DOES UNSUPERVISED DOMAIN ADAPTATION IMPROVE THE ROBUSTNESS OF AMORTIZED BAYESIAN INFERENCE? A SYSTEMATIC EVALUATION

Lasse Elsemüller[1], Valentin Pratz[1,2], Mischa von Krause[1], Paul-Christian Bürkner[3] & Stefan T. Radev[4]

[1] Heidelberg University, Germany, [2] Zuse School ELIZA, Germany,
[3] TU Dortmund University, Germany, [4] Rensselaer Polytechnic Institute, USA

ABSTRACT

Neural networks are fragile when confronted with data that significantly deviates from their training distribution. This is true in particular for simulation-based inference methods, such as neural amortized Bayesian inference (ABI), where models trained on simulated data are deployed on noisy real-world observations. Recent robust approaches employ unsupervised domain adaptation (UDA) to match the embedding spaces of simulated and observed data. However, the lack of comprehensive evaluations across different domain mismatches raises concerns about the reliability in high-stakes applications. We address this gap by systematically testing UDA approaches across a wide range of misspecification scenarios in both a controlled and a high-dimensional benchmark. We demonstrate that aligning summary spaces between domains effectively mitigates the impact of unmodeled phenomena or noise. However, the same alignment mechanism can lead to failures under prior misspecifications—a critical finding with practical consequences. Our results underscore the need for careful consideration of misspecification types when using UDA techniques to increase the robustness of ABI in practice.

1 INTRODUCTION

Synthetic data can augment numerous real-world applications (Savage, 2023), including complex statistical workflows. In line with this perspective, amortized Bayesian inference (ABI; Gershman & Goodman, 2014) redefines the classical sampling problem in Bayesian estimation by training generative neural networks on simulations derived from computational models (Bürkner et al., 2023). The trained neural networks are then deployed to efficiently solve inference tasks as diverse as inferring evolutionary parameters (Avecilla et al., 2022) or gravitational waves (Pacilio et al., 2024).

Evidently, the faithfulness of any simulation-based method rests on a critical assumption: That statistical patterns learned from simulated data can be extrapolated to real observations. This assumption inevitably situates ABI in a domain-shift regime, exacerbated by the degree of potential mismatch between model simulations and reality. As such, *robustness to model misspecification* has been identified as the primary challenge for amortized methods in different fields (Dingeldein et al., 2024; Rainforth et al., 2024; Cannon et al., 2022).

Unsupervised Domain Adaptation (UDA) studies the transfer of knowledge from a labeled source domain to an unlabeled target domain. It aims to mitigate domain shifts by aligning the *embedding spaces* of the two domains. This property makes UDA a promising approach for addressing domain shifts in ABI, as the latter typically combines inference with embedding high-dimensional data into *learned summary statistics* (Radev et al., 2020; Chan et al., 2018). Indeed, recent research has underscored the critical role of *in-distribution* summary statistics for achieving robust simulation-based inference (Schmitt et al., 2023; Frazier et al., 2024; Huang et al., 2023; Wehenkel et al., 2024).

So far, only two pioneering studies (Swierc et al., 2024; Huang et al., 2023) have explored the potential of UDA methods for robustifying simulation-based inference. Both approaches align the embedding spaces by minimizing the maximum mean discrepancy (MMD; Gretton et al., 2012) be-

tween simulated and observed summary statistics. However, despite their promising results, several gaps remain. In particular, Huang et al. (2023) did not make an explicit connection to UDA and explored a non-amortized approach. While Swierc et al. (2024) acknowledged the connection to UDA, their work focused on a specific gravitational lensing application. Both works mainly evaluated likelihood misspecification, leaving the behavior under prior shifts largely untapped. Finally, the utility of the widely used UDA method *domain-adversarial neural networks* (DANN; Ganin et al., 2016) remains completely unexplored. To address these gaps, we make the following contributions:

1. We adapt domain-adversarial neural networks for neural posterior estimation (NPE) and evaluate their utility for robust amortized Bayesian inference.
2. We categorize robust methods by inference targets, enabling a theoretical assessment of their strengths and limitations based on the source of misspecification.
3. We evaluate the robustness of UDA-based ABI methods across multiple misspecification scenarios in two benchmarks, confirming the central role of the source of misspecification.

2 Background

Amortized Bayesian Inference (ABI) Amortized methods are a subset of the simulation-based inference (SBI; Cranmer et al., 2020) family. Their defining characteristic is the ability to perform zero-shot inference on model parameters $\boldsymbol{\theta}$ by learning a conditional distribution $q(\boldsymbol{\theta} \mid \boldsymbol{x})$ that requires no further training or approximation algorithms (see Appendix A.1 for details). The *amortized distribution* $q(\boldsymbol{\theta} \mid \boldsymbol{x})$ is typically parameterized by a generative neural network that can generate random samples $\boldsymbol{\theta} \sim q(\boldsymbol{\theta} \mid \boldsymbol{x})$, akin to a standard Markov chain Monte Carlo (MCMC) sampler, but orders of magnitude faster. Following a potentially expensive simulation-based training phase, the network can be queried with any *new* data $\boldsymbol{x}_{\text{new}}$ to rapidly approximate the target distribution $p(\boldsymbol{\theta} \mid \boldsymbol{x}_{\text{new}})$. Initially dismissed as inefficient compared to sequential methods optimized for a specific data set $\boldsymbol{x}_{\text{obs}}$ (Papamakarios & Murray, 2016), amortized methods have since achieved notable successes across various domains (Bürkner et al., 2023; Zammit-Mangion et al., 2024).

Unsupervised Domain Adaptation (UDA) UDA is a subfield of transductive transfer learning where labeled data is only available for the source domain $\mathcal{D}_S = \{(\boldsymbol{x}_S^i, \boldsymbol{y}_S^i)\}_{i=1}^{N_S}$, distributed according to $p_S(\boldsymbol{x}, \boldsymbol{y})$, but not for the target domain $\mathcal{D}_T = \{\boldsymbol{x}_T^i\}_{i=1}^{N_T}$, distributed according to $p_T(\boldsymbol{x}_T, \boldsymbol{y}_T)$ (Johansson et al., 2019). UDA methods are based on the seminal theoretical works of Ben-David et al. (2006; 2010), who introduced generalization bounds for binary classification tasks that bound the risk in the target domain R_T of a hypothesis $h \in \mathcal{H}$:

$$R_T(h) \leq R_S(h) + d_{\mathcal{H}\Delta\mathcal{H}}(p_S, p_T) + \lambda_{\mathcal{H}}, \tag{1}$$

