±0.085) | 0.261(±0.042) | 0.088(±0.031) | 0.107(±0.015) |
| | DANN | IWV | 0.289(±0.056) | 0.429(±0.052) | 10.167(±2.894) | 18.172(±3.222) | 0.040(±0.008) | 0.047(±0.007) | 283.202(±70.570) | 1805.938(±144.882) | 0.019(±0.008) | 0.036(±0.009) | 0.145(±0.064) | 0.248(±0.042) | 0.079(±0.026) | 0.114(±0.012) |
| | DANN | SB | 0.228(±0.016) | 0.494(±0.026) | 6.668(±1.013) | 20.129(±2.380) | 0.031(±0.002) | 0.055(±0.002) | 207.063(±13.650) | 2102.712(±614.979) | 0.010(±0.002) | 0.041(±0.008) | 0.063(±0.018) | 0.286(±0.023) | 0.047(±0.010) | 0.128(±0.007) |
| | DANN | TB | 0.304(±0.036) | 0.397(±0.019) | 10.964(±1.411) | 15.719(±1.387) | 0.041(±0.005) | 0.043(±0.002) | 331.207(±140.774) | 1907.682(±232.159) | 0.021(±0.004) | 0.029(±0.006) | 0.159(±0.027) | 0.212(±0.021) | 0.086(±0.011) | 0.114(±0.010) |
| | CMD | DEV | 0.424(±0.003) | 0.443(±0.003) | 16.292(±0.119) | 20.676(±0.141) | 0.042(±0.001) | 0.042(±0.001) | 2388.245(±17.831) | 2464.906(±43.671) | 0.042(±0.000) | 0.048(±0.001) | 0.272(±0.002) | 0.286(±0.009) | 0.080(±0.002) | 0.109(±0.003) |
| | CMD | IWV | 0.239(±0.008) | 0.480(±0.020) | 7.577(±0.479) | 18.524(±1.213) | 0.033(±0.001) | 0.051(±0.002) | 192.803(±4.658) | 2455.172(±118.483) | 0.012(±0.001) | 0.035(±0.003) | 0.082(±0.009) | 0.246(±0.012) | 0.055(±0.004) | 0.130(±0.002) |
| | CMD | SB | 0.238(±0.007) | 0.477(±0.023) | 7.483(±0.406) | 18.666(±1.057) | 0.033(±0.001) | 0.051(±0.002) | 196.144(±5.022) | 2375.674(±157.350) | 0.012(±0.001) | 0.035(±0.003) | 0.080(±0.008) | 0.246(±0.012) | 0.054(±0.003) | 0.129(±0.002) |
| | CMD | TB | 0.302(±0.086) | 0.442(±0.018) | 10.801(±4.087) | 17.800(±2.256) | 0.037(±0.004) | 0.046(±0.004) | 756.939(±1076.860) | 2288.916(±108.201) | 0.022(±0.014) | 0.036(±0.009) | 0.150(±0.087) | 0.234(±0.033) | 0.069(±0.014) | 0.121(±0.010) |
| | DARE-GRAM | DEV | 0.220(±0.009) | 0.393(±0.003) | 6.286(±0.618) | 14.444(±0.800) | 0.030(±0.001) | 0.045(±0.001) | 202.806(±6.781) | 1556.506(±260.868) | 0.009(±0.001) | 0.020(±0.004) | 0.057(±0.015) | 0.215(±0.013) | 0.043(±0.008) | 0.125(±0.002) |
| | DARE-GRAM | IWV | 0.219(±0.009) | 0.378(±0.027) | 6.196(±0.632) | 13.346(±1.525) | 0.030(±0.001) | 0.044(±0.003) | 201.887(±6.192) | 1473.618(±311.372) | 0.009(±0.001) | 0.018(±0.003) | 0.055(±0.015) | 0.210(±0.012) | 0.043(±0.008) | 0.122(±0.007) |
| | DARE-GRAM | SB | 0.214(±0.003) | 0.371(±0.031) | 5.887(±0.161) | 12.343(±1.062) | 0.029(±0.000) | 0.043(±0.004) | 200.738(±7.351) | 1401.808(±235.957) | 0.009(±0.000) | 0.015(±0.002) | 0.049(±0.003) | 0.215(±0.016) | 0.038(±0.002) | 0.121(±0.007) |
| | DARE-GRAM | TB | 0.218(±0.009) | 0.368(±0.027) | 6.155(±0.662) | 12.800(±1.537) | 0.030(±0.001) | 0.042(±0.003) | 202.412(±6.875) | 1398.169(±230.128) | 0.009(±0.001) | 0.018(±0.003) | 0.055(±0.015) | 0.208(±0.013) | 0.042(±0.009) | 0.121(±0.007) |
| | Deep Coral | DEV | 0.275(±0.070) | 0.394(±0.048) | 9.313(±3.544) | 17.959(±2.281) | 0.038(±0.010) | 0.044(±0.006) | 238.834(±83.989) | 1001.401(±504.884) | 0.017(±0.010) | 0.033(±0.012) | 0.125(±0.081) | 0.282(±0.015) | 0.069(±0.028) | 0.109(±0.014) |
| | Deep Coral | IWV | 0.270(±0.061) | 0.394(±0.048) | 9.071(±3.069) | 17.428(±1.939) | 0.037(±0.009) | 0.044(±0.006) | 224.078(±54.512) | 1037.166(±573.817) | 0.016(±0.008) | 0.031(±0.010) | 0.117(±0.067) | 0.276(±0.024) | 0.070(±0.029) | 0.112(±0.010) |
| | Deep Coral | SB | 0.270(±0.061) | 0.394(±0.048) | 9.071(±3.069) | 17.428(±1.939) | 0.037(±0.009) | 0.044(±0.006) | 224.078(±54.512) | 1037.166(±573.817) | 0.016(±0.008) | 0.031(±0.010) | 0.117(±0.067) | 0.276(±0.024) | 0.070(±0.029) | 0.112(±0.010) |
| | Deep Coral | TB | 0.343(±0.063) | 0.384(±0.042) | 12.763(±3.067) | 18.517(±2.502) | 0.047(±0.009) | 0.042(±0.004) | 324.404(±102.861) | 1438.782(±427.051) | 0.027(±0.009) | 0.040(±0.012) | 0.205(±0.075) | 0.276(±0.021) | 0.098(±0.024) | 0.095(±0.011) |
| **Transolver** | - | - | 0.245(±0.004) | 0.446(±0.018) | 4.327(±0.059) | 9.720(±0.272) | 0.024(±0.000) | 0.038(±0.001) | 307.197(±32.557) | 1686.316(±263.085) | 0.007(±0.000) | 0.010(±0.001) | 0.035(±0.002) | 0.083(±0.007) | 0.021(±0.002) | 0.078(±0.004) |
| | DANN | DEV | 0.183(±0.001) | 0.436(±0.011) | 4.318(±0.081) | 15.272(±0.927) | 0.024(±0.000) | 0.049(±0.001) | 302.659(±26.598) | 2032.574(±47.931) | 0.007(±0.000) | 0.023(±0.002) | 0.035(±0.004) | 0.211(±0.008) | 0.021(±0.001) | 0.131(±0.002) |
| | DANN | IWV | 0.213(±0.056) | 0.425(±0.023) | 5.905(±2.938) | 14.712(±1.183) | 0.028(±0.007) | 0.047(±0.004) | 374.773(±149.925) | 2164.583(±190.783) | 0.011(±0.006) | 0.023(±0.002) | 0.067(±0.060) | 0.204(±0.007) | 0.036(±0.027) | 0.128(±0.005) |
| | DANN | SB | 0.183(±0.002) | 0.422(±0.015) | 4.335(±0.077) | 13.936(±1.133) | 0.024(±0.000) | 0.047(±0.002) | 294.823(±24.069) | 2100.219(±65.406) | 0.007(±0.000) | 0.021(±0.002) | 0.035(±0.005) | 0.193(±0.010) | 0.021(±0.001) | 0.131(±0.002) |
| | DANN | TB | 0.242(±0.068) | 0.406(±0.024) | 7.477(±3.603) | 14.114(±0.341) | 0.032(±0.009) | 0.045(±0.003) | 436.249(±137.337) | 2067.819(±155.414) | 0.014(±0.008) | 0.023(±0.003) | 0.093(±0.066) | 0.207(±0.006) | 0.051(±0.035) | 0.125(±0.004) |
| | CMD | DEV | 0.312(±0.088) | 0.437(±0.041) | 11.189(±4.591) | 16.986(±3.498) | 0.029(±0.004) | 0.043(±0.002) | 1886.108(±1078.918) | 2700.699(±448.359) | 0.030(±0.015) | 0.038(±0.014) | 0.233(±0.136) | 0.232(±0.040) | 0.036(±0.014) | 0.116(±0.006) |
| | CMD | IWV | 0.181(±0.001) | 0.375(±0.032) | 4.266(±0.139) | 11.923(±2.486) | 0.024(±0.000) | 0.042(±0.004) | 274.272(±9.637) | 1816.616(±31.411) | 0.007(±0.000) | 0.015(±0.006) | 0.031(±0.004) | 0.181(±0.012) | 0.021(±0.002) | 0.124(±0.007) |
| | CMD | SB | 0.180(±0.001) | 0.413(±0.044) | 4.281(±0.109) | 13.836(±2.825) | 0.024(±0.000) | 0.047(±0.005) | 300.372(±28.729) | 1960.714(±99.356) | 0.007(±0.000) | 0.021(±0.008) | 0.034(±0.003) | 0.189(±0.015) | 0.021(±0.002) | 0.126(±0.005) |
| | CMD | TB | 0.180(±0.002) | 0.367(±0.025) | 4.232(±0.135) | 11.217(±2.086) | 0.024(±0.000) | 0.041(±0.003) | 260.544(±21.474) | 1894.720(±185.168) | 0.007(±0.000) | 0.014(±0.005) | 0.030(±0.003) | 0.176(±0.008) | 0.021(±0.002) | 0.124(±0.007) |
| | DARE-GRAM | DEV | 0.181(±0.003) | 0.346(±0.036) | 4.255(±0.051) | 10.142(±1.074) | 0.024(±0.000) | 0.039(±0.003) | 292.275(±21.590) | 1725.831(±344.902) | 0.007(±0.000) | 0.010(±0.001) | 0.034(±0.002) | 0.173(±0.013) | 0.021(±0.001) | 0.119(±0.002) |
| | DARE-GRAM | IWV | 0.181(±0.003) | 0.333(±0.030) | 4.290(±0.086) | 9.856(±1.025) | 0.024(±0.000) | 0.038(±0.003) | 280.252(±13.840) | 1627.061(±277.975) | 0.007(±0.000) | 0.011(±0.001) | 0.035(±0.002) | 0.174(±0.016) | 0.023(±0.002) | 0.118(±0.003) |
| | DARE-GRAM | SB | 0.180(±0.003) | 0.354(±0.026) | 4.268(±0.127) | 10.753(±1.187) | 0.024(±0.000) | 0.040(±0.003) | 285.610(±15.083) | 1768.578(±187.647) | 0.007(±0.000) | 0.012(±0.003) | 0.033(±0.003) | 0.176(±0.002) | 0.022(±0.002) | 0.120(±0.006) |
| | DARE-GRAM | TB | 0.182(±0.003) | 0.332(±0.028) | 4.345(±0.064) | 9.840(±0.983) | 0.024(±0.000) | 0.038(±0.003) | 282.034(±17.742) | 1649.841(±294.017) | 0.007(±0.000) | 0.010(±0.001) | 0.034(±0.002) | 0.171(±0.011) | 0.023(±0.001) | 0.119(±0.003) |
| | **Deep Coral** | **DEV** | 0.181(±0.001) | **0.318(±0.027)** | 4.251(±0.073) | 9.081(±0.980) | 0.024(±0.000) | 0.037(±0.002) | 284.773(±13.081) | 1425.648(±323.586) | 0.007(±0.000) | 0.009(±0.002) | 0.031(±0.003) | 0.170(±0.010) | 0.022(±0.001) | 0.117(±0.004) |
| | Deep Coral | IWV | 0.182(±0.001) | 0.337(±0.010) | 4.309(±0.037) | 9.993(±0.208) | 0.024(±0.000) | 0.038(±0.000) | 287.654(±13.792) | 1663.737(±219.883) | 0.007(±0.000) | 0.010(±0.001) | 0.034(±0.002) | 0.183(±0.002) | 0.022(±0.001) | 0.122(±0.003) |
| | Deep Coral | SB | 0.180(±0.002) | 0.341(±0.021) | 4.269(±0.039) | 9.982(±0.946) | 0.024(±0.000) | 0.039(±0.002) | 284.878(±11.282) | 1712.116(±272.961) | 0.007(±0.000) | 0.010(±0.002) | 0.032(±0.001) | 0.181(±0.011) | 0.022(±0.001) | 0.119(±0.003) |
| | Deep Coral | TB | 0.181(±0.001) | 0.310(±0.020) | 4.253(±0.073) | 8.718(±0.833) | 0.024(±0.000) | 0.037(±0.002) | 290.683(±22.141) | 1334.387(±174.740) | 0.007(±0.000) | 0.009(±0.002) | 0.033(±0.005) | 0.166(±0.002) | 0.021(±0.001) | 0.117(±0.004) |
| UPT | - | - | 0.244(±0.002) | 0.441(±0.024) | 4.316(±0.028) | 13.033(±1.059) | 0.025(±0.000) | 0.040(±0.002) | 231.716(±14.316) | 816.181(±48.995) | 0.006(±0.000) | 0.013(±0.004) | 0.028(±0.002) | 0.127(±0.025) | 0.026(±0.002) | 0.071(±0.006) |
| | DANN | DEV | 0.188(±0.011) | 0.446(±0.026) | 4.651(±0.781) | 15.580(±0.609) | 0.026(±0.002) | 0.050(±0.003) | 222.968(±13.096) | 2164.527(±302.095) | 0.007(±0.002) | 0.024(±0.003) | 0.034(±0.018) | 0.215(±0.013) | 0.032(±0.014) | 0.122(±0.003) |
| | DANN | IWV | 0.194(±0.012) | 0.468(±0.038) | 5.039(±0.887) | 15.283(±1.155) | 0.026(±0.002) | 0.050(±0.002) | 241.463(±18.499) | 2693.651(±570.114) | 0.008(±0.002) | 0.023(±0.007) | 0.044(±0.019) | 0.222(±0.017) | 0.039(±0.014) | 0.122(±0.002) |
| | DANN | SB | 0.184(±0.002) | 0.480(±0.018) | 4.285(±0.072) | 15.689(±0.806) | 0.025(±0.000) | 0.051(±0.001) | 243.566(±23.766) | 2729.453(±516.783) | 0.006(±0.000) | 0.023(±0.003) | 0.025(±0.003) | 0.225(±0.005) | 0.025(±0.002) | 0.125(±0.003) |
| | DANN | TB | 0.273(±0.092) | 0.398(±0.038) | 9.411(±4.841) | 15.644(±3.334) | 0.037(±0.012) | 0.043(±0.004) | 285.494(±72.700) | 1872.493(±365.646) | 0.019(±0.013) | 0.027(±0.011) | 0.136(±0.096) | 0.222(±0.040) | 0.074(±0.032) | 0.108(±0.018) |
| | CMD | DEV | 0.210(±0.055) | 0.406(±0.046) | 5.994(±3.353) | 14.289(±2.054) | 0.028(±0.007) | 0.046(±0.005) | 236.012(±21.869) | 1873.691(±394.369) | 0.010(±0.008) | 0.021(±0.006) | 0.060(±0.061) | 0.202(±0.019) | 0.039(±0.025) | 0.123(±0.005) |
| | CMD | IWV | 0.182(±0.000) | 0.363(±0.015) | 4.297(±0.038) | 12.908(±0.487) | 0.025(±0.000) | 0.043(±0.001) | 221.371(±9.451) | 1365.372(±256.867) | 0.006(±0.000) | 0.017(±0.001) | 0.028(±0.003) | 0.204(±0.020) | 0.026(±0.001) | 0.118(±0.001) |
| | CMD | SB | 0.179(±0.001) | 0.444(±0.010) | 4.135(±0.026) | 16.130(±0.627) | 0.024(±0.000) | 0.050(±0.001) | 230.955(±7.948) | 1918.677(±52.455) | 0.006(±0.000) | 0.025(±0.002) | 0.022(±0.001) | 0.220(±0.006) | 0.022(±0.000) | 0.126(±0.004) |
| | CMD | TB | 0.182(±0.000) | 0.363(±0.015) | 4.297(±0.038) | 12.908(±0.487) | 0.025(±0.000) | 0.043(±0.001) | 221.371(±9.451) | 1365.372(±256.867) | 0.006(±0.000) | 0.017(±0.001) | 0.028(±0.003) | 0.204(±0.020) | 0.026(±0.001) | 0.118(±0.001) |
| | DARE-GRAM | DEV | 0.181(±0.001) | 0.335(±0.006) | 4.215(±0.052) | 11.291(±0.774) | 0.024(±0.000) | 0.039(±0.001) | 236.908(±22.992) | 1079.205(±251.854) | 0.006(±0.000) | 0.008(±0.002) | 0.025(±0.003) | 0.227(±0.017) | 0.025(±0.002) | 0.115(±0.003) |
| | DARE-GRAM | IWV | 0.182(±0.001) | 0.349(±0.006) | 4.271(±0.033) | 13.077(±0.930) | 0.025(±0.000) | 0.041(±0.001) | 225.229(±2.040) | 1007.086(±192.883) | 0.006(±0.000) | 0.011(±0.003) | 0.026(±0.003) | 0.250(±0.017) | 0.025(±0.001) | 0.113(±0.004) |
| | DARE-GRAM | SB | 0.182(±0.001) | 0.331(±0.013) | 4.310(±0.113) | 12.065(±1.734) | 0.025(±0.000) | 0.039(±0.002) | 242.949(±17.732) | 1050.490(±240.188) | 0.006(±0.000) | 0.011(±0.003) | 0.028(±0.002) | 0.233(±0.021) | 0.026(±0.002) | 0.113(±0.004) |
| | DARE-GRAM | TB | 0.181(±0.001) | 0.326(±0.009) | 4.261(±0.117) | 12.270(±0.971) | 0.025(±0.000) | 0.039(±0.002) | 229.463(±15.792) | 867.115(±236.837) | 0.006(±0.000) | 0.010(±0.003) | 0.025(±0.004) | 0.242(±0.006) | 0.025(±0.002) | 0.110(±0.005) |
| | Deep Coral | DEV | 0.183(±0.001) | 0.345(±0.013) | 4.318(±0.067) | 13.290(±0.655) | 0.025(±0.000) | 0.041(±0.001) | 221.346(±8.107) | 809.557(±99.406) | 0.006(±0.000) | 0.010(±0.002) | 0.028(±0.002) | 0.244(±0.010) | 0.026(±0.001) | 0.109(±0.003) |
| | Deep Coral | IWV | 0.183(±0.001) | 0.339(±0.020) | 4.344(±0.055) | 13.037(±1.027) | 0.025(±0.000) | 0.041(±0.002) | 223.375(±7.211) | 778.306(±64.952) | 0.006(±0.000) | 0.010(±0.002) | 0.028(±0.002) | 0.244(±0.009) | 0.026(±0.001) | 0.110(±0.002) |
| | Deep Coral | SB | 0.182(±0.000) | 0.325(±0.008) | 4.307(±0.042) | 12.414(±1.209) | 0.025(±0.000) | 0.039(±0.001) | 214.055(±7.216) | 840.409(±183.867) | 0.006(±0.000) | 0.013(±0.007) | 0.027(±0.004) | 0.236(±0.012) | 0.027(±0.001) | 0.107(±0.007) |
| | Deep Coral | TB | 0.182(±0.000) | 0.321(±0.008) | 4.347(±0.039) | 12.637(±0.949) | 0.025(±0.000) | 0.039(±0.001) | 218.167(±11.586) | 791.889(±121.634) | 0.006(±0.000) | 0.013(±0.007) | 0.028(±0.004) | 0.235(±0.015) | 0.027(±0.001) | 0.107(±0.007) |
| GINO | - | - | 0.336(±0.011) | 0.484(±0.023) | 8.539(±0.477) | 13.895(±0.204) | 0.034(±0.001) | 0.042(±0.002) | 278.161(±12.622) | 1079.753(±152.799) | 0.014(±0.001) | 0.016(±0.002) | 0.092(±0.009) | 0.137(±0.007) | 0.059(±0.005) | 0.083(±0.004) |
| | DANN | DEV | 0.253(±0.010) | 0.486(±0.026) | 8.465(±0.532) | 18.087(±2.490) | 0.034(±0.001) | 0.052(±0.002) | 292.040(±18.691) | 2431.868(±734.757) | 0.014(±0.001) | 0.034(±0.010) | 0.094(±0.006) | 0.261(±0.018) | 0.060(±0.003) | 0.125(±0.004) |
| | DANN | IWV | 0.253(±0.010) | 0.487(±0.023) | 8.481(±0.542) | 18.827(±2.718) | 0.034(±0.001) | 0.053(±0.003) | 287.835(±16.154) | 2247.152(±687.350) | 0.014(±0.001) | 0.035(±0.010) | 0.095(±0.006) | 0.271(±0.020) | 0.061(±0.004) | 0.127(±0.005) |
| | DANN | SB | 0.250(±0.009) | 0.490(±0.025) | 8.303(±0.502) | 19.127(±1.534) | 0.034(±0.001) | 0.054(±0.001) | 283.814(±14.682) | 2210.682(±649.838) | 0.014(±0.001) | 0.037(±0.005) | 0.091(±0.008) | 0.271(±0.013) | 0.059(±0.004) | 0.126(±0.004) |
| | DANN | TB | 0.265(±0.019) | 0.452(±0.025) | 9.122(±1.029) | 17.060(±1.705) | 0.035(±0.002) | 0.049(±0.001) | 335.654(±60.476) | 2141.697(±662.675) | 0.015(±0.002) | 0.031(±0.009) | 0.105(±0.018) | 0.249(±0.012) | 0.066(±0.009) | 0.123(±0.004) |
| | CMD | DEV | 0.493(±0.009) | 0.523(±0.015) | 18.407(±0.472) | 25.595(±1.117) | 0.051(±0.001) | 0.052(±0.002) | 2529.365(±25.778) | 2437.075(±36.475) | 0.045(±0.002) | 0.068(±0.005) | 0.328(±0.017) | 0.354(±0.015) | 0.104(±0.008) | 0.113(±0.007) |
| | CMD | IWV | 0.249(±0.008) | 0.494(±0.007) | 8.101(±0.230) | 19.163(±0.806) | 0.033(±0.001) | 0.055(±0.001) | 287.728(±34.038) | 2154.704(±15.594) | 0.013(±0.000) | 0.036(±0.002) | 0.088(±0.005) | 0.260(±0.005) | 0.057(±0.001) | 0.126(±0.003) |
| | CMD | SB | 0.248(±0.004) | 0.485(±0.010) | 8.227(±0.279) | 19.200(±0.884) | 0.033(±0.001) | 0.054(±0.001) | 284.873(±15.767) | 2040.150(±163.039) | 0.014(±0.001) | 0.037(±0.003) | 0.089(±0.003) | 0.262(±0.009) | 0.058(±0.001) | 0.125(±0.003) |
| | CMD | TB | 0.252(±0.003) | 0.480(±0.010) | 8.392(±0.066) | 18.718(±1.336) | 0.034(±0.000) | 0.053(±0.001) | 299.252(±9.470) | 2055.212(±179.452) | 0.014(±0.000) | 0.037(±0.003) | 0.090(±0.003) | 0.251(±0.020) | 0.059(±0.001) | 0.124(±0.003) |
| | DARE-GRAM | DEV | 0.249(±0.003) | 0.393(±0.040) | 8.204(±0.122) | 14.737(±1.785) | 0.033(±0.001) | 0.044(±0.004) | 279.833(±11.947) | 1399.506(±654.664) | 0.013(±0.000) | 0.019(±0.005) | 0.087(±0.002) | 0.250(±0.020) | 0.058(±0.002) | 0.114(±0.006) |
| | DARE-GRAM | IWV | 0.249(±0.003) | 0.393(±0.040) | 8.204(±0.122) | 14.737(±1.785) | 0.033(±0.001) | 0.044(±0.004) | 279.833(±11.947) | 1399.506(±654.664) | 0.013(±0.000) | 0.019(±0.005) | 0.087(±0.002) | 0.250(±0.020) | 0.058(±0.002) | 0.114(±0.006) |
| | DARE-GRAM | SB | 0.249(±0.004) | 0.413(±0.071) | 8.136(±0.143) | 14.686(±2.786) | 0.033(±0.001) | 0.046(±0.007) | 283.028(±14.455) | 1729.784(±747.392) | 0.013(±0.000) | 0.022(±0.011) | 0.088(±0.002) | 0.244(±0.018) | 0.057(±0.002) | 0.116(±0.010) |
| | DARE-GRAM | TB | 0.250(±0.008) | 0.372(±0.029) | 8.312(±0.464) | 14.587(±1.570) | 0.034(±0.001) | 0.043(±0.002) | 290.095(±9.324) | 1144.219(±455.032) | 0.014(±0.001) | 0.021(±0.006) | 0.090(±0.007) | 0.242(±0.023) | 0.058(±0.004) | 0.111(±0.004) |
| | Deep Coral | DEV | 0.251(±0.005) | 0.382(±0.034) | 8.467(±0.283) | 15.271(±2.039) | 0.034(±0.001) | 0.045(±0.005) | 283.264(±4.893) | 967.525(±133.924) | 0.014(±0.001) | 0.019(±0.009) | 0.094(±0.005) | 0.255(±0.016) | 0.061(±0.004) | 0.111(±0.006) |
| | Deep Coral | IWV | 0.249(±0.006) | 0.380(±0.043) | 8.324(±0.294) | 14.801(±1.201) | 0.033(±0.001) | 0.044(±0.005) | 282.224(±9.132) | 1036.671(±195.506) | 0.014(±0.001) | 0.018(±0.005) | 0.093(±0.006) | 0.252(±0.008) | 0.059(±0.004) | 0.110(±0.006) |
| | Deep Coral | SB | 0.248(±0.006) | 0.356(±0.011) | 8.234(±0.295) | 14.031(±0.738) | 0.033(±0.001) | 0.041(±0.001) | 280.474(±6.420) | 1002.131(±167.663) | 0.014(±0.001) | 0.017(±0.003) | 0.091(±0.006) | 0.245(±0.014) | 0.058(±0.004) | 0.107(±0.001) |
| | Deep Coral | TB | 0.248(±0.006) | 0.356(±0.012) | 8.230(±0.287) | 14.066(±0.740) | 0.033(±0.001) | 0.041(±0.002) | 279.836(±6.636) | 1010.928(±179.067) | 0.014(±0.001) | 0.017(±0.003) | 0.090(±0.004) | 0.245(±0.014) | 0.058(±0.004) | 0.110(±0.007) |

Again, we further investigate model performance under distribution shift by examining predictions from the best performing model, selected by lowest average error in the target domain. Figure 20 presents the respective distribution of prediction errors in the source and target domain, clearly indicating the negative effects of the distribution shift on model performance.

In this task, the temperature field is the most critical for downstream analysis and optimization, which is why we focus our detailed analysis on it.

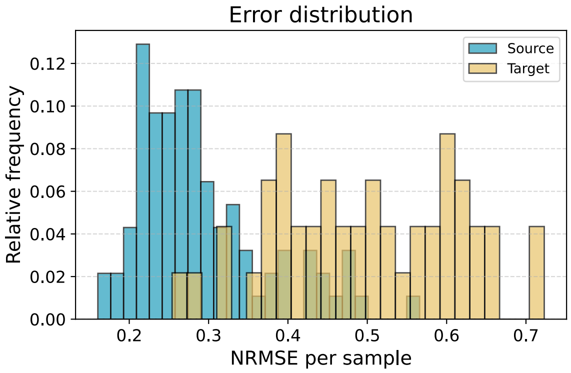


Figure 20: Distribution of NRMSE (averaged across all fields) for the test sets of the source (blue) and target (yellow) domains in the *heatsink design* dataset. Bar height indicates the relative frequency of samples within each bin.