where $R_S(h)$ is the source domain risk, $d_{\mathcal{H}\Delta\mathcal{H}}(p_S, p_T)$ measures the divergence between the domain distributions, and $\lambda_{\mathcal{H}}$ is the minimum combined risk of the optimal hypothesis, $\lambda_{\mathcal{H}} = \inf_{h\in\mathcal{H}}|R_S(h) + R_T(h)|$ (Johansson et al., 2019). This suggests that domain adaptation from $\mathcal{D}_S$ to $\mathcal{D}_T$ can be facilitated by minimizing the divergence between the marginal domain distributions. Although the domain distribution divergence cannot be reduced directly, the representation divergence $d(\phi(\boldsymbol{x}_S), \phi(\boldsymbol{x}_T))$ from a transformation $\phi : \mathcal{X} \to \mathcal{Z}$ can be readily minimized (Ben-David et al., 2006). The core idea of UDA is thus twofold: (i) to minimize the source-domain error $R_S(h)$ during training, and (ii) to align the domain representations $\phi(\boldsymbol{x}_S)$ and $\phi(\boldsymbol{x}_T)$ to achieve *domain-invariant* embeddings that generalize to the target domain. UDA methods include discrepancy-based approaches, which minimize statistical divergences like the MMD between source and target embeddings (Tzeng et al., 2014), and, most prominently, adversarial-based approaches, such as Domain-Adversarial Neural Networks (DANN) (Ganin et al., 2016), which learn domain-invariant embeddings via a minimax game between a feature extractor and a domain classifier.

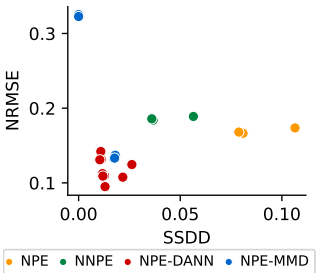


Figure 1: **Experiment 2**: Summary space domain distance (SSDD) vs. normalized root mean squared error (NRMSE) for a scale shift. We observe a sweet spot of domain alignment without losing important information.

The vast majority of UDA research, including its theoretical foundations, focuses on classification tasks (Redko et al., 2022; Ben-David et al., 2010; Liu et al., 2022), with some works on regression

tasks (Cortes & Mohri, 2014; Mansour et al., 2009) and only a few on generative tasks (Uppaal et al., 2024). More recently, UDA methods have been successfully applied to address simulation-to-reality (sim2real) problems (Ćiprijanović et al., 2020; Swierc et al., 2023) which seek to generalize patterns learned in a simulated source domain to a real-world target domain. These problems seem pertinent to any simulation-based method relying on data generation from imperfect models.

From Simulated to Real Domains The preceding discussion makes the connection between UDA and ABI immediately apparent: When the distance between the data distribution $p(\boldsymbol{x}_{\text{obs}})$ and the model-implied distribution $p(\boldsymbol{x}) = \mathbb{E}_{p(\boldsymbol{\theta})}\left[p(\boldsymbol{x} \mid \boldsymbol{\theta})\right]$ is non-zero, the risk of extrapolation error for atypical data $\boldsymbol{x}_{\text{obs}}$ may increase. Indeed, this behavior has been observed repeatedly in the context of SBI (Ward et al., 2022; Schmitt et al., 2023; Huang et al., 2023; Frazier et al., 2024). In particular, Frazier et al. (2024) notes that ABI is especially prone to "extrapolation bias" for observed summary statistics $\phi(\boldsymbol{x})$ that are far in the tails of the model-implied (i.e., prior predictive) density $p(\boldsymbol{x})$. The scenario can be equivalently stated by invoking the notion of a *typical set* (Cover & Thomas, 2012), which denotes a subset of the support of $p(\boldsymbol{x})$ where most of the probability mass concentrates around the entropy $H(p)$:

$$A_\epsilon = \{\boldsymbol{x} \in \mathcal{X} : |-\log p(\boldsymbol{x}) - H(p)| \leq \epsilon\}\,. \tag{2}$$

Accordingly, for any problem-specific ϵ, observed data $\boldsymbol{x}_{\text{obs}} \notin A_\epsilon$ may result in a biased posterior approximation $q(\boldsymbol{\theta} \mid \boldsymbol{x}_{\text{obs}})$. As further noted in the comprehensive theoretical exposition by Frazier et al. (2024), matching summary statistics $\phi(\boldsymbol{x}_{\text{obs}})$ to the model-implied distribution of $\phi(\boldsymbol{x})$ can be a useful heuristic for reducing extrapolation bias. This observation harmonizes with the UDA literature as well (Ben-David et al., 2010). Pre-asymptotically, the success of such matching depends on multiple factors, including (i) the type and hyperparameters of the matching method (see Figure 1); (ii) the degree and nature of domain mismatch; (iii) the complexity of the learning problem; and (iv) even the choice of success metric. Thus, a primary goal of this work is to systematically examine the effects of these factors on a variety of metrics that can index potential robustness gains.

3 Methods

3.1 Unsupervised Domain Adaptation for Amortized Bayesian Inference

We start with the observation that model misspecification in ABI (Schmitt et al., 2023), and also more generally in neural SBI, can naturally be framed as an UDA problem: Ground-truth parameter values are only available for the simulated source domain $\mathcal{D} = \{(\boldsymbol{x}^i, \boldsymbol{\theta}^i)\}_{i=1}^N$ but not the observed target domain $\mathcal{D}_{\text{obs}} = \{\boldsymbol{x}_{\text{obs}}^i\}_{i=1}^{N_{\text{obs}}}$. In most machine learning applications, the collection of reliable ground-truth values is costly but feasible, whereas in SBI, collecting ground-truth parameter values $\boldsymbol{\theta}_{\text{obs}}$ of observed data is typically impossible. A general optimization objective for NPE-UDA methods can be formulated by extending the standard negative log-posterior NPE objective:

$$\begin{aligned}\mathcal{L}_{\text{NPE-UDA}}(q, \phi) :&= \mathcal{L}_{\text{NPE}} + \lambda \cdot \mathcal{L}_{\text{UDA}} &(3)\\ &= \mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})p(\boldsymbol{x}_{\text{obs}})}\big[-\log q(\boldsymbol{\theta} \mid \phi(\boldsymbol{x})) + \lambda \cdot d(\phi(\boldsymbol{x}), \phi(\boldsymbol{x}_{\text{obs}}))\big], &(4)\end{aligned}$$

where λ controls the regularization weight of the UDA loss and $d(\cdot, \cdot)$ is a divergence measure that attains its global minimum if and only if $\phi(\boldsymbol{x}) = \phi(\boldsymbol{x}_{\text{obs}})$.

$\mathcal{L}_{\text{NPE-UDA}}$ incurs a trade-off between approximation performance in the simulated domain and domain divergence in the summary space, depending on the degree of domain mismatch. In the well-specified case, $p(\boldsymbol{x}) = p(\boldsymbol{x}_{\text{obs}})$, $\mathcal{L}_{\text{NPE-UDA}}$ reduces to the standard NPE loss. In the misspecified case, $p(\boldsymbol{x}) \neq p(\boldsymbol{x}_{\text{obs}})$, the summary network ϕ optimizes the summary statistics to both *maximize information extraction* in the simulated domain and *minimize domain shift* in summary space. Thereby, the approximator $q(\boldsymbol{\theta} \mid \phi(\boldsymbol{x}))$ needs to rely on domain-invariant information shared between the simulated and the observed domain. The common UDA assumption that there exists a low-error hypothesis for both domains (Redko et al., 2022, cf. Eq.1) suggests an upper bound on the amount of domain shift that can be handled by NPE-UDA methods. Next, we formulate two NPE-UDA variants based on popular UDA methods with strong benchmark performance (Musgrave et al., 2021).

3.2 NPE-MMD

The maximum mean discrepancy (MMD; Gretton et al., 2012) is a popular probability integral metric in SBI, since it can be efficiently estimated from a finite number of samples (Bischoff et al.,

2024; Schmitt et al., 2023). For the same reason, it has been employed by various UDA works (Pan et al., 2010; Tzeng et al., 2014; Long et al., 2015) to measure the divergence between (transformed) samples from different domains. We categorize the combination of NPE and UDA based on MMD, such as the variants of Huang et al. (2023) and Swierc et al. (2024), as NPE-MMD. Choosing the MMD as $\mathcal{L}_{\text{UDA}}$, Eq. 3 becomes

$$\mathcal{L}_{\text{NPE-MMD}}(q, \phi) := \mathbb{E}_{p(\boldsymbol{\theta}, \boldsymbol{x})}\big[-\log q(\boldsymbol{\theta} \,|\, \phi(\boldsymbol{x}))\big] + \lambda \cdot \text{MMD}^2\big[\phi(\boldsymbol{x}) \,||\, \phi(\boldsymbol{x}_{\text{obs}})\big]. \tag{5}$$

The most important hyperparameter of NPE-MMD is the choice of kernel in the sample-based MMD estimator. In our experiments, we obtained good results with a sum of inverse multiquadric kernels (Ardizzone et al., 2018), but other choices have been explored in the context of robust ABI as well, such as (sums of) Gaussian kernels (Schmitt et al., 2023; Huang et al., 2023).

3.3 NPE-DANN

Domain-adversarial neural networks (DANN; Ganin et al., 2016), which have not been considered for NPE to date, introduce a domain classifier $\psi(\cdot)$ to reduce domain distance. Unlike typical adversarial training, which alternates between objectives, DANN achieves minimax optimization in a single-step update via a gradient reversal layer (Ganin et al., 2016). This layer flips the gradient sign from the classifier to the feature extractor (e.g., summary network) ϕ during backpropagation, encouraging the feature extractor to generate less domain-specific summary statistics. Similarly to NPE-MMD, DANN can be integrated into Eq. 3 to achieve NPE-DANN:

$$\mathcal{L}_{\text{NPE-DANN}}(q, \phi, \psi) := \mathbb{E}_{p(\boldsymbol{\theta}, \boldsymbol{x})}\big[-\log q(\boldsymbol{\theta} \,|\, \phi(\boldsymbol{x}))\big] + \lambda \cdot \mathcal{L}_D(\psi, \phi). \tag{6}$$

The discriminator loss $\mathcal{L}_D$ is given by:

$$\mathcal{L}_D(\psi, \phi) := -\mathbb{E}_{p(\boldsymbol{x})}\big[\log(p(\psi(\phi(\boldsymbol{x}))))\big] - \mathbb{E}_{p(\boldsymbol{x}_{\text{obs}})}\big[\log(1 - p(\psi(\phi(\boldsymbol{x}_{\text{obs}}))))\big], \tag{7}$$

where ψ is the domain classifier and the equation represents the binary cross-entropy loss on the domains, where a gradient reversal layer enables updating ϕ and ψ in opposing directions.

While DANN is a powerful and popular UDA method (Zhou et al., 2022), it has two important drawbacks. First, the unstable training dynamics and convergence issues generally associated with adversarial learning can also occur with DANN (Sener et al., 2016; Sun et al., 2019). Second, adversarial training adds new hyperparameters, including the domain classifier architecture, an optional weight for gradient reversal balance (Ganin et al., 2016), and stabilization techniques like label smoothing (Zhang et al., 2023). Notably, although λ is a shared hyperparameter in NPE-MMD and NPE-DANN, its effect on training dynamics will vary across applications due to differing $\mathcal{L}_{\text{UDA}}$ scales.

3.4 What Is the Target of Robustness?

To better understand the strengths and limitations of robust methods, including NPE-UDA, we suggest to distinguish between the following inference goals:

- **Target 1**: The analytic (true) posterior $p(\boldsymbol{\theta} \mid \boldsymbol{x}_{\text{obs}}) \propto p(\boldsymbol{x}_{\text{obs}} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})$ of the assumed probabilistic model given the observed data $\boldsymbol{x}_{\text{obs}}$.
- **Target 2**: A posterior $p(\boldsymbol{\theta} \mid \tilde{\boldsymbol{x}}_{\text{obs}}) \propto p(\tilde{\boldsymbol{x}}_{\text{obs}} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})$ of the assumed probabilistic model given *adjusted data* $\tilde{\boldsymbol{x}}_{\text{obs}}$.
- **Target 3**: A posterior $\tilde{p}(\boldsymbol{\theta} \mid \boldsymbol{x}_{\text{obs}}) \propto p(\boldsymbol{x}_{\text{obs}} \mid \boldsymbol{\theta})\, \tilde{p}(\boldsymbol{\theta})$ from an *adjusted prior* $\tilde{p}(\boldsymbol{\theta})$ given the observed data $\boldsymbol{x}_{\text{obs}}$.