Table 11 compares the absolute temperature prediction errors for the best and worst samples from both the source and target test sets. The corresponding scatter plots are shown in Figures 21 to 24, comparing the ground truth and predicted temperature fields, alongside their absolute errors.

While the best source domain prediction is quite accurate, with low average and percentile errors (Table 11, Figure 21), the 99th percentile of the worst source domain prediction reaches up to 29K. Given a total temperature range of 100K, this represents a relative error of nearly 30%. The worst target domain prediction is even less accurate, showing substantial visual and quantitative deviations from the ground truth (Table 11, Figure 24).

Table 11: Absolute error (K) of temperature predictions for the best and worst samples in the source and target domain of the *heatsink design* dataset. Lowest value per metric is bold.

| **Metric** | **Source** | | **Target** | |
|---|---|---|---|---|
| | **Best** | **Worst** | **Best** | **Worst** |
| Mean | **1.84e+00** | 5.79e+00 | 2.23e+00 | 1.42e+01 |
| Std | **1.94e+00** | 5.90e+00 | 2.85e+00 | 1.46e+01 |
| Median | **1.25e+00** | 4.06e+00 | 1.31e+00 | 8.84e+00 |
| $Q_{01}$ | **2.17e-02** | 7.51e-02 | 2.41e-02 | 1.62e-01 |
| $Q_{25}$ | **5.49e-01** | 1.92e+00 | 5.95e-01 | 4.49e+00 |
| $Q_{75}$ | **2.44e+00** | 7.48e+00 | 2.68e+00 | 1.87e+01 |
| $Q_{99}$ | **9.26e+00** | 2.88e+01 | 1.49e+01 | 6.61e+01 |

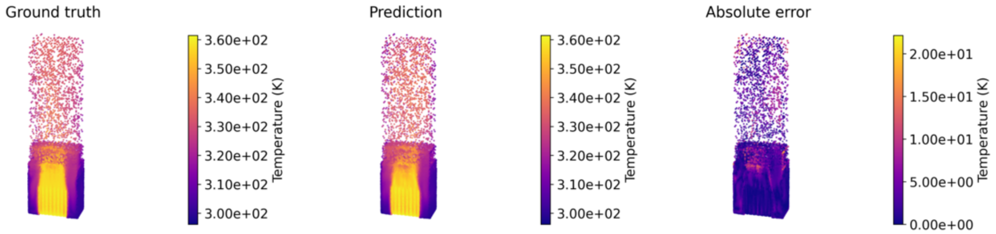


Figure 21: Sliced scatter plot of the *heatsink design* dataset (best source sample). Shown is the ground truth (left) and predicted (middle) temperature field, as well as the absolute error (right).

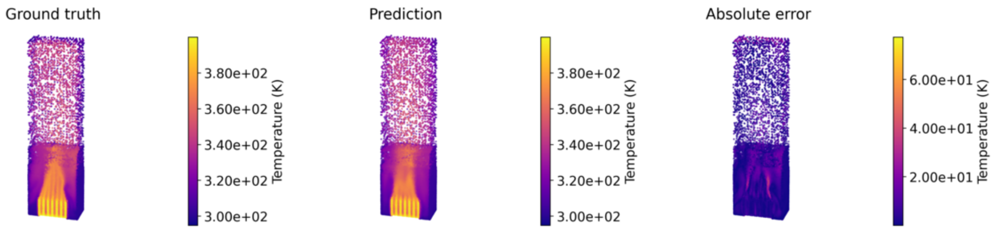


Figure 22: Sliced scatter plot of the *heatsink design* dataset (worst source sample). Shown is the ground truth (left) and predicted (middle) temperature field, as well as the absolute error (right).

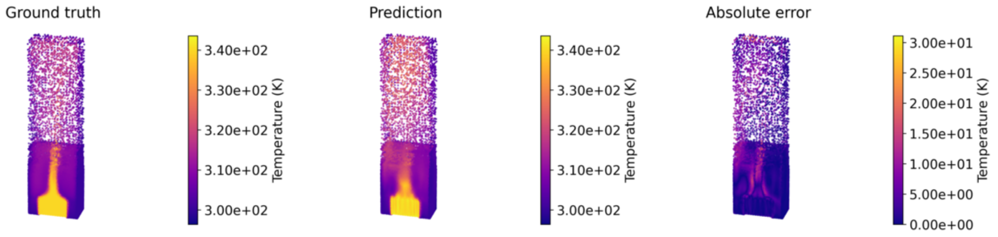


Figure 23: Sliced scatter plot of the *heatsink design* dataset (best target sample). Shown is the ground truth (left) and predicted (middle) temperature field, as well as the absolute error (right).

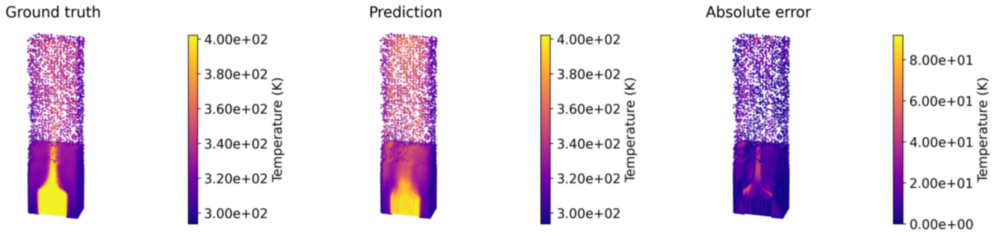


Figure 24: Sliced scatter plot of the *heatsink design* dataset (worst target sample). Shown is the ground truth (left) and predicted (middle) temperature field, as well as the absolute error (right).

## C DISTRIBUTION SHIFTS

To gain more insights into the parameter importance besides the domain experts' opinion, we visualize the latent space of the conditioning network for all presented datasets in Figures 25 to 28.

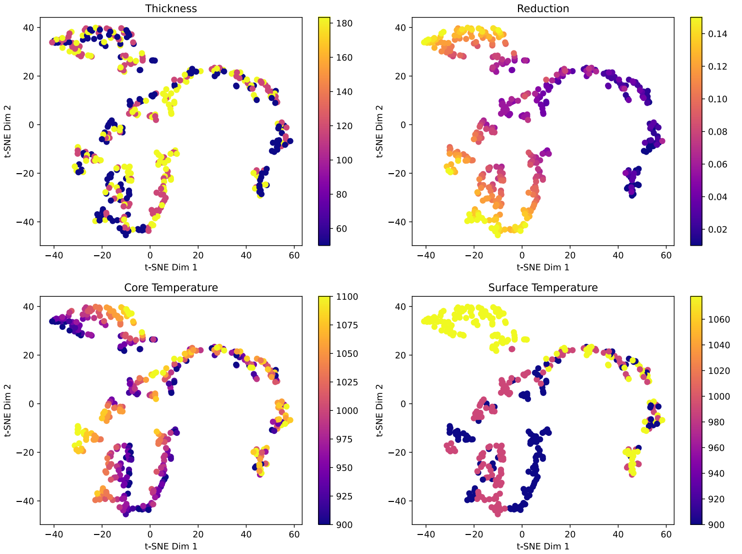


Figure 25: T-SNE visualization of the conditioning vectors for the *hot rolling* dataset. Point color indicates the magnitude of the respective parameter. While the slab thickness $t$ appears to be uniformly distributed, the remaining three exhibit distinct clustering patterns. Taking into account domain knowledge from industry experts, we defined the reduction parameter $r$ as the basis for constructing distribution shifts.

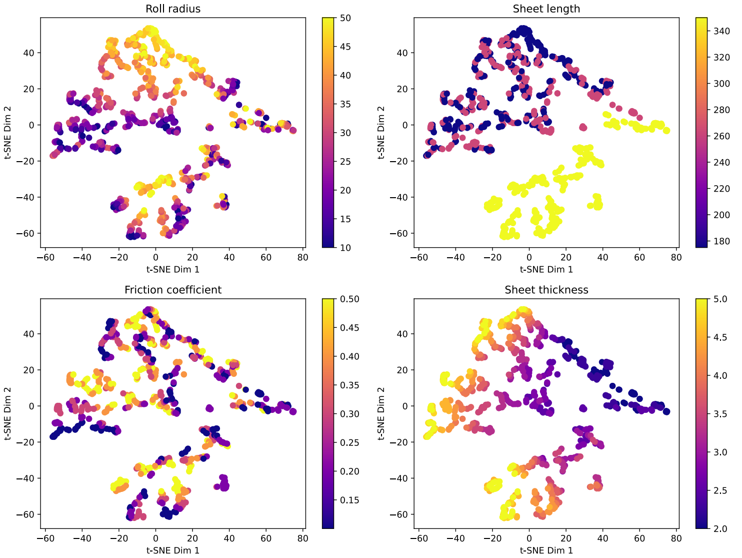


Figure 26: T-SNE visualization of the conditioning vectors for the *sheet metal forming* dataset. Point color indicates the magnitude of the respective parameter. The sheet length $l$ shows the most distinct groupings, but with only three discrete values, it is unsuitable for defining domain splits. The friction coefficient $\mu$ appears uniformly distributed across the embedding. In contrast, sheet thickness $t$ and roll radius $r$ show clustering behavior, making them more appropriate candidates for inducing distribution shifts. We choose $t$ as the domain defining parameter.

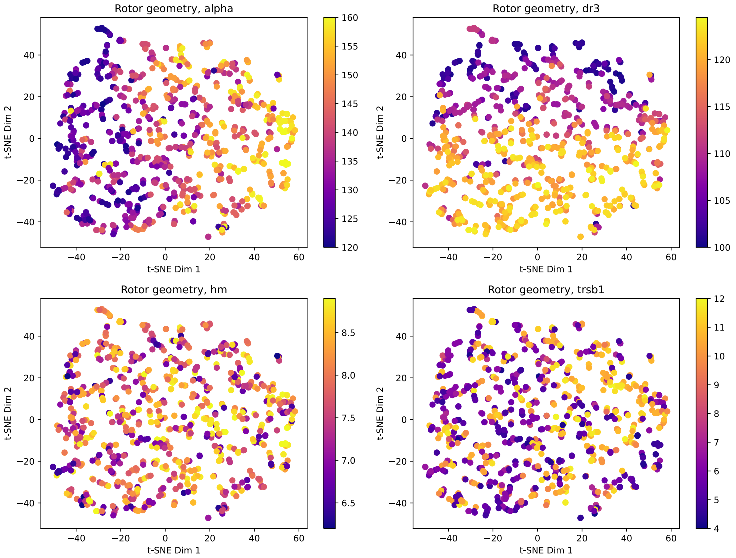


Figure 27: T-SNE visualization of the conditioning vectors for the *electric motor design* dataset. Point color indicates the magnitude of the respective parameter. For clarity, we only show selected parameters. The only parameter for which exhibits see some structure in the latent space is $d_{r3}$, we therefore choose this to be our domain defining parameter in accordance with domain experts.

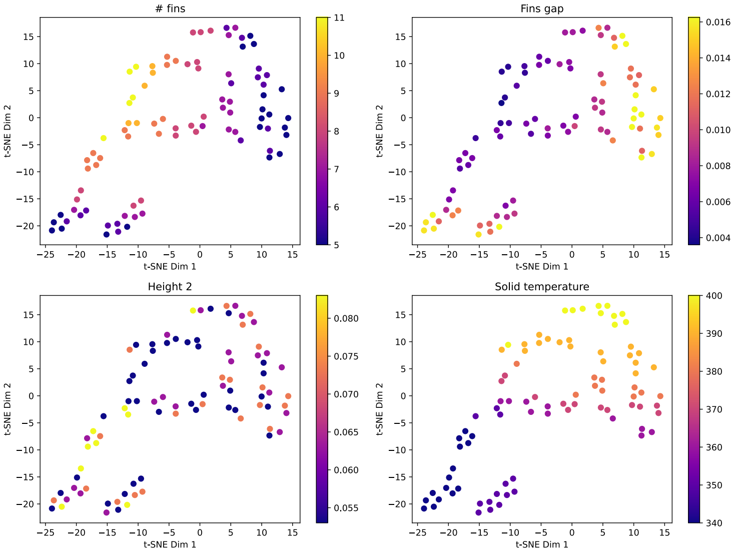


Figure 28: T-SNE visualization of the conditioning vectors for the *heatsink design* dataset. Point color indicates the magnitude of the respective parameter. Height 2 is distributed equally across the representation, but the other parameters show concrete grouping behavior. We therefore choose the number of fins as the domain defining parameter.

# D DISTANCE MEASURES

**Deep CORAL.** This distance measures the difference in second-order statistics (covariances) of source and target latent features and can be calculated as follows:

$$d_{\text{deep\_coral}}(\phi(\mathbf{x}), \phi(\mathbf{x}')) = \frac{1}{4k^2} \|\mathbf{C} - \mathbf{C}'\|_F^2 \,,$$

where $\phi(\mathbf{x}), \phi(\mathbf{x}') \in \mathbb{R}^{n \times k}$ denote latent source and target features for a batch size $n$ and a feature dimension $k$, $\mathbf{C}$ and $\mathbf{C}'$ are the source and target feature covariances and $\| \cdot \|_F^2$ is the squared Frobenius norm.