Target 1 is the most common target in Bayesian inference. Classical approximation methods such as MCMC almost always consider this target (Carpenter et al., 2017). **Target 2**, an explicit deviation from the true posterior, is often targeted by methods that seek to improve the robustness of Bayesian inference. Their goal is to reduce the influence of unmodeled phenomena in $\boldsymbol{x}_{\text{obs}}$, such as additional noise or external contamination, by approximating a target posterior $p(\boldsymbol{\theta} \mid \tilde{\boldsymbol{x}}_{\text{obs}})$ based on denoised or uncontaminated data $\tilde{\boldsymbol{x}}_{\text{obs}}$. This can be achieved either explicitly, by transforming $\boldsymbol{x}_{\text{obs}}$ into $\tilde{\boldsymbol{x}}_{\text{obs}}$, or implicitly, by using an *adjusted (implicit) likelihood* $\tilde{p}(\boldsymbol{x}_{\text{obs}} \mid \boldsymbol{\theta})$.

Since **Target 2** implies ignoring parts of the data that are in disagreement with the assumed probabilistic model, we expect corresponding methods to perform worse under prior misspecification:

When a data-generating parameter $\boldsymbol{\theta}^*$ is impossible or highly unlikely under the assumed prior, *ignoring conflicting information effectively reduces the amount of information available to counteract a poorly chosen prior*. Generalized Bayes approaches (Bissiri et al., 2016) also aim to reduce the influence of undesired parts of the data. They move away from the classical Bayes rule by replacing the likelihood with a loss function, which can be interpreted as an adjusted likelihood $\tilde{p}(\boldsymbol{x}_{\text{obs}} \mid \boldsymbol{\theta})$ according to **Target 2**. Lastly, **Target 3** can directly reduce the impact of prior misspecification by adjusting the prior based on $\boldsymbol{x}_{\text{obs}}$. However, compared to **Target 2**, it is more challenging to conceptualize the desired target priors $\tilde{p}(\boldsymbol{\theta})$ and posteriors $\tilde{p}(\boldsymbol{\theta} \mid \boldsymbol{x}_{\text{obs}})$ under model misspecification.

Given this categorization, what is the target of NPE-UDA? Unsurprisingly, the classic NPE loss $\mathcal{L}_{\text{NPE}}$ aims at **Target 1**. In contrast, the additional $\mathcal{L}_{\text{UDA}}$ loss governs the alignment of the summary space between simulated and observed data, effectively adjusting the observed data seen by the model. Thus, $\mathcal{L}_{\text{UDA}}$ introduces a shift towards **Target 2**, with λ governing its relative importance compared to **Target 1**. As hypothesized above, methods aiming at **Target 2** may not perform well under prior misspecification, which is confirmed for the NPE-UDA methods throughout our experiments. While Huang et al. (2023) suggested that their NPE-MMD variant is robust to prior mean shift, this conclusion was based on a single tested $\boldsymbol{x}_{\text{obs}}$ and our comprehensive evaluation could not replicate the result.

In line with our hypothesis and empirical results, Huang et al. (2023) observed that increasing values of λ encourage trading off the information content of $\boldsymbol{x}$ to minimize the domain distance in summary space, leading the posterior to converge to the assumed prior $p(\boldsymbol{\theta})$. Thus, the critical importance of the tunable hyperparameter λ in UDA contexts (Zellinger et al., 2021) directly translates to ABI applications, where λ controls a trade-off between *improving* approximation under likelihood misspecification and *degrading* approximation under prior misspecification.

4 Related Work

Robust Neural SBI Robustness in neural SBI has become a rapidly growing area of research, with most approaches enhancing robustness for a single data set at the cost of amortization, e.g., due to additional MCMC runs or post-hoc corrections. The majority of these approaches focuses on **Target 2** by incorporating an misspecification model (Ward et al., 2022), shifting observed summary statistics with low support (Kelly et al., 2023), reducing the influence of unmodeled data shifts via generalized SBI (Gao et al., 2023), or using the single-data-set NPE-MMD variant previously discussed (Huang et al., 2023). Focusing on **Target 1**, Siahkoohi et al. (2023) highlighted the role of the approximator's latent space in domain shifts and proposed a latent space correction based on the observed data $\boldsymbol{x}_{\text{obs}}$. Differently, Wang et al. (2024) focus on **Target 3** by using an upfront ABC run to filter the part of the parameter space causing the highest discrepancy between $\boldsymbol{x}$ and $\boldsymbol{x}_{\text{obs}}$.

Robust ABI In contrast, research on robustifying inference while retaining amortization has been sparse. Extending the scope of the training data via additive noise (Cranmer et al., 2020; Bernaerts et al., 2023), such as the spike-and-slab noise approach of Noisy NPE (NNPE; Ward et al., 2022), can be seen as a light modification to the simulator-implied likelihood as in **Target 2**, but requires strong assumptions about the misspecification-generating process. Wehenkel et al. (2024) also approach **Target 2** by framing domain shift as an optimal transport problem in summary space, but this requires *observed* "ground-truth" parameters $\boldsymbol{\theta}^*_{\text{obs}}$ that are hard to obtain in most ABI settings. Swierc et al. (2024) provided evidence for the potential of NPE-MMD for robust ABI but focused their evaluation on a gravitational lensing application with synthetically added noise. Finally, Glöckler et al. (2023) proposed an efficient regularization technique that can increase robustness against adversarial attacks and thus attain more reliable performance under **Target 1**.

5 Experiments

The previous two NPE-MMD approaches mainly evaluated performance against contamination (Huang et al., 2023), where a fraction of the sample is replaced with corrupted observations (Huber, 1981), or noise applied to all observations (Swierc et al., 2024). Both of these scenarios are cases of likelihood misspecification where ignoring noise is desirable (**Target 2**). To obtain a clearer insight into the strengths and limitations of NPE-UDA methods, we systematically evaluate the behavior of NPE-MMD and NPE-DANN in various likelihood/data and prior misspecification scenarios.