**CMD.** CMD measures not only the difference in first and second moments of source and target latent features, but also in higher-order central moments. Let $\phi(\mathbf{x}), \phi(\mathbf{x}') \in \mathbb{R}^{n \times k}$ denote the latent activations for a batch size $n$ and feature dimension $k$. The CMD distance up to order $P$ is defined as

$$d_{\text{cmd}}(\phi(\mathbf{x}), \phi(\mathbf{x}')) = \frac{1}{|b-a|} \|\boldsymbol{\mu} - \boldsymbol{\mu}'\|_2 + \sum_{p=2}^{P} \frac{1}{|b-a|^p} \|\mathbf{c}_p(\phi(\mathbf{x})) - \mathbf{c}_p(\phi(\mathbf{x}'))\|_2 \,,$$

where $\boldsymbol{\mu}, \boldsymbol{\mu}' \in \mathbb{R}^k$ are the source and target empirical mean feature vectors, $|b-a|^p$ can be seen as a hyperparameter of the method which we set to 2 to reflect the original implementation, and $\mathbf{c}_p(\phi(\mathbf{x})), \mathbf{c}'_p(\phi(\mathbf{x}')) \in \mathbb{R}^k$ are the respective $p$-th central moments which are calculated as:

$$\mathbf{c}_p(\phi(\mathbf{x})) = \frac{1}{n} \sum_{i=1}^{n} (\phi(\mathbf{x})_i - \boldsymbol{\mu})^{\odot p}, \qquad \mathbf{c}_p(\phi(\mathbf{x}')) = \frac{1}{n} \sum_{i=1}^{n} (\phi(\mathbf{x}')_i - \boldsymbol{\mu}')^{\odot p}.$$

Above, $(\cdot)^{\odot p}$ denotes the element-wise $p$-th power. Choosing the number of higher-order moments to align is another hyperparameter of the method. For our benchmark, we choose $P = 5$.

**DANN.** DANN is introduced to minimize an upper bound on the $\mathcal{H}$-divergence between source and target feature distributions. Since it is intractable to compute this directly, the authors use a domain classifier in the form of a small MLP trained to distinguish whether a latent feature comes from the source or the target domain. The error of this classifier is then used to compute the PAD, which, up to a constant depending on the model's VC dimension, upper-bounds the $\mathcal{H}$-divergence (Ganin et al., 2015).

Training is performed via a min–max optimization, i.e. the domain classifier is trained to maximize its classification accuracy, while the feature encoder $\phi$ is trained to *minimize* this separability by using a gradient reversal layer. This adversarial interaction encourages the latent representations of source and target samples to become indistinguishable, thereby promoting domain-invariant features.

**DARE-GRAM.** DARE-GRAM aims to align a selected low-rank subspace of the pseudo-inverse Gram matrices of source and target features. Given feature matrices $\phi(\mathbf{x}), \phi(\mathbf{x}') \in \mathbb{R}^{n \times k}$ for a batch size $n$ and feature dimension $k$, we can compute their Gram matrices:

$$G = \phi(\mathbf{x})^\top \phi(\mathbf{x}), \qquad G' = \phi(\mathbf{x})'^\top \phi(\mathbf{x})'.$$

Each Gram matrix is then decomposed via eigendecomposition, and its truncated Moore–Penrose pseudo-inverse is formed by keeping the top $p^*$ eigenvalues that explain a fixed proportion of variance (95% for our implementation):

$$G^+ = U_{1:p^*} \, \Lambda_{1:p^*}^{-1} \, U_{1:p^*}^\top \,, \qquad (G')^+ = U'_{1:p^*} \, (\Lambda'_{1:p^*})^{-1} \, (U'_{1:p^*})^\top \,.$$

We can then define the difference in angles as

$$d_{\text{angle}}(G, G') = \left\| \mathbf{1} - \cos(\theta_{1:p^*}) \right\|_1,$$

where $\cos(\theta_i)$ is the cosine similarity between the $i$-th column of $G^+$ and $(G')^+$.

Furthermore, we can define the difference in scale as

$$d_{\text{scale}}(G, G') = \left\| \lambda_{1:p^*} - \lambda'_{1:p^*} \right\|_2 .$$

The first term aligns the orientation of the dominant inverse-Gram subspaces, whereas the second term matches the principal eigenvalues of the Gram matrices to ensure that feature scale is consistent across source and target.

The total DARE-GRAM distance is defined as a weighted sum of the two:

$$d_{\text{dare\_gram}} = \alpha_{\text{angle}}\, d_{\text{angle}} + \gamma_{\text{scale}}\, d_{\text{scale}},$$

where the $\alpha$ and $\gamma$ are hyperparameters. Following the original authors, we set $\alpha_{\text{angle}} = 0.02$ and $\gamma_{\text{scale}} = 0.001$.

## E MODEL ARCHITECTURES

This section provides explanations of all model architectures used in our benchmark. All models are implemented in PyTorch and are adapted to our conditional regression task. All models have in common, that they take node coordinates as inputs and embed them using a sinusoidal positional encoding. Additionally, all models are conditioned on the input parameters of the respective simulation sample, which are encoded through a conditioning network described below.

**Conditioning Network.** The conditioning module used for all neural surrogate architectures embeds the simulation input parameters into a latent vector used for conditioning. The network consists of a sinusoidal encoding followed by a simple MLP. The dimension of the latent encoding is 8 throughout all experiments.

**PointNet.** Our PointNet implementation is adapted from (Qi et al., 2017) for node-level regression. Input node coordinates are first encoded using sinusoidal embeddings and passed through an encoder MLP. The resulting representations are aggregated globally using max pooling over nodes to obtain a global feature vector. To propagate this global feature, it is concatenated back to each point's feature vector. This fused representation is then fed into a final MLP, which produces the output fields. The conditioning is performed by concatenating the conditioning vector to the global feature before propagating it to the nodes features. We use a PointNet base dimension of 16 for the small model and 32 for the larger model.

**GraphSAGE.** We adapt GraphSAGE (Hamilton et al., 2017) to the conditional mesh regression setting. Again, input node coordinates are embedded using a sinusoidal encoding and passed through an MLP encoder. The main body of the model consists of multiple GraphSAGE message passing layers with mean aggregation. We support two conditioning modes, namely concatenating the latent conditioning vector to the node features, or applying FiLM style modulation (Perez et al., 2018) to the node features before each message passing layer. We always use FiLM modulation in the presented results. After message passing, the node representations are passed through a final MLP decoder to produce the output fields. The base dimension of the model is kept at 128 and we employ 4 GraphSAGE layers.

**Transolver.** The Transolver model follows the originally introduced architecture (Wu et al., 2024). Similar to the other models, node coordinates first are embedded using a sinusoidal encoding and passed through an MLP encoder to produce initial features. Through learned assignement, each node then gets mapped to a slice, and inter- as well as intra-slice attention is performed. Afterwards, fields are decoded using an MLP readout. The architecture supports two conditioning modes:

concatenation, where the conditioning vector is concatenated to the input node features before projection, or modulation through DiT layers across the network. For our experiments, DiT is used. We choose a latent dimension of 128, a slice base of 32 and we apply four attention blocks for the small model. For the larger model, we scale to 256, 128 and 8 layers respectively.

**UPT.** Our UPT implementation builds on the architecture proposed in (Alkin et al., 2024a). First, a fixed number of supernodes are uniformly sampled from the input nodes. Node coordinates are embedded using a sinusoidal encoding followed by an MLP. The supernodes aggregate features from nearby nodes using one-directional message passing and serve as tokens for subsequent transformer processing. They are then processed by stack of DiT blocks, which condition the network on the simulation input parameters. For prediction, we employ a DiT Perceiver (Jaegle et al., 2022) decoder that performs cross-attention between the latent representation and a set of query positions. This allows the model to generate field predictions at arbitrary spatial locations, which is a desirable property for inference. We sample 4096 supernodes and use a base dimension of 192. We use 8 DiT blocks for processing and 4 DiT Perceiver blocks for decoding.

**GINO.** GINO was proposed in (Li et al., 2023b). Input coordinates are again embedded via sinusoidal encoding, after which the mesh is projected onto a regular latent grid. This is achieved via message passing with connections generated via a radius graph. On the latent grid, the conditioning is concatenated to the features at each grid point before Fourier Neural Operator (FNO) Li et al. (2020a) layers are employed. Afterwards, features are mapped back onto the output grid by querying the latent grid, again via message passing. Our implementation uses a latent grid of size $(16 \times 16 \times 16)$ with 16 latent channels and a radius of 0.1 to construct the radius graph for messing passing operations. For our implementation, we use the library of the original authors.[3]

# F EXPERIMENTS

This section provides a detailed overview of the performed experiments for this benchmark. First, we explain the benchmarking setup used to generate the benchmarking results in detail in Appendix F.1 and the evaluation procedure in Appendix F.2. Furthermore, we provide information about training times for the presented methods in Appendix F.3.

## F.1 EXPERIMENTAL SETUP

**Dataset Splits.** We split each dataset into source and target domains as outlined in Section 3.5 and Appendix C. Within source domains, we use a 50%/25%/25% split for training, validation, and testing, respectively. For target domains, where labels are unavailable during training in our UDA setup, we use a 50%/50% split for training and test sets. The large validation and test sets are motivated the industrial relevance of our benchmark, where reliable performance estimation on unseen data is a crucial factor.

**Training Pipeline.** For training, we use a dataset wide per field z-score normalization strategy, with statistics computed on the source domain training set. We use a batch size of 16 and the AdamW optimizer (Loshchilov & Hutter, 2019) with a weight decay of 1e-5 and a cosine learning rate schedule, starting from 1e-3. Gradients are clipped to a maximum norm of 1. For the large scale *heatsink design* dataset, we enable Automatic Mixed Precision (AMP) to reduce memory consumption and training time. Additionally, we use Exponential Moving Average (EMA) updates with a decay factor of 0.95 to stabilize training.

Performance metrics are evaluated every 10 epochs, and we train all models for a maximum of 3000 epochs with early stopping after 500 epochs of no improvement on the source domain validation loss.

**Domain Adaptation Specifics.** To enable UDA algorithms, we jointly sample mini batches from the source and target domains at each training step and pass them thorugh the model. Since target labels are not available, we compute supervised losses only on the source domain outputs. In addition,

---

[3] https://github.com/neuraloperator/neuraloperator

we compute DA losses on the latent representations of source and target domains in order to encourage domain invariance.

Since a crucial factor in the performance of UDA algorithms is the choice of the domain adaptation loss weight $\lambda$, we perform extensive sweeps over this hyperparameter and select models using the unsupervised model selection strategies described in Section 4.3.

For the three smaller datasets, we sweep $\lambda$ logarithmically over $\lambda \in \{10^{-1}, 10^{-2}, \dots, 10^{-9}\}$, while for the large scale ***Heatsink design*** dataset, we sweep a smaller range, namely $\lambda \in \{10^{2}, 10^{-1}, \dots, 10^{-2}\}$, motivated by the balancing principle (Zellinger et al., 2021b).

Table 12 provides an overview of the number of trained models for benchmarking performance of all models and all UDA algorithms on the *medium* difficulty domain shifts across all datasets.

Table 12: Overview of the benchmarking setup and number of trained models across all datasets.

| **Dataset** | **Models** | **UDA algorithms** | **$\lambda$ values** | **# seeds** | **# models trained** |
|---|---|---|---|---|---|
| Rolling | PointNet, GraphSAGE, Transolver | Deep Coral, CMD, DANN, DARE-GRAM | $\{10^{-1}; 10^{-9}\}$ | 4 | 432 |
| | | w/o UDA | – | 4 | 12 |
| Forming | PointNet, GraphSAGE, Transolver | Deep Coral, CMD, DANN, DARE-GRAM | $\{10^{-1}; 10^{-9}\}$ | 4 | 432 |
| | | w/o UDA | – | 4 | 12 |
| Motor | PointNet, GraphSAGE, Transolver | Deep Coral, CMD, DANN, DARE-GRAM | $\{10^{-1}; 10^{-9}\}$ | 4 | 432 |
| | | w/o UDA | – | 4 | 12 |
| Heatsink | PointNet, Transover, UPT, GINO | Deep Coral, CMD, DANN, DARE-GRAM | $\{10^{2}; 10^{-2}\}$ | 4 | 320 |
| | | w/o UDA | – | 4 | 12 |
| **Sum** | | | | | **1,664** |

**Additional Details.** For the three smaller datasets, we use smaller networks, while for the large scale *heatsink design* dataset, we train larger model configurations to accommodate the increased data complexity. An overview of model sizes along with average training times per dataset is provided in Table 13. We also refer to the accompanying code repository for a complete listing of all model hyperparameters, where we provide all baseline configuration files and detailed step by step instructions for reproducibility of our results.

Another important detail is that, during training on the *heatsink design* dataset, we randomly subsample 16,000 nodes from the mesh in each training step to ensure computational tractability. However, all reported performance metrics are computed on the full resolution of the data without any subsampling.

## F.2 EVALUATION METRICS

### F.2.1 GENERAL METRICS

We report the RMSE for each predicted output field. For field $i$, the RMSE is defined as:

$$\text{RMSE}_i^{\text{field}} = \frac{1}{M} \sum_{m=1}^{M} \sqrt{\frac{1}{N_m} \sum_{n=1}^{N_m} \left( y_{m,n}^{(i)} - f(x)_{m,n}^{(i)} \right)^2},$$

where $M$ is the number of test samples (graphs), $N_m$ the number of nodes in graph $m$, $y_{m,n}^{(i)}$ the ground truth value of field $i$ at node $n$ of graph $m$, and $f(x)_{m,n}^{(i)}$ the respective model prediction.

For aggregated evaluation, we define the total Normalized RMSE (NRMSE) as:

$$\text{NRMSE} = \frac{1}{K} \sum_{i=1}^{K} \text{NRMSE}_i^{\text{field}},$$

where $K$ is the number of predicted fields. For this metric, all individual field errors are computed on normalized fields before aggregation.

In addition to the error on the fields, we report the mean Euclidean error of the predicted node displacement. This is computed based on the predicted coordinates $\hat{\mathbf{c}}_{m,n} \in \mathbb{R}^d$ and the ground truth coordinates $\mathbf{c}_{m,n} \in \mathbb{R}^d$, where $d \in \{2, 3\}$ is the spatial dimensionality, as follows:

$$\text{RMSE}^{\text{deformation}} = \frac{1}{M}\sum_{m=1}^{M}\sqrt{\frac{1}{N_m}\sum_{n=1}^{N_m}\|\mathbf{c}_{m,n} - \hat{\mathbf{c}}_{m,n}\|_2}.$$

### F.2.2 PHYSICS METRICS

**Von Mises stress consistency.** For all structural simulations in our benchmark, we predict both the relevant Cauchy stress tensor components and the von Mises equivalent stress. This allows for an internal consistency check using the standard von Mises definition:

$$\sigma_{vM} = \sqrt{\frac{1}{2}\left[(\sigma_{11}-\sigma_{22})^2 + (\sigma_{22}-\sigma_{33})^2 + (\sigma_{33}-\sigma_{11})^2 + 6\tau_{12}^2\right]},$$

with $\sigma_{11}, \sigma_{22}, \sigma_{33}$ denote the normal stresses and $\tau_{12}$ the in-plane shear stress.