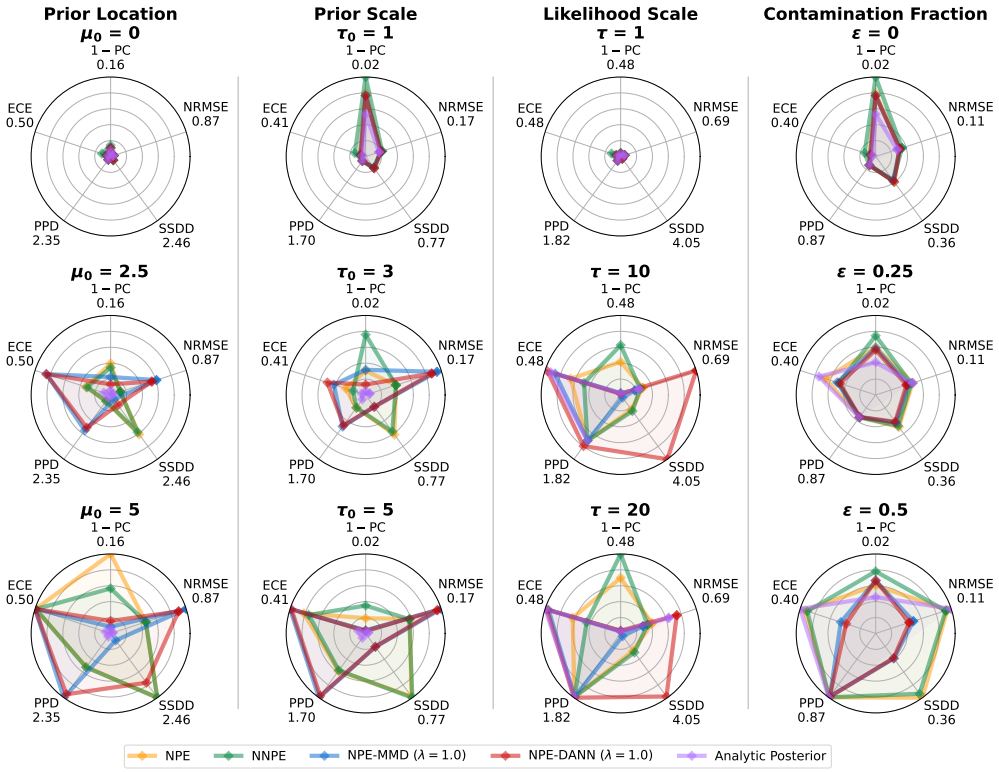


Figure 2: **Experiment 1**. Metrics of the methods in all misspecification scenarios (columns), averaged across 3 runs. The first row shows the well-specified setting, with misspecification increasing from top to bottom within each column. Metric values are centered at 0 and normalized by each column's/scenario's maximum value, which is displayed below the metric name at the border of each radar plot. Lower values indicate better performance for all metrics but SSDD. $1-$ PC $=$ $1-$ Posterior Contraction. NRMSE = Normalized Root Mean Squared Error. SSDD = Summary Space Domain Distance (MMD; not applicable for Analytic Posterior). PPD = Posterior Predictive Distance (MMD). ECE = Expected Calibration Error. NPE-UDA methods fail under prior misspecification but can be advantageous under likelihood misspecification, especially contamination.

Experiment 5.1 starts with a simple and controllable setting that allows for comparing the NPE-UDA methods not only against standard NPE and NNPE (Ward et al., 2022), but also the analytic posterior under **Target 1**. Afterwards, Experiment 5.2 explores whether the results can be replicated in a challenging setting with a high-dimensional parameter space. We evaluate a range of metrics: (i) normalized root mean squared error (NRMSE) to measure approximation error; (ii) expected calibration error (ECE) to measure probabilistic calibration; (iii) posterior contraction (PC) to measure information gain from prior to posterior; (iv) posterior predictive distance (PPD; via MMD), the only performance metric obtainable on $\boldsymbol{x}_{\text{obs}}$ in real-world settings; and (v) summary space domain distance (SSDD; via MMD) to measure domain alignment. Please refer to the Appendix for details concerning the metrics (B.2), Experiment 5.1 (B.3), and Experiment 5.2 (B.4).

5.1 Experiment 1 - 2D Gaussian Means: Controlled Setting

Setup Inspired by Schmitt et al. (2023), we set the stage with a simple and controllable task of modeling the means of a 2-dimensional Gaussian model, enabling the comparison with an analytic posterior. The well-specified setting uses a multivariate standard normal prior and an identity likelihood covariance matrix. We evaluate performance under increasing misspecification in two prior misspecification scenarios – prior location $\boldsymbol{\mu_0}$ and prior scale $\boldsymbol{\Sigma_0} = \tau_0 \boldsymbol{I}_2$ – and two likelihood misspecification scenarios – likelihood scale $\boldsymbol{\Sigma} = \tau \boldsymbol{I}_2$ and contamination ϵ (see Table B.1). For the contamination misspecification, a fraction ϵ of the observations is replaced by negative and positive vectors of the constant $c = 1.5$ to obtain atypical observations without affecting overall location or scale. Each simulated data set contains $M = 100$ exchangeable observations. All methods

Method	λ	Prior (MNIST → USPS) NRMSE ↓	PPD ↓	SSDD	Likelihood Scale NRMSE ↓	PPD ↓	SSDD	Contamination (Noise) NRMSE ↓	PPD ↓	SSDD	Contamination (Rows) NRMSE ↓	PPD ↓	SSDD
NPE	-	**0.252**	**0.081**	0.249	0.169	0.019	0.089	0.326	0.090	0.374	0.326	0.090	0.457
NNPE	-	0.312	0.106	0.218	0.186	0.027	0.043	**0.176**	**0.025**	0.038	0.202	0.032	0.036
NPE-DANN	0.01	0.342	0.150	0.019	**0.109**	0.015	0.020	0.231	0.045	0.020	0.174	0.033	0.024
NPE-DANN	0.10	0.344	0.152	0.016	0.110	0.014	0.012	0.207	0.034	0.013	**0.173**	0.028	0.014
NPE-DANN	1.00	0.373	0.169	0.067	0.135	0.017	0.011	0.252	0.047	0.013	0.223	0.039	0.014
NPE-MMD	0.01	0.312	0.134	0.026	0.134	**0.012**	0.018	0.266	0.053	0.020	0.264	0.054	0.017
NPE-MMD	0.10	0.322	0.141	0.013	0.323	0.085	0.000	0.253	0.048	0.013	0.175	**0.026**	0.011
NPE-MMD	1.00	0.393	0.185	-0.000	0.325	0.085	0.000	0.324	0.085	0.000	0.324	0.085	0.000

Table 1: **Experiment 2**. Metrics of the methods in all misspecification scenarios, averaged across 3 runs. NRMSE: Normalized Root Mean Squared Error (lower is better). PPD: Posterior Predictive Distance (NRMSE) (lower is better). SSDD: Summary Space Domain Distance (MMD). Lower values indicate better summary space alignment, but too much alignment (i.e., vanishing SSDD) can lead to an uninformative summary space (e.g., NPE-MMD with $\lambda = 1.00$).

train on $N = 49\,920$ well-specified data sets, with the NPE-UDA methods additionally exposed to $N_{\text{obs}} = 49\,920$ observed data sets, and are evaluated on $N_{\text{obs}} = 100$ observed data sets (unseen by NPE-UDA methods).