We can recompute $\sigma_{vM}$ from the predicted tensor components and compare it to the predicted von Mises value using a normalized mean absolute error:

$$\text{Consistency}_{\text{vM}} = \frac{\sum_{i=1}^{N}|\sigma_{vM,i} - \sigma_{vM_{recalc},i}|}{\sum_{i=1}^{N}|\sigma_{vM,i}|}$$

**Constitutive law consistency.** For the *sheet metal forming* dataset, the material is modeled as elastoplastic with von Mises plasticity and linear isotropic hardening. This defines a yield surface, $\sigma_y$, which represents the material's current strength as a function of the equivalent plastic strain ($\varepsilon_p$):

$$\sigma_y(\varepsilon_p) = \sigma_{y0} + H\,\varepsilon_p,$$

where $\sigma_{y0}$ is the initial yield stress, H the hardening modulus, and $\varepsilon_p$ the equivalent plastic strain.

A physically-correct model must adhere to two conditions based on this law:

1. Elastic nodes ($\varepsilon_p = 0$) must have a stress below this surface: $\sigma_{vM} \leq \sigma_{y0}$.
2. Plastic nodes ($\varepsilon_p > 0$) must have a stress on this surface: $\sigma_{vM} = \sigma_y(\varepsilon_p)$.

Based on these two conditions, we introduce two metrics to evaluate the physical consistency of the predictions:

1. Elastic violation rate (percentage of elastic nodes that incorrectly violate the initial yield stress):

$$\text{Violation}_{\text{elastic}} = \frac{1}{N_{\text{el}}}\sum_{i\in\mathcal{E}}\mathbf{1}[\,\sigma_{vM,i} > \sigma_{y0}\,],$$

   where $\mathcal{E}$ is the set points in the elastic regime and $N_{\text{el}} = |\mathcal{E}|$ is the number of elastic nodes.

2. Plastic Law Residual (NMAE for all plastic nodes):

$$\text{Residual}_{\text{plastic}} = \frac{1}{N_{\text{pl}}}\sum_{i\in\mathcal{P}}\frac{|\sigma_{vM,i} - (\sigma_{y0} + H\,\varepsilon_{p,i})|}{\sigma_{y0}},$$

   where $\mathcal{P}$ is the set points in the plastic regime and $N_{\text{pl}} = |\mathcal{P}|$ is the number of plastic nodes.

**Boundary condition satisfaction.** The *heatsink design* simulations impose two important Dirichlet Boundary Conditions (BCs) on the fin surfaces: no slip velocity and the solid temperature of the fins. Therefore we define the two following errors to measure the violation of these BCs for our surrogates:

$$\text{BC-violation}_T = \frac{1}{N_{\text{fin}}}\sum_{i\in\mathcal{F}}\frac{|T_i - T_{\text{solid}}|}{|T_{\text{solid}} - T_{\text{env}}|},$$

and

$$\text{BC-violation}_{\mathbf{u}} = \frac{1}{N_{\text{fin}}} \sum_{i \in \mathcal{F}} \|\mathbf{u}_i\| \,,$$

where $\mathcal{F}$ is the set of fin nodes, $N_{\text{fin}} = |\mathcal{F}|$ is the number of fin nodes, and $T_i$ and $\mathbf{u}_i$ are the respective predictions for temperature and velocity at node $i$.

These should be interpreted as “soft” BC consistency checks. OpenFOAM enforces Dirichlet BCs on faces of boundary patches, whereas our dataset contains cell center values. Cells adjacent to the fins generally exhibit nonzero gradients, meaning their ground-truth temperatures and velocities do not exactly satisfy the BCs, also in the ground truth data.

### F.3 COMPUTATIONAL RESOURCES AND TIMINGS

While generating the results reported on the *medium* difficulty level of our benchmark, we measured average training times per dataset and model architecture. All our runs were timed on a single NVIDIA H200 144GB GPU for a fair comparison. While the total compute budget is difficult to estimate due early stopping, we provide a detailed analysis of the average training times for 2000 epochs in Table 13.

This table refers to models trained with Deep CORAL, however different UDA algorithms do not add significant computational cost. What is more impactful concerning the full pipeline (including model selection) is the number of hyperparameter variations. The total cost of one UDA algorithm & model selection pipeline can be estimated by multiplying the average training time by the number of trained models (e.g. $\times 9$ if one sweeps over 9 hyperparameters of $\lambda$), for sequential execution. Furthermore, the model selection method’s runtime training is negligible compared to the training times.

Table 13: Average training times (averaged for 2000 epochs) and parameter counts for each model on the *medium* difficulty benchmark tasks. Times are measured on a H200 144GB GPU using a batch size of 16.

| **Dataset** | **# samples** | **Avg. # nodes** | **Model** | **# parameters** | **Avg. training time (h)** |
|---|---|---|---|---|---|
| Rolling | 4,750 | 576 | PointNet | 0.3M | 0.75 |
| | | | GraphSAGE | 0.2M | 1.77 |
| | | | Transolver | 0.57M | 1.77 |
| Forming | 3,315 | 6,417 | PointNet | 0.3M | 2.35 |
| | | | GraphSAGE | 0.2M | 10.82 |
| | | | Transolver | 0.57M | 3.74 |
| Motor | 3,196 | 9,052 | PointNet | 0.3M | 2.35 |
| | | | GraphSAGE | 0.2M | 12.14 |
| | | | Transolver | 0.57M | 3.60 |
| Heatsink | 460 | 1,385,594 | PointNet | 1.08M | 3.88 |
| | | | Transolver | 4.07M | 4.94 |
| | | | UPT | 5.77M | 4.73 |
| | | | GINO | 2.5M | 5.94 |

## G DATASET DETAILS

### G.1 HOT ROLLING

The *hot rolling* dataset represents a hot rolling process in which a metal slab undergoes plastic deformation to form a sheet metal product. The model considers a plane-strain representation of a heated steel slab segment with a core temperature $T_{\text{core}}$ and a surface temperature $T_{\text{Surf}}$, initially at thickness $t$, passing through a simplified roll stand with a nominal roll gap $g$ (see Figure 2a). This roll gap effectively matches the exit thickness of the workpiece. Given the material properties, the initial temperature distribution over the slab thickness and the specified pass reduction, the model aims to capture the evolution of the thermo-mechanical state of the workpiece as it traverses the roll gap.

To reduce computational complexity, the analysis is confined to the vertical midplane along the rolling direction based on a plane-strain assumption. This is well justified by the high width-to-thickness ratio characteristic of the workpiece. Additionally, vertical symmetry is also exploited. Consequently, only the upper half of the workpiece and the upper work roll are modeled.

The workpiece is discretized using plane-strain, reduced-integration, quadrilateral elements. Mesh generation is fully automated, with the element size calibrated according to findings from a mesh convergence study. In terms of mechanical behavior, the workpiece is modeled as elasto-plastic with isotropic hardening, employing tabulated flow curves representative for a titanium alloy (Lesuer, 2000; Lu et al., 2018). The elastic modulus and flow stress are temperature dependent, with the latter also influenced by the plastic strain rate. In contrast, material density and Poisson's ratio are assumed to remain constant. The work roll with a diameter of 1000 mm is idealized as an analytically defined rigid body.

In addition to the mechanical behavior, the elements also feature a temperature degree of freedom that captures thermal phenomena, which are in turn fully coupled with the mechanical field. Heat conduction within the workpiece is governed by temperature dependent thermal conductivity and specific heat capacity. Heat transfer at the interface between the workpiece and the roll is modeled as proportional to the temperature difference between the contacting surfaces, using a heat transfer coefficient of 5 $\mathrm{mW/mm^2K}$. The model also accounts for internal heat generation due to plastic deformation, based on the standard assumption that 90% of plastic work is converted into heat. Additionally, all frictional energy is assumed to be fully transformed into heat and evenly divided between the workpiece and the roll. However, since the analysis focuses on the workpiece, only the portion of this heat entering the workpiece is considered.

The FE simulation is performed with the *Abaqus* explicit solver using a relatively high mass scaling factor of 100. This mass scaling proved to be a suitable choice for maintaining both computational efficiency and solution accuracy. The pre-processing, evaluation and post-processing of the simulations was automated in Python. A full factorial design of experiments was conducted by varying the parameters outlined in Table 14. Simulation outputs from Abaqus (.odb files) were converted to a more suitable .h5 format in post-processing, enabling seamless integration into the SIMSHIFT framework. All simulations were run on a Gigabyte Aorus 15P KD consumer laptop equipped with an Intel Core i7-11800H CPU (8 cores, 16 threads, 2.30–4.60 GHz), 16 GB DDR4 RAM at 3200 MHz and a 1 TB NVMe SSD. The single-core CPU time for one simulation was 25 seconds on average, depending on the mesh size and convergence speed.

Table 14: Input parameter ranges for the *hot rolling* simulations. Samples are generated by equally spacing each parameter within the specified range using the indicated number of steps, resulting in $5 \times 19 \times 10 \times 5 = 4750$ total samples.

| Parameter | Description | Min | Max | Steps |
|---|---|---|---|---|
| $t$ $(mm)$ | Initial slab thickness. | 50.0 | 183.3 | 5 |
| reduction $(-)$ | Reduction of initial slab thickness. | 0.01 | 0.15 | 19 |
| $T_{\text{core}}$ $(^{\circ}C)$ | Core slab temperature. | 900.0 | 1000.0 | 10 |
| $T_{\text{surf}}$ $(^{\circ}C)$ | Surface slab temperature. | 900.0 | 1077.77 | 5 |

## G.2 SHEET METAL FORMING

For the *sheet metal forming* dataset, a w-shaped bending process was selected due to its complex contact interactions and the highly nonlinear progression of bending forces. For this purpose, a parameterized 2D FE model of the process was developed using the commercial FEM software *Abaqus* and its implicit solver, with the simulation pipeline implemented in Python. The initial configuration of the finite element model is shown in Figure 29 and described below.

Due to geometric and loading symmetry, only the right half of the sheet with a thickness $t$ was modeled. The die and punch were idealized as rigid circular segments with a shared radius $r$. Additionally, a

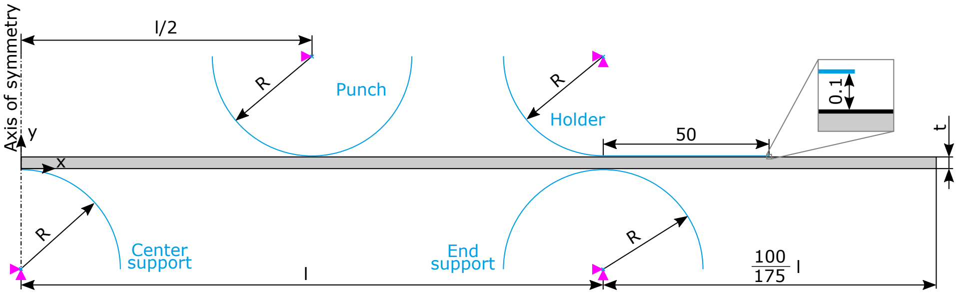


Figure 29: Bending process abstraction, initial configuration.

rigid blank holder comprising an arc and a straight segment was positioned $0.1$ mm above the sheet to maintain contact and restrain vertical motion. The required sheet length was determined by the support span $l$, enabling material flow toward the center in response to the downward motion of the punch.

The sheet was discretized using bilinear, plane-strain quadrilateral elements with reduced integration and hourglass control (Abaqus element type CPE4R). A prior mesh convergence study indicated that accurate simulation results require a minimum of 10 element rows across the sheet thickness. The element size was fixed at $0.125 \times 0.1$ mm to ensure a uniform aspect ratio, constraining the sheet thickness to $t > 1$ mm.

The sheet material was modeled as elastoplastic with von Mises plasticity and linear isotropic hardening. The following properties were assigned: Young's modulus of 210 GPa, Poisson's ratio of 0.3, yield stress of 410 MPa, and hardening modulus of 2268 MPa.

For all contact interfaces, a normal contact formulation with surface-to-surface discretization, penalty enforcement, and finite-sliding tracking was employed. Tangential contact was modeled via a Coulomb friction law with a coefficient $\mu$.

The supports and blank holder were fixed by constraining horizontal and vertical translations as well as in-plane rotations. These constraints were applied at the centroid of each arc segment, representing the reference point for the respective rigid body. The punch was similarly constrained against horizontal movement and rotation but retained vertical mobility. The deformed configuration following a vertical displacement $U$ of the punch is illustrated in Figure 30.

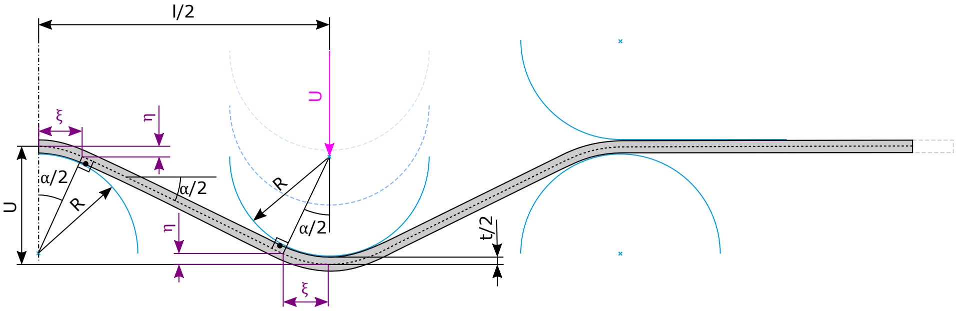


Figure 30: Bending process abstraction, deformed configuration.

A full factorial design of experiments was conducted by varying the parameters outlined in Table 15. As for the hot rolling simulations, outputs from Abaqus (.odb files) were converted to .h5 format in post-processing, to integrate them into the SIMSHIFT framework. All simulations were run on a Gigabyte Aorus 15P KD consumer laptop equipped with an Intel Core i7-11800H CPU (8 cores, 16 threads, 2.30–4.60 GHz), 16 GB DDR4 RAM at 3200 MHz and a 1 TB NVMe SSD. The single-core CPU time for one simulation run was 300 seconds on average, depending on mesh size and convergence speed.

Table 15: Input parameter ranges for the *sheet metal forming* simulations. Samples are generated by equally spacing each parameter within the specified range using the indicated number of steps, resulting in $17 \times 13 \times 3 \times 5 = 3315$ total samples.

| Parameter | Description | Min | Max | Steps |
|---|---|---|---|---|
| $r\ (mm)$ | Roll radius. | 10.0 | 50.0 | 17 |
| $t\ (mm)$ | Sheet thickness. | 2.0 | 5.0 | 13 |
| $l\ (mm)$ | Sheet length. | 175.0 | 350.0 | 3 |
| $\mu\ (-)$ | Friction coefficient between the sheet and the rolls. | 0.1 | 0.5 | 5 |

### G.3 ELECTRIC MOTOR DESIGN

The *electric motor design* dataset includes a structural FE simulation of a rotor within electric machinery, subjected to mechanical loading at burst speed. The rotor topology is modeled after the motor architecture of the 2010 Toyota Prius (Burress et al., 2011), an industry-recognized benchmark frequently used for validation and comparison in academic and industrial research. The Prius rotor topology is based on a V-shaped magnet configuration as shown in Figure 31.

Structural rotor simulations are essential in multi-physics design optimization, where motor performance is evaluated across multiple domains including electromagnetic, thermal, acoustic, and structural. Using a design optimization framework, stator and rotor design are iteratively refined to identify Pareto-optimal solutions based on objectives such as efficiency, torque, weight, and speed. In this process, the structural FE model predicts stress and deformation due to loading ensuring the rotor's structural integrity.

The set up and execution of the structural simulations for this dataset are automated and implemented in the open source design optimization framework *SyMSpace*[4]. The FE simulation of the rotor is performed using a mixed 2D plane stress and plane strain formulation with triangular elements. To enhance computational efficiency, geometric symmetry is exploited and only a 1/16 sector of the full rotor is modeled. The mechanical simulation is static and evaluates the rotor under centrifugal loading, incorporating press-fit conditions between the rotor core and shaft, as well as contact interactions between the rotor core and embedded magnets.

An elastic material behavior is employed for all components, including the rotor core, shaft, and magnets. Material properties are summarized in Table 16. Based on the parametrized CAD model of the rotor topology, the geometry is automatically meshed using *Netgen*[5]. The design optimization tool also automatically identifies nodes for boundary conditions and contact surfaces and applies the corresponding constraints and interactions required for the simulation. The implicit FE solver *HOTINT* is used to compute the quasi-static response of the system, providing local stress and strain fields across the rotor topology.

Table 16: Material parameters for the structural *electric motor design* simulations.

|  | Rotor Core | Rotor Shaft | Permanent Magnet |
|---|---|---|---|
| Material | NO27-14 Y420HP | 42CrMo4 | BMN-40SH |
| Density ($\mathrm{kg/dm^3}$) | 7.6 | 7.72 | 7.55 |
| Possions ratio (-) | 0.29 | 0.3 | 0.24 |
| Young's Modulus ($\mathrm{kN/mm^2}$) | 185.0 | 210.0 | 175.0 |
| Tensile Strength ($\mathrm{kN/mm^2}$) | 550.0 | 850.0 | 250.0 |

To generate the electric motor dataset, a comprehensive motor optimization study was conducted using *SyMSpace*, based on design specifications of the 2010 Toyota Prius. The optimization aimed to minimize multiple performance metrics, including motor mass, material costs, rotor torque ripple, motor losses, coil temperature, stator terminal current, and elastic rotor deformation. A genetic algorithm was employed to explore the design space and identify Pareto-optimal solutions. In the

---

[4] https://symspace.lcm.at/

[5] https://ngsolve.org/

process, 3,196 motor configurations were evaluated by varying, among other factors, the rotor's topological parameters within the bounds specified in Table 17. The outputs of the structural simulations were generated in .vtk format and then stored in .h5 files, allowing direct integration into the SIMSHIFT framework. Each structural simulation required approximately 4 to 5 minutes of single-core CPU time on a Intel Core i9-14900KS processor (24 Cores, 3200 MHz), depending on convergence speed of the contact algorithm.

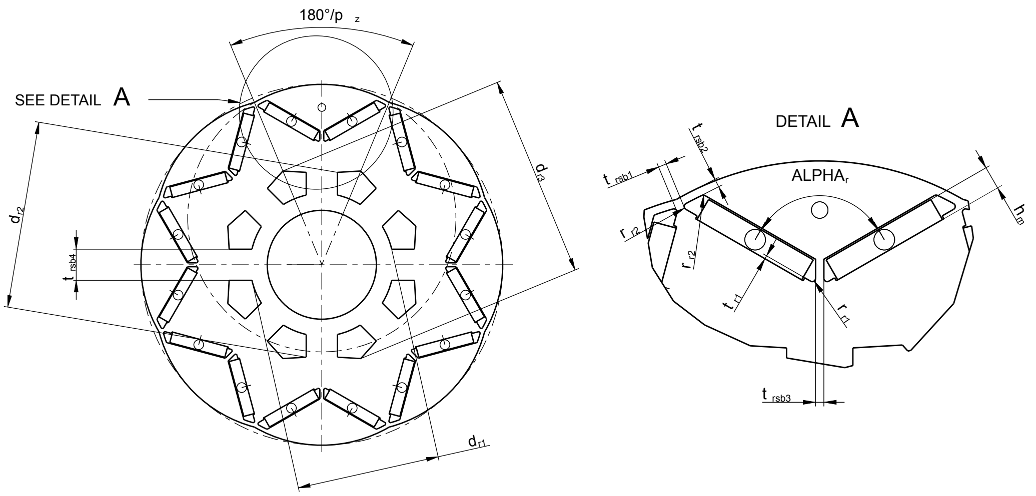


Figure 31: Technical drawing of the electrical motor. Sampling ranges for the shown parameters can be found in Table 17.

Table 17: Input parameters for the *electric motor design* simulations. Since the design space was explored by a genetic algorithm, the parameters are not uniformly sampled as in the previous simulation scenarios. In total, 3196 simulations were performed.

| Parameter | Description | Min | Max |
|---|---|---|---|
| $d_{si}$ $(mm)$ | Stator inner diameter. | 150.0 | 180.0 |
| $h_m$ $(mm)$ | Magnet height. | 6.0 | 9.0 |
| $\alpha_r$ $(°)$ | Angle between magnets. | 120.0 | 160.0 |
| $t_{r1}$ $(mm)$ | Magnet step. | 1.0 | 5.0 |
| $r_{r1}$ $(mm)$ | Rotor slot fillet radius 1. | 0.5 | 2.5 |
| $r_{r2}$ $(mm)$ | Rotor slot fillet radius 2. | 0.5 | 3.5 |
| $r_{r3}$ $(mm)$ | Rotor slot fillet radius 3. | 0.5 | 5.0 |
| $r_{r4}$ $(mm)$ | Rotor slot fillet radius 4. | 0.5 | 3.0 |
| $t_{rsb1}$ $(mm)$ | Thickness saturation bar 1. | 4.0 | 12.0 |
| $t_{rsb2}$ $(mm)$ | Thickness saturation bar 2. | 1.0 | 3.0 |
| $t_{rsb3}$ $(mm)$ | Thickness saturation bar 3. | 1.2 | 4.0 |
| $t_{rsb4}$ $(mm)$ | Thickness saturation bar 4. | 5.0 | 12.0 |
| $d_{r1}$ $(mm)$ | Rotor slot diameter 1. | 60.0 | 80.0 |
| $d_{r2}$ $(mm)$ | Rotor slot diameter 2. | 80.0 | 120.0 |
| $d_{r3}$ $(mm)$ | Rotor slot diameter 3. | 100.0 | 125.0 |

### G.4 HEATSINK DESIGN

The *heatsink design* dataset consists of heatsink geometries similar to the example shown in Figure 32, placed centrally at the bottom of a surrounding box-shaped domain filled with air. The dimensions of the surrounding enclosure are 0.14 m × 0.14 m × 0.5 m (length × width × height).

The geometric configuration of each heatsink is defined by several parameters, which were varied within specified bounds for the design study. These parameters and their corresponding value ranges are summarized in Table 18. A total of 460 simulation cases were generated, with non-uniform sampling across the parameter space.

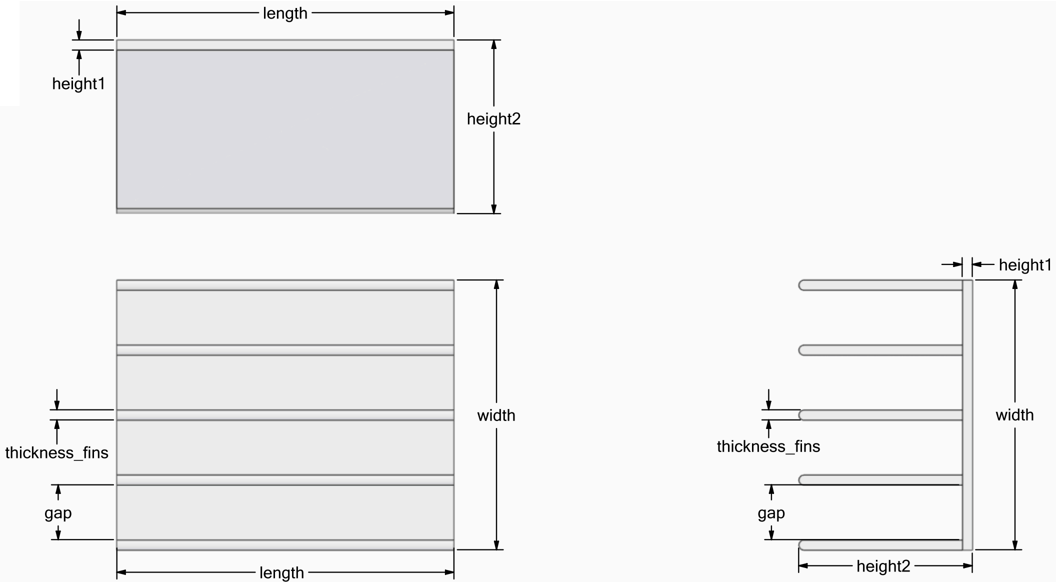


Figure 32: Technical drawing of the solid body in the *heatsink design* dataset. Some of the shown parameters are varied for data generation (see Table 18).

Table 18: Geometric and physical parameters of the *heatsink design* simulations. The variable parameters were not uniformly sampled. In total, $460$ simulations were performed.

| Parameter | Description | fixed Value | Min | Max |
|---|---|---|---|---|
| length $(\mathrm{m})$ | Heatsink length | 0.1 | - | - |
| width $(\mathrm{m})$ | Heatsink width | 0.08 | - | - |
| height1 $(\mathrm{m})$ | Baseplate height | 0.003 | - | - |
| T(amb) $(\mathrm{K})$ | Ambient Temperature | 300 | - | - |
| fins $(-)$ | Number of fins | - | 5 | 14 |
| gap $(\mathrm{m})$ | Gap between fins | - | 0.0023 | 0.01625 |
| thickness_fins $(\mathrm{m})$ | Thickness of fins | - | 0.003 | 0.004 |
| height2 $(\mathrm{m})$ | Heatsink height | - | 0.053 | 0.083 |
| T (solid) $(\mathrm{K})$ | Temperature of the solid fins | - | 340 | 400 |

The dataset was generated using CFD simulations based on the Reynolds-Averaged Navier-Stokes (RANS) equations coupled with the energy equation. All simulations were conducted in the open-source CFD suite *OpenFOAM 9*.

The computational domain was discretized using a finite volume method with second-order spatial discretization schemes. A structured hexahedral background mesh was generated with the blockMesh utility in OpenFOAM, followed by mesh refinement using snappyHexMesh to accurately resolve the heatsink structure defined in STL format.

To simulate buoyancy driven natural convection, the buoyantSimpleFoam solver was employed. This solver is designed for steady state, compressible, buoyant flows, using the SIMPLE algorithm for pressure-momentum coupling, extended with under relaxation techniques to enhance numerical stability and robust convergence.

Boundary conditions were applied as follows:

- Walls of the surrounding: no-slip velocity condition with fixed ambient temperature as defined in Table 18.
- Walls of the heatsink: no-slip velocity condition with solid temperature within the range specified for parameter T (solid) in Table 18.

Given the turbulent nature of the flow, the RANS equations were closed using the SST k–$\omega$ turbulence model (Menter et al., 2003). Near-wall regions were modeled using a $y^+$-insensitive near-wall treatment, allowing accurate resolution of boundary layers without the need for excessively fine meshes.

A mesh convergence study was conducted to ensure numerical accuracy. Depending on mesh resolution, each simulation required approximately 11 to 18 hours of single-core CPU time on an Intel Core i9-14900KS processor (24 cores, 3.2 GHz).

## H ABLATION STUDIES

In the following sections, we present ablations on the SIMSHIFT framework.

### H.1 GEOMETRIC ENCODING

The design concept of SIMSHIFT is to allow plug-in integration of any UDA algorithm and model architecture, as long as the model can be conditioned in some way (see Figure 1). However, explicitly conditioning models on scalar geometric parameters is not the only option: for instance, domain-specific information may be encoded implicitly in the mesh itself. To investigate this, we provide an ablation in which the model encodes the mesh directly and is not explicitly conditioned on the scalar parameters. Specifically, we replace the feed-forward conditioning network with a geometric PointNet based encoder to embed the input mesh into a global latent vector, on which UDA is then performed.

We report results of this setup on the *electric motor design* dataset. The setup follows the benchmarking procedure described in Section 4 and Appendix F.1: for each UDA algorithm, we train across 9 different regularizer strengths and 4 random seeds.