Results Figure 2 displays the results for all misspecification scenarios. We invert the meaning of the posterior contraction (PC) metric so that lower means better for all metrics and the performance of a method can mostly be inferred from its area. All methods perform reliably well in the well-specified case (first row), whereas we observe distinct but consistent patterns for the different methods under increasing mismatch. In the prior misspecification scenarios, all NPE methods perform poorly compared to the analytic posterior, but the NPE-UDA methods perform especially poorly in terms of NRMSE, PPD, and ECE.

In the likelihood scale misspecification scenario, the NPE methods are less sensitive to the misspecification than the analytic posterior. NPE-MMD successfully aligns the summary space between domains, leading to a slightly lower NRMSE but high ECE compared to NPE. NPE-DANN, on the other hand, fails to align the summary space for both misspecification levels, which translates to poor performance. This *drastic failure in the observed domain is not detectable in the simulated domain*, where all methods, even NPE-DANN in the $\tau = 20$ scenario, perform well (see Figure B2). While NNPE mostly performs similarly to standard NPE, with lower posterior contraction resulting from its noisier training in most settings, it achieves better calibration and a slightly lower NRMSE than NPE for likelihood scale shifts. In the contamination scenario, deviating from the true posterior via **Target 2** enables NPE-MMD and NPE-DANN to excel, achieving much lower NRMSE and ECE than NPE, NNPE, and even the analytic posterior.

Finally, PPD reliably detects NPE-UDA failures under prior misspecification. We suspect that its indifference to likelihood misspecifications is due to the structure of the simple Gaussian mean model. Nevertheless, SSDD reliably indicates NPE-UDA alignment failures, which translate to poor approximation performance in likelihood misspecification scenarios.

5.2 Experiment 2 - Bayesian Denoising: High-Dimensional Setting

Setup We base our high-dimensional benchmark on a noisy camera model, similar to Ramesh et al. (2022). The parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{256}$ represents the original image, whereas the observation $\boldsymbol{x} \in \mathbb{R}^{256}$ is a blurred version of the original image generated by the noisy camera. The training data set consists of $N = 50\,000$ images from the MNIST data set (Lecun et al., 1998), downscaled to 16×16 pixels for compatibility with the USPS data set (Hull, 1994). We test four different misspecification scenarios (see Table 2 for examples). In the prior misspecification scenario, we keep the settings of the noisy camera model constant but use images from the USPS data set (Hull, 1994). While both data sets contain digits, the USPS data set features smaller margins, giving the priors different support. In the likelihood scale scenario, we increase the amount of blur. In the noise contamination scenario, we replace 10% of the pixels with salt-and-pepper noise (i.e., set them to black or white). In the row contamination scenario, we randomly set two rows (12.5% of the pixels) of each observation to black. We evaluate the performance on $N_{\text{obs}} = 1,000$ observed data sets (seen by NPE-UDA methods during training).

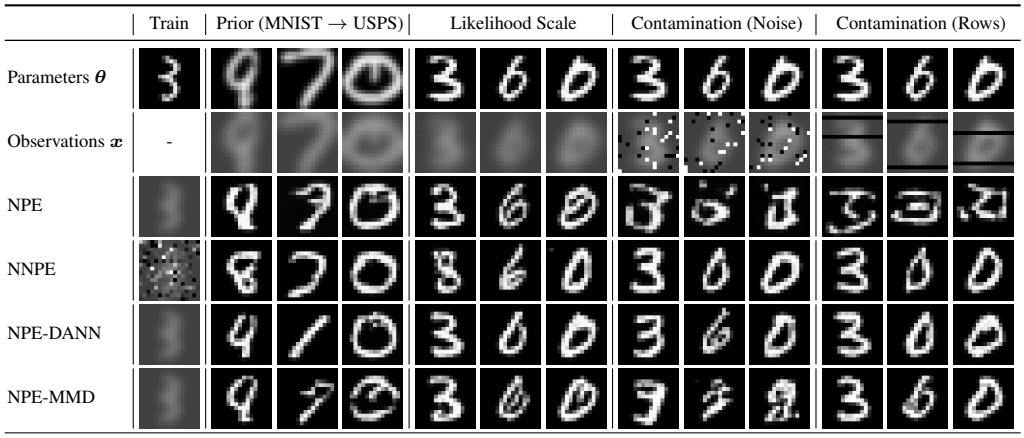


Table 2: **Experiment 2**. Parameters, observations, and samples from the run with the lowest NRMSE for each scenario and method. *Train* shows a sample from the parameters $\boldsymbol{\theta}$ of the training distribution and the corresponding observations $\boldsymbol{x}_{\text{obs}}$. The observations are identical for NPE, NPE-DANN, and NPE-MMD, whereas spike-and-slab noise is added for NNPE. The similarity to the observations in the *Contamination (Noise)* scenario explains the good performance of NNPE.

Results Table 1 displays an overview of the metrics in all scenarios. Table 2 shows samples from the best run (lowest NRMSE) for each scenario and method. We observe worse approximations for all robust methods compared to NPE in the prior misspecification scenario, even though the summary space domain distance (SSDD) is strongly diminished for NPE-DANN and NPE-MMD. This is somewhat expected, as performance improvements would also require an adaptation of the approximator, which cannot be induced by the methods tested here. NNPE is beneficial in the two contamination scenarios, whereas NPE-DANN and NPE-MMD improve performance in all three likelihood misspecification scenarios. The results highlight the differences between the robust methods: While NNPE mainly excels in the noise contamination scenario, where its misspecification model matches the domain shift, NPE-UDA methods effectively adapt to different likelihood shifts.

Overall, NPE-DANN achieves good performance over a wide range of λ values. In contrast, NPE-MMD is prone to overregularizing the summary space, leading to a complete loss of information in the summary space (see also Figure 1). This is indicated by a huge drop in performance and vanishing SSDD. We found NPE-MMD highly sensitive to the chosen batch size, which we had to increase from 32 to 128 to achieve acceptable results. Thus, increasing the batch size and reducing λ can counteract excessive regularization in higher-dimensional problems. Finally, the close correspondence between the NRMSE and PPD metrics confirms our hypothesis that the limited diagnostic power of PPD in the likelihood misspecification scenarios of Experiment 5.1 was caused by the limited informativeness of data simulated from a simple Gaussian model.

6 Conclusion

We argued that introducing UDA to NPE methods shifts the inference goal from the standard analytic posterior $p(\boldsymbol{\theta} \mid \boldsymbol{x}_{\text{obs}})$ to another posterior $p(\boldsymbol{\theta} \mid \tilde{\boldsymbol{x}}_{\text{obs}})$ based on adjusted data $\tilde{\boldsymbol{x}}_{\text{obs}}$. This implies potential robustness gains under likelihood misspecification, where ignoring unmodeled phenomena in the observed data can be desirable, but reduces the amount of information available to counteract prior misspecification. We consistently found these patterns throughout our systematic evaluations for both the existing NPE-MMD and a new NPE-DANN method. Whereas NPE-DANN was less stable than NPE-MMD in the low-dimensional benchmark, it excelled in the likelihood misspecification scenarios of the high-dimensional benchmark. Lastly, we confirmed the existence of an application-specific optimal amount of UDA regularization (Zellinger et al., 2021) in the NPE context. In light of our results, we propose a two-step approach for diagnosing NPE-UDA methods in real-world applications relative to an NPE baseline: (1) assessing summary space alignment via summary space domain distance and (2) evaluating whether this alignment improves fit to empirical data via posterior predictive distance.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers and program chairs of the Frontiers in Probabilistic Inference: Learning meets Sampling Workshop at ICLR 2025 for their constructive feedback and thoughtful suggestions. The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant INST 35/1597-1 FUGG. Additionally, LE was supported by a grant from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation; GRK 2277; project number 310365261) to the research training group Statistical Modeling in Psychology (SMiP). VP is supported by the Konrad Zuse School of Excellence in Learning and Intelligent Systems (ELIZA) through the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Education and Research. M.V.K. acknowledges funding by the German Research Foundation (DFG) through grant KR 6065/1-1. PB acknowledges support of DFG Project 528702768 and DFG Collaborative Research Center 391 (Spatio-Temporal Statistics for the Transition of Energy and Transport) – 520388526. STR is supported by NSF under Grant No. 2448380.

REFERENCES

Lynton Ardizzone, Jakob Kruse, Sebastian Wirkert, Daniel Rahner, Eric W Pellegrini, Ralf S Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. *arXiv preprint arXiv:1808.04730*, 2018.

Lynton Ardizzone, Jakob Kruse, Carsten Lüth, Niels Bracher, Carsten Rother, and Ullrich Köthe. Conditional invertible neural networks for diverse image-to-image translation. In *Pattern Recognition: 42nd DAGM German Conference, DAGM GCPR 2020, Tübingen, Germany, September 28–October 1, 2020, Proceedings 42*, pp. 373–387. Springer, 2021.

Grace Avecilla, Julie N Chuong, Fangfei Li, Gavin Sherlock, David Gresham, and Yoav Ram. Neural networks enable efficient and accurate simulation-based inference of evolutionary parameters from adaptation dynamics. *PLoS Biology*, 20(5):e3001633, 2022.

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. *Machine learning*, 79:151–175, 2010.

Yves Bernaerts, Michael Deistler, Pedro J Gonçalves, Jonas Beck, Marcel Stimberg, Federico Scala, Andreas S Tolias, Jakob Macke, Dmitry Kobak, and Philipp Berens. Combined statistical-mechanistic modeling links ion channel genes to physiology of cortical neuron types. *bioRxiv*, pp. 2023–03, 2023.

Michael Betancourt. Calibrating model-based inferences and decisions. *arXiv preprint arXiv:1803.08393*, 2018.

Sebastian Bischoff, Alana Darcher, Michael Deistler, Richard Gao, Franziska Gerken, Manuel Gloeckler, Lisa Haxel, Jaivardhan Kapoor, Janne K Lappalainen, Jakob H Macke, et al. A practical guide to statistical distances for evaluating generative models in science. *arXiv preprint arXiv:2403.12636*, 2024.

Pier Giovanni Bissiri, Chris C Holmes, and Stephen G Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130, 2016.

Paul-Christian Bürkner, Maximilian Scholz, and Stefan T Radev. Some models are useful, but how do we know which ones? towards a unified bayesian model taxonomy. *Statistic Surveys*, 17: 216–310, 2023.

Patrick Cannon, Daniel Ward, and Sebastian M Schmon. Investigating the impact of model misspecification in neural simulation-based inference. *arXiv preprint arXiv:2209.01845*, 2022.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

Jeffrey Chan, Valerio Perrone, Jeffrey Spence, Paul Jenkins, Sara Mathieson, and Yun Song. A likelihood-free inference framework for population genetic data using exchangeable neural networks. *Advances in neural information processing systems*, 31, 2018.

A Ćiprijanović, Diana Kafkes, S Jenkins, K Downey, Gabriel N Perdue, Sandeep Madireddy, T Johnston, and Brian Nord. Domain adaptation techniques for improved cross-domain study of galaxy mergers. *arXiv preprint arXiv:2011.03591*, 2020.

Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, January 2014. ISSN 0304-3975. doi: 10.1016/j.tcs.2013.09.027.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

Lars Dingeldein, Pilar Cossio, and Roberto Covino. Simulation-based inference of single-molecule experiments, 2024. URL https://arxiv.org/abs/2410.15896.

Lasse Elsemüller, Hans Olischläger, Marvin Schmitt, Paul-Christian Bürkner, Ullrich Köthe, and Stefan T Radev. Sensitivity-aware amortized bayesian inference. *Transactions on Machine Learning Research (TMLR)*, 2024.

David T. Frazier, Ryan Kelly, Christopher Drovandi, and David J. Warne. The statistical accuracy of neural posterior and likelihood estimation, 2024. URL https://arxiv.org/abs/2411.12068.

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

Richard Gao, Michael Deistler, and Jakob H Macke. Generalized bayesian inference for scientific simulators via amortized cost estimation. *Advances in Neural Information Processing Systems*, 36:80191–80219, 2023.

Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014.

Manuel Glöckler, Michael Deistler, and Jakob H Macke. Variational methods for simulation-based inference. *arXiv preprint arXiv:2203.04176*, 2022.

Manuel Glöckler, Michael Deistler, and Jakob H Macke. Adversarial robustness of amortized bayesian inference. *arXiv preprint arXiv:2305.14984*, 2023.

Manuel Gloeckler, Michael Deistler, Christian Weilbach, Frank Wood, and Jakob H Macke. All-in-one simulation-based inference. *arXiv preprint arXiv:2404.09636*, 2024.

A Gretton, K. Borgwardt, Malte Rasch, Bernhard Schölkopf, and AJ Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13:723–773, 03 2012.

Joeri Hermans, Volodimir Begy, and Gilles Louppe. Likelihood-free mcmc with amortized approximate ratio estimators. In *International conference on machine learning*, pp. 4239–4248. PMLR, 2020.

Daolang Huang, Ayush Bharti, Amauri Souza, Luigi Acerbi, and Samuel Kaski. Learning robust statistics for simulation-based inference under model misspecification. *Advances in Neural Information Processing Systems*, 36, 2023.