Table 19: RMSE (mean ± std over 4 seeds) on the *electric motor design* dataset when using a PointNet geometry encoder. Values are target domain errors (lower is better). Bold marks the overall best model + UDA algorithm + model selection combination. For each architecture, the unregularized baseline row is shaded beige, whereas the best UDA + selection within that architecture is underlined and shaded green.

| Model | DA Algorithm | Model Selection | All Fields Normalized Avg (-) | | Rel Custom Error (-) | | Deformation (m) | | Logarithmic Strain ($\times10^{-2}$) | | Principal Strain ($\times10^{-2}$) | | Stress (MPa) | | Cauchy Stress (MPa) | | Mises Stress (MPa) | | Principal Stress (MPa) | | Total Strain ($\times10^{-2}$) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT |
| **PointNetGeometry** | - | - | 0.355(±0.020) | 0.467(±0.034) | 0.367(±0.039) | 0.369(±0.029) | 0.002(±0.001) | 0.002(±0.001) | 0.009(±0.001) | 0.012(±0.001) | 0.009(±0.001) | 0.012(±0.001) | 12.037(±0.725) | 15.991(±1.237) | 12.059(±0.726) | 16.023(±1.238) | 26.765(±1.835) | 36.775(±3.416) | 13.514(±0.818) | 17.995(±1.486) | 0.008(±0.001) | 0.011(±0.001) |
| | DANN | DEV | 0.354(±0.102) | 0.480(±0.095) | 0.420(±0.139) | 0.362(±0.102) | 0.001(±0.000) | 0.001(±0.000) | 0.010(±0.003) | 0.013(±0.003) | 0.009(±0.003) | 0.013(±0.003) | 12.976(±3.986) | 17.751(±3.737) | 12.999(±3.992) | 17.786(±3.743) | 28.599(±8.806) | 40.588(±8.503) | 14.536(±4.369) | 19.933(±4.183) | 0.009(±0.003) | 0.012(±0.003) |
| | DANN | IWV | 0.296(±0.006) | 0.412(±0.039) | 0.343(±0.014) | 0.307(±0.022) | 0.001(±0.001) | 0.001(±0.001) | 0.008(±0.000) | 0.011(±0.001) | 0.007(±0.000) | 0.011(±0.001) | 10.688(±0.243) | 15.097(±1.533) | 10.708(±0.243) | 15.128(±1.535) | 23.453(±0.646) | 34.398(±3.811) | 12.022(±0.301) | 16.966(±1.771) | 0.007(±0.000) | 0.010(±0.001) |
| | DANN | SB | 0.294(±0.004) | 0.396(±0.027) | 0.338(±0.007) | 0.303(±0.014) | 0.001(±0.001) | 0.001(±0.001) | 0.008(±0.000) | 0.011(±0.001) | 0.007(±0.000) | 0.010(±0.001) | 10.633(±0.167) | 14.479(±1.046) | 10.653(±0.168) | 14.510(±1.048) | 23.333(±0.471) | 32.962(±2.726) | 11.957(±0.205) | 16.278(±1.226) | 0.007(±0.000) | 0.010(±0.001) |
| | DANN | TB | 0.294(±0.004) | 0.383(±0.007) | 0.337(±0.005) | 0.298(±0.014) | 0.001(±0.001) | 0.001(±0.001) | 0.008(±0.000) | 0.010(±0.000) | 0.007(±0.000) | 0.010(±0.000) | 10.628(±0.161) | 13.975(±0.265) | 10.649(±0.163) | 14.004(±0.266) | 23.345(±0.484) | 31.721(±0.852) | 11.957(±0.205) | 15.698(±0.350) | 0.007(±0.000) | 0.009(±0.000) |
| | CMD | DEV | 0.507(±0.000) | 0.620(±0.001) | 0.629(±0.004) | 0.515(±0.003) | 0.001(±0.000) | 0.001(±0.000) | 0.014(±0.000) | 0.017(±0.000) | 0.013(±0.000) | 0.017(±0.000) | 18.950(±0.009) | 23.261(±0.040) | 18.981(±0.009) | 23.304(±0.041) | 41.807(±0.034) | 53.096(±0.143) | 21.086(±0.015) | 26.089(±0.054) | 0.013(±0.000) | 0.016(±0.000) |
| | CMD | IWV | 0.313(±0.017) | 0.397(±0.021) | 0.341(±0.017) | 0.295(±0.010) | 0.001(±0.001) | 0.001(±0.001) | 0.008(±0.001) | 0.011(±0.001) | 0.008(±0.001) | 0.010(±0.001) | 11.372(±0.682) | 14.542(±0.836) | 11.393(±0.683) | 14.574(±0.838) | 25.074(±1.647) | 32.984(±1.909) | 12.810(±0.796) | 16.342(±0.964) | 0.008(±0.000) | 0.010(±0.001) |
| | CMD | SB | 0.306(±0.013) | 0.393(±0.017) | 0.344(±0.012) | 0.294(±0.011) | 0.002(±0.001) | 0.002(±0.001) | 0.008(±0.000) | 0.011(±0.001) | 0.008(±0.000) | 0.010(±0.001) | 11.088(±0.533) | 14.369(±0.678) | 11.108(±0.533) | 14.400(±0.679) | 24.390(±1.290) | 32.786(±1.757) | 12.480(±0.607) | 16.156(±0.769) | 0.007(±0.000) | 0.010(±0.000) |
| | CMD | TB | 0.309(±0.013) | 0.392(±0.017) | 0.339(±0.018) | 0.291(±0.013) | 0.002(±0.000) | 0.002(±0.001) | 0.008(±0.000) | 0.011(±0.001) | 0.008(±0.000) | 0.010(±0.001) | 11.201(±0.529) | 14.349(±0.669) | 11.222(±0.530) | 14.380(±0.671) | 24.677(±1.285) | 32.669(±1.705) | 12.596(±0.599) | 16.095(±0.742) | 0.008(±0.000) | 0.010(±0.000) |
| | Deep Coral | DEV | 0.361(±0.103) | 2.135(±3.368) | 0.417(±0.139) | 0.753(±0.762) | 0.002(±0.001) | 0.013(±0.023) | 0.010(±0.003) | 0.028(±0.032) | 0.009(±0.003) | 0.032(±0.041) | 13.229(±4.044) | 57.507(±81.842) | 13.251(±4.049) | 57.438(±81.637) | 29.221(±9.058) | 112.987(±151.069) | 14.840(±4.437) | 85.342(±133.435) | 0.009(±0.003) | 0.024(±0.026) |
| | Deep Coral | IWV | 0.303(±0.018) | 0.392(±0.028) | 0.336(±0.010) | 0.308(±0.005) | 0.002(±0.001) | 0.001(±0.000) | 0.008(±0.001) | 0.011(±0.001) | 0.008(±0.001) | 0.010(±0.001) | 10.987(±0.722) | 14.320(±1.123) | 11.008(±0.724) | 14.351(±1.125) | 24.161(±1.833) | 32.534(±2.994) | 12.354(±0.805) | 16.100(±1.305) | 0.007(±0.001) | 0.010(±0.001) |
| | **Deep Coral** | **SB** | 0.290(±0.005) | **0.378(±0.011)** | 0.337(±0.002) | 0.298(±0.011) | 0.002(±0.001) | 0.002(±0.001) | 0.008(±0.000) | 0.010(±0.000) | 0.007(±0.000) | 0.010(±0.000) | 10.458(±0.189) | 13.767(±0.434) | 10.478(±0.189) | 13.796(±0.435) | 22.812(±0.552) | 31.249(±1.240) | 11.776(±0.215) | 15.487(±0.515) | 0.007(±0.000) | 0.009(±0.000) |
| | Deep Coral | TB | 0.290(±0.005) | 0.378(±0.011) | 0.337(±0.002) | 0.298(±0.011) | 0.002(±0.001) | 0.002(±0.001) | 0.008(±0.000) | 0.010(±0.000) | 0.007(±0.000) | 0.010(±0.000) | 10.458(±0.189) | 13.767(±0.434) | 10.478(±0.189) | 13.796(±0.435) | 22.812(±0.552) | 31.249(±1.240) | 11.776(±0.215) | 15.487(±0.515) | 0.007(±0.000) | 0.009(±0.000) |

Table 19 shows that UDA algorithms can boost target performance compared to the unregularized baseline model. However compared to our chosen benchmarking design in Table 8, both the performance of the unregularized baseline as well as the one of the best performing UDA method is worse, which supports our choice of explicitly conditioning on scalar parameters in the main benchmark.

### H.2 TWO-DIMENSIONAL SHIFTS

Defining shifts based on one parameter allows for controlled experiments, also given that the parameters were picked based on preliminary experiments (see Appendix C) and consultation with domain experts. In real-world scenarios, however, distribution shifts often affect multiple parameters simultaneously rather than only a single one. It is therefore important to investigate the performance of the benchmarked UDA algorithms under multidimensional parameter shifts. As a step in this direction, we provide an ablation on the *electric motor design* dataset for a two-dimensional parameter shift.

To be concise, we jointly shift the rotor slot diameter $d_{r3}$ (parameter shift in the main benchmark) and the angle between the magnets $\alpha_r$. Table 20 shows the corresponding two-dimensional distribution shift between the source and the target domain.

Table 20: Parameter ranges for the two-dimensional distribution shift on the *electric motor design* dataset.

| Parameter | Source range | Target range |
|---|---|---|
| Rotor slot diameter 3 $d_{r3}$ $(mm)$ | [100, 120) | [120, 126] |
| Angle between magnets $\alpha_r$ $(°)$ | [119, 153) | [153, 170] |

We train all models with each UDA algorithm following the procedure in Section 4.5 and Appendix F.1, and report the results in Table 21.

Table 21: RMSE (mean ± std over 4 seeds) on the *electric motor design* dataset at with a two-dimensional distribution shift in parameter space. Values are target domain errors (lower is better). Bold marks the overall best model + UDA algorithm + model selection combination. For each architecture, the unregularized baseline row is shaded beige, whereas the best UDA + selection within that architecture is underlined and shaded green.