Peter J Huber. *Robust statistics*. John Wiley & Sons, 1981.

J.J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994. doi: 10.1109/34.291440.

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.

Ryan P Kelly, David J Nott, David T Frazier, David J Warne, and Chris Drovandi. Misspecification-robust sequential neural likelihood. *arXiv preprint arXiv:2301.13368*, 2023.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xiaofeng Liu, Chaehwa Yoo, Fangxu Xing, Hyejin Oh, Georges El Fakhri, Je-Won Kang, Jonghye Woo, et al. Deep unsupervised domain adaptation: A review of recent advances and perspectives. *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. Unsupervised domain adaptation: A reality check. *arXiv preprint arXiv:2111.15672*, 2021.

Costantino Pacilio, Swetha Bhagwat, and Roberto Cotesta. Simulation-based inference of black hole ringdowns in the time domain. *Physical Review D*, 110(8):083010, 2024.

Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE transactions on neural networks*, 22(2):199–210, 2010.

George Papamakarios and Iain Murray. Fast ε-free inference of simulation models with bayesian conditional density estimation. *Advances in neural information processing systems*, 29, 2016.

Christian S Perone, Pedro Ballester, Rodrigo C Barros, and Julien Cohen-Adad. Unsupervised domain adaptation for medical imaging segmentation with self-ensembling. *NeuroImage*, 194: 1–11, 2019.

Stefan T Radev, Ulf K Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE transactions on neural networks and learning systems*, 2020.

Stefan T Radev, Marvin Schmitt, Lukas Schumacher, Lasse Elsemüller, Valentin Pratz, Yannik Schälte, Ullrich Köthe, and Paul-Christian Bürkner. Bayesflow: Amortized bayesian workflows with neural networks. *arXiv preprint arXiv:2306.16015*, 2023.

Tom Rainforth, Adam Foster, Desi R Ivanova, and Freddie Bickford Smith. Modern bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024.

Poornima Ramesh, Jan-Matthis Lueckmann, Jan Boelts, Álvaro Tejero-Cantero, David S. Greenberg, Pedro J. Goncalves, and Jakob H. Macke. GATSBI: Generative adversarial training for simulation-based inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=kR1hC6j48Tp.

Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. A survey on domain adaptation theory: Learning bounds and theoretical guarantees, July 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Neil Savage. Synthetic data could be better than real data. *Nature*, 2023.

Marvin Schmitt, Paul-Christian Bürkner, and Köthe. Detecting model misspecification in amortized bayesian inference with neural networks. *Proceedings of the German Conference on Pattern Recognition (GCPR)*, 2023.

Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. *Advances in neural information processing systems*, 29, 2016.

Ali Siahkoohi, Gabrio Rizzuti, Rafael Orozco, and Felix J Herrmann. Reliable amortized variational inference with physics-based latent distribution correction. *Geophysics*, 88(3):R297–R322, 2023.

Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.

Paxson Swierc, Megan Zhao, Aleksandra Ćiprijanović, and Brian Nord. Domain adaptation for measurements of strong gravitational lenses. *arXiv preprint arXiv:2311.17238*, 2023.

Paxson Swierc, Marcos Tamargo-Arizmendi, Aleksandra Ćiprijanović, and Brian D Nord. Domain-adaptive neural posterior estimation for strong gravitational lens analysis. *arXiv preprint arXiv:2410.16347*, 2024.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Rheeya Uppaal, Yixuan Li, and Junjie Hu. How useful is continued pre-training for generative unsupervised domain adaptation? In *Proceedings of the 9th Workshop on Representation Learning for NLP (RepL4NLP-2024)*, pp. 99–117, 2024.

Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014. ISSN 2167-8359. doi: 10.7717/peerj.453. URL https://doi.org/10.7717/peerj.453.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Xiaoyu Wang, Ryan P. Kelly, David J Warne, and Christopher Drovandi. Preconditioned neural posterior estimation for likelihood-free inference. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=vgIBAOkIhY.

Daniel Ward, Patrick Cannon, Mark Beaumont, Matteo Fasiolo, and Sebastian Schmon. Robust neural posterior estimation and statistical model criticism. *Advances in Neural Information Processing Systems*, 35:33845–33859, 2022.

Antoine Wehenkel, Juan L Gamella, Ozan Sener, Jens Behrmann, Guillermo Sapiro, Marco Cuturi, and Jörn-Henrik Jacobsen. Addressing misspecification in simulation-based inference through data-driven calibration. *arXiv preprint arXiv:2405.08719*, 2024.

Jonas Bernhard Wildberger, Maximilian Dax, Simon Buchholz, Stephen R Green, Jakob H. Macke, and Bernhard Schölkopf. Flow matching for scalable simulation-based inference. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=D2cS6SoYlP.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Andrew Zammit-Mangion, Matthew Sainsbury-Dale, and Raphaël Huser. Neural methods for amortized inference. *Annual Review of Statistics and Its Application*, 12, 2024.

Werner Zellinger, Natalia Shepeleva, Marius-Constantin Dinu, Hamid Eghbal-zadeh, Hoan Duc Nguyen, Bernhard Nessler, Sergei Pereverzyev, and Bernhard A Moser. The balancing principle for parameter choice in distance-regularized domain adaptation. *Advances in Neural Information Processing Systems*, 34:20798–20811, 2021.

YiFan Zhang, Xue Wang, Jian Liang, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. Free lunch for domain adversarial training: Environment label smoothing. *arXiv preprint arXiv:2302.00194*, 2023.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2022.

APPENDIX

A THEORETICAL DETAILS

A.1 DEFINING AMORTIZED BAYESIAN INFERENCE

The term "amortized" has been used inconsistently throughout the literature, often denoting different generalization scopes. To clarify this concept for the discussion within this work, we offer the following definition:

Definition 1. *Let $\mathcal{A}$ denote a learner, $\boldsymbol{y}$ denote target variables, $\boldsymbol{x}$ represent input data, and $\boldsymbol{c}$ denote context variables. A learner $\boldsymbol{y} \sim \mathcal{A}(\boldsymbol{x}, \boldsymbol{c})$ is an* amortized Bayesian approximator *of a target quantity $\boldsymbol{y}$ with respect to a joint distribution $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{c})$ if it can directly approximate $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{c})$ for any $(\boldsymbol{x}, \boldsymbol{c}) \sim p(\boldsymbol{x}, \boldsymbol{c})$ without requiring further training or additional approximation algorithms.*

By this definition, sequential methods that necessitate further training for new data (Papamakarios