| Model | DA Algorithm | Model Selection | All Fields Normalized Avg (-) | | Deformation (m) | | Logarithmic Strain ($\times10^{-2}$) | | Principal Strain ($\times10^{-2}$) | | Stress (MPa) | | Cauchy Stress (MPa) | | Mises Stress (MPa) | | Principal Stress (MPa) | | Total Strain ($\times10^{-2}$) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT | SRC | TGT |
| GraphSAGE | - | - | 0.324(±0.025) | 0.379(±0.017) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.001) | 0.010(±0.000) | 0.008(±0.001) | 0.009(±0.000) | 11.027(±0.778) | 12.953(±0.502) | 11.045(±0.775) | 12.984(±0.507) | 24.438(±1.730) | 29.664(±0.853) | 12.510(±0.814) | 14.816(±0.481) | 0.007(±0.001) | 0.009(±0.000) |
| | DANN | DEV | 0.284(±0.013) | 0.382(±0.067) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.000) | 0.011(±0.002) | 0.007(±0.000) | 0.010(±0.002) | 10.412(±0.403) | 14.196(±2.641) | 10.431(±0.404) | 14.224(±2.646) | 22.782(±0.904) | 32.960(±6.648) | 11.857(±0.406) | 16.314(±3.055) | 0.007(±0.000) | 0.010(±0.002) |
| | DANN | IWV | 0.284(±0.012) | 0.342(±0.010) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 0.007(±0.000) | 0.009(±0.000) | 10.439(±0.388) | 12.620(±0.368) | 10.457(±0.390) | 12.647(±0.370) | 22.994(±0.840) | 28.910(±1.073) | 11.918(±0.354) | 14.474(±0.420) | 0.007(±0.000) | 0.009(±0.000) |
| | DANN | SB | 0.284(±0.008) | 0.353(±0.019) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.000) | 0.010(±0.001) | 0.007(±0.000) | 0.010(±0.001) | 10.402(±0.294) | 13.056(±0.748) | 10.421(±0.295) | 13.084(±0.749) | 22.890(±0.806) | 30.202(±1.998) | 11.882(±0.288) | 14.993(±0.825) | 0.007(±0.000) | 0.009(±0.001) |
| | DANN | TB | 0.299(±0.013) | 0.343(±0.008) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 10.957(±0.457) | 12.640(±0.246) | 10.977(±0.459) | 12.666(±0.246) | 24.296(±1.127) | 28.930(±0.399) | 12.430(±0.514) | 14.460(±0.225) | 0.007(±0.000) | 0.009(±0.000) |
| | CMD | DEV | 0.306(±0.003) | 0.359(±0.011) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.000) | 0.010(±0.000) | 0.008(±0.000) | 0.010(±0.000) | 11.195(±0.100) | 13.203(±0.465) | 11.220(±0.104) | 13.230(±0.468) | 24.742(±0.244) | 30.241(±1.550) | 12.652(±0.144) | 15.065(±0.552) | 0.008(±0.000) | 0.009(±0.000) |
| | CMD | IWV | 0.287(±0.023) | 0.347(±0.023) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.001) | 0.010(±0.001) | 0.007(±0.001) | 0.009(±0.001) | 10.515(±0.823) | 12.812(±0.849) | 10.536(±0.827) | 12.837(±0.850) | 23.067(±2.069) | 29.412(±2.331) | 11.924(±0.882) | 14.653(±0.953) | 0.007(±0.001) | 0.009(±0.001) |
| | CMD | SB | 0.278(±0.021) | 0.340(±0.018) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.001) | 0.009(±0.000) | 0.007(±0.001) | 0.009(±0.000) | 10.176(±0.756) | 12.534(±0.642) | 10.192(±0.754) | 12.560(±0.644) | 22.292(±1.968) | 28.655(±1.563) | 11.580(±0.827) | 14.341(±0.683) | 0.007(±0.001) | 0.009(±0.000) |
| | CMD | TB | 0.290(±0.027) | 0.343(±0.017) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.001) | 0.009(±0.000) | 0.007(±0.001) | 0.009(±0.000) | 10.609(±0.933) | 12.608(±0.581) | 10.627(±0.933) | 12.634(±0.582) | 23.292(±2.338) | 28.678(±1.334) | 12.061(±1.045) | 14.410(±0.634) | 0.007(±0.001) | 0.009(±0.000) |
| | Deep Coral | DEV | 0.303(±0.006) | 0.358(±0.002) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.000) | 0.010(±0.000) | 0.008(±0.000) | 0.010(±0.000) | 11.090(±0.161) | 13.184(±0.057) | 11.110(±0.160) | 13.210(±0.057) | 24.492(±0.274) | 30.206(±0.372) | 12.531(±0.176) | 15.040(±0.128) | 0.007(±0.000) | 0.009(±0.000) |
| | Deep Coral | IWV | 0.276(±0.002) | 0.334(±0.007) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 0.007(±0.000) | 0.009(±0.000) | 10.142(±0.089) | 12.322(±0.261) | 10.161(±0.089) | 12.348(±0.262) | 22.259(±0.186) | 28.271(±0.656) | 11.540(±0.087) | 14.144(±0.288) | 0.007(±0.000) | 0.008(±0.000) |
| | Deep Coral | SB | 0.278(±0.002) | 0.339(±0.007) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 0.007(±0.000) | 0.009(±0.000) | 10.223(±0.101) | 12.518(±0.253) | 10.242(±0.101) | 12.544(±0.253) | 22.490(±0.309) | 28.784(±0.715) | 11.644(±0.129) | 14.374(±0.295) | 0.007(±0.000) | 0.009(±0.000) |
| | Deep Coral | TB | 0.281(±0.012) | 0.336(±0.010) | 0.001(±0.000) | 0.000(±0.000) | 0.008(±0.000) | 0.009(±0.000) | 0.007(±0.000) | 0.009(±0.000) | 10.294(±0.385) | 12.367(±0.350) | 10.313(±0.385) | 12.393(±0.351) | 22.589(±0.841) | 28.284(±0.679) | 11.697(±0.391) | 14.174(±0.346) | 0.007(±0.000) | 0.008(±0.000) |
| PointNet | - | - | 0.404(±0.061) | 0.441(±0.064) | 0.001(±0.000) | 0.001(±0.000) | 0.010(±0.001) | 0.011(±0.002) | 0.010(±0.002) | 0.011(±0.002) | 13.536(±2.116) | 14.947(±2.259) | 13.530(±2.091) | 14.957(±2.241) | 29.759(±4.853) | 33.860(±5.135) | 15.433(±2.479) | 17.310(±2.587) | 0.009(±0.001) | 0.010(±0.001) |
| | DANN | DEV | 0.333(±0.079) | 0.383(±0.077) | 0.001(±0.000) | 0.000(±0.000) | 0.009(±0.002) | 0.010(±0.002) | 0.009(±0.002) | 0.010(±0.002) | 12.052(±2.867) | 14.014(±2.877) | 12.056(±2.846) | 14.030(±2.866) | 26.183(±6.669) | 31.715(±6.844) | 13.702(±3.348) | 16.135(±3.416) | 0.008(±0.002) | 0.010(±0.002) |
| | DANN | IWV | 0.344(±0.080) | 0.379(±0.074) | 0.001(±0.000) | 0.000(±0.000) | 0.009(±0.002) | 0.010(±0.002) | 0.009(±0.002) | 0.010(±0.002) | 12.476(±2.926) | 13.850(±2.763) | 12.469(±2.909) | 13.860(±2.750) | 27.020(±6.762) | 30.977(±6.339) | 14.155(±3.392) | 15.937(±3.286) | 0.008(±0.002) | 0.009(±0.002) |
| | DANN | SB | 0.320(±0.066) | 0.355(±0.060) | 0.001(±0.000) | 0.000(±0.000) | 0.009(±0.002) | 0.010(±0.002) | 0.008(±0.002) | 0.009(±0.002) | 11.565(±2.387) | 12.905(±2.214) | 11.568(±2.361) | 12.921(±2.199) | 25.087(±5.559) | 28.844(±5.027) | 13.108(±2.779) | 14.832(±2.609) | 0.008(±0.002) | 0.009(±0.001) |
| | DANN | TB | 0.320(±0.066) | 0.355(±0.060) | 0.001(±0.000) | 0.000(±0.000) | 0.009(±0.002) | 0.010(±0.002) | 0.008(±0.002) | 0.009(±0.002) | 11.565(±2.387) | 12.905(±2.214) | 11.568(±2.361) | 12.921(±2.199) | 25.087(±5.559) | 28.844(±5.027) | 13.108(±2.779) | 14.832(±2.609) | 0.008(±0.002) | 0.009(±0.001) |
| | CMD | DEV | 0.338(±0.064) | 0.398(±0.042) | 0.001(±0.000) | 0.000(±0.000) | 0.009(±0.002) | 0.011(±0.001) | 0.009(±0.002) | 0.011(±0.001) | 12.173(±2.395) | 14.649(±1.516) | 12.178(±2.381) | 14.665(±1.508) | 26.658(±5.253) | 33.615(±3.357) | 13.839(±2.688) | 16.815(±1.806) | 0.008(±0.002) | 0.010(±0.001) |
| | CMD | IWV | 0.368(±0.031) | 0.414(±0.009) | 0.001(±0.000) | 0.001(±0.000) | 0.010(±0.001) | 0.011(±0.000) | 0.009(±0.001) | 0.011(±0.000) | 13.275(±1.104) | 15.175(±0.343) | 13.277(±1.085) | 15.183(±0.328) | 28.952(±2.713) | 34.480(±1.329) | 15.004(±1.304) | 17.411(±0.451) | 0.009(±0.001) | 0.010(±0.000) |
| | CMD | SB | 0.355(±0.037) | 0.403(±0.032) | 0.001(±0.000) | 0.000(±0.000) | 0.010(±0.001) | 0.011(±0.001) | 0.009(±0.001) | 0.011(±0.001) | 12.821(±1.387) | 14.758(±1.300) | 12.828(±1.367) | 14.773(±1.293) | 27.878(±3.425) | 33.618(±3.351) | 14.537(±1.634) | 16.989(±1.460) | 0.009(±0.001) | 0.010(±0.001) |
| | CMD | TB | 0.334(±0.059) | 0.380(±0.040) | 0.001(±0.000) | 0.000(±0.000) | 0.009(±0.002) | 0.010(±0.001) | 0.009(±0.002) | 0.010(±0.001) | 11.996(±2.224) | 13.845(±1.431) | 12.008(±2.217) | 13.868(±1.418) | 26.384(±4.897) | 31.387(±3.242) | 13.665(±2.474) | 15.919(±1.717) | 0.008(±0.001) | 0.009(±0.001) |
| | Deep Coral | DEV | 0.350(±0.039) | 0.383(±0.038) | 0.001(±0.000) | 0.000(±0.000) | 0.009(±0.001) | 0.010(±0.001) | 0.009(±0.001) | 0.010(±0.001) | 12.608(±1.434) | 13.923(±1.438) | 12.604(±1.396) | 13.944(±1.431) | 27.257(±3.653) | 31.231(±3.564) | 14.263(±1.750) | 16.034(±1.720) | 0.008(±0.001) | 0.009(±0.001) |
| | Deep Coral | IWV | 0.351(±0.042) | 0.384(±0.040) | 0.001(±0.000) | 0.000(±0.000) | 0.009(±0.001) | 0.010(±0.001) | 0.009(±0.001) | 0.010(±0.001) | 12.664(±1.539) | 13.988(±1.552) | 12.664(±1.506) | 13.997(±1.523) | 27.397(±3.913) | 31.369(±3.801) | 14.340(±1.891) | 16.108(±1.847) | 0.009(±0.001) | 0.009(±0.001) |
| | Deep Coral | SB | 0.358(±0.042) | 0.393(±0.042) | 0.001(±0.000) | 0.000(±0.000) | 0.010(±0.001) | 0.011(±0.001) | 0.009(±0.001) | 0.010(±0.001) | 12.915(±1.522) | 14.274(±1.566) | 12.920(±1.505) | 14.296(±1.557) | 28.150(±3.736) | 32.076(±3.728) | 14.713(±1.880) | 16.505(±1.872) | 0.009(±0.001) | 0.010(±0.001) |
| | Deep Coral | TB | 0.341(±0.045) | 0.371(±0.045) | 0.001(±0.000) | 0.000(±0.000) | 0.009(±0.001) | 0.010(±0.001) | 0.009(±0.001) | 0.010(±0.001) | 12.259(±1.691) | 13.481(±1.683) | 12.262(±1.653) | 13.505(±1.673) | 26.681(±3.884) | 30.213(±3.909) | 13.887(±1.948) | 15.502(±1.950) | 0.008(±0.001) | 0.009(±0.001) |
| **Transolver** | - | - | 0.134(±0.009) | 0.146(±0.010) | 0.001(±0.000) | 0.000(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 4.411(±0.261) | 4.836(±0.289) | 4.419(±0.262) | 4.845(±0.290) | 9.486(±0.610) | 10.368(±0.615) | 5.252(±0.329) | 5.820(±0.350) | 0.003(±0.000) | 0.003(±0.000) |
| | DANN | DEV | 0.111(±0.008) | 0.121(±0.009) | 0.001(±0.000) | 0.000(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 3.920(±0.260) | 4.347(±0.333) | 3.927(±0.260) | 4.355(±0.333) | 8.341(±0.634) | 9.329(±0.814) | 4.688(±0.317) | 5.236(±0.393) | 0.003(±0.000) | 0.003(±0.000) |
| | DANN | IWV | 0.107(±0.008) | 0.115(±0.009) | 0.001(±0.000) | 0.000(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 3.817(±0.310) | 4.114(±0.336) | 3.824(±0.311) | 4.121(±0.336) | 8.048(±0.705) | 8.699(±0.742) | 4.569(±0.328) | 4.982(±0.389) | 0.003(±0.000) | 0.003(±0.000) |
| | **DANN** | **SB** | 0.099(±0.004) | **0.110(±0.005)** | 0.001(±0.000) | 0.000(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.002(±0.000) | 0.003(±0.000) | 3.549(±0.157) | 3.945(±0.186) | 3.555(±0.157) | 3.952(±0.186) | 7.368(±0.389) | 8.361(±0.440) | 4.228(±0.188) | 4.772(±0.215) | 0.002(±0.000) | 0.003(±0.000) |
| | DANN | TB | 0.102(±0.004) | 0.111(±0.006) | 0.001(±0.000) | 0.000(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 3.644(±0.124) | 3.968(±0.196) | 3.650(±0.124) | 3.975(±0.197) | 7.540(±0.360) | 8.386(±0.453) | 4.332(±0.204) | 4.802(±0.232) | 0.002(±0.000) | 0.003(±0.000) |
| | CMD | DEV | 0.107(±0.008) | 0.119(±0.009) | 0.001(±0.000) | 0.000(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 3.796(±0.229) | 4.251(±0.316) | 3.803(±0.229) | 4.259(±0.317) | 8.091(±0.555) | 9.037(±0.762) | 4.591(±0.277) | 5.141(±0.363) | 0.002(±0.000) | 0.003(±0.000) |
| | CMD | IWV | 0.102(±0.006) | 0.112(±0.007) | 0.001(±0.000) | 0.000(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 3.631(±0.218) | 3.990(±0.264) | 3.638(±0.219) | 3.998(±0.264) | 7.714(±0.389) | 8.407(±0.616) | 4.402(±0.203) | 4.838(±0.300) | 0.002(±0.000) | 0.003(±0.000) |
| | CMD | SB | 0.104(±0.005) | 0.115(±0.008) | 0.001(±0.000) | 0.000(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 3.701(±0.211) | 4.100(±0.282) | 3.707(±0.211) | 4.107(±0.283) | 7.820(±0.377) | 8.685(±0.662) | 4.463(±0.187) | 4.973(±0.325) | 0.002(±0.000) | 0.003(±0.000) |
| | CMD | TB | 0.101(±0.005) | 0.110(±0.008) | 0.001(±0.000) | 0.000(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 3.570(±0.169) | 3.922(±0.274) | 3.577(±0.169) | 3.929(±0.275) | 7.658(±0.316) | 8.303(±0.630) | 4.344(±0.200) | 4.761(±0.334) | 0.002(±0.000) | 0.003(±0.000) |
| | Deep Coral | DEV | 0.107(±0.004) | 0.116(±0.005) | 0.001(±0.000) | 0.000(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 3.794(±0.126) | 4.121(±0.180) | 3.800(±0.127) | 4.128(±0.180) | 8.042(±0.300) | 8.790(±0.414) | 4.571(±0.142) | 5.033(±0.197) | 0.002(±0.000) | 0.003(±0.000) |
| | Deep Coral | IWV | 0.105(±0.004) | 0.115(±0.003) | 0.001(±0.000) | 0.000(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 3.721(±0.142) | 4.113(±0.087) | 3.727(±0.142) | 4.119(±0.088) | 7.854(±0.253) | 8.748(±0.210) | 4.482(±0.153) | 4.994(±0.131) | 0.002(±0.000) | 0.003(±0.000) |
| | Deep Coral | SB | 0.106(±0.004) | 0.116(±0.003) | 0.001(±0.000) | 0.000(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 3.764(±0.104) | 4.157(±0.104) | 3.770(±0.104) | 4.164(±0.104) | 7.949(±0.270) | 8.832(±0.205) | 4.527(±0.145) | 5.040(±0.139) | 0.002(±0.000) | 0.003(±0.000) |
| | Deep Coral | TB | 0.105(±0.003) | 0.113(±0.003) | 0.001(±0.000) | 0.000(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 0.003(±0.000) | 3.701(±0.089) | 4.016(±0.109) | 3.707(±0.089) | 4.023(±0.109) | 7.933(±0.293) | 8.514(±0.229) | 4.508(±0.136) | 4.891(±0.114) | 0.002(±0.000) | 0.003(±0.000) |

Comparing these results with the original one-dimensional shift (Table 8), two observations stand out: (i) For two out of three architectures, both the unregularized baseline and the best UDA algorithm and model selection combination exhibit higher errors in the average field NRMSE than in the one-dimensional shift setting, confirming that the two-dimensional shift is a more challenging task. (ii) The relative improvements over the unregularized baselines are larger, indicating that UDA training provides greater benefits under this more challenging distribution shift.

These findings highlight the potential of UDA to handle increasingly complex distribution shifts, underscoring its practical relevance for real-world applications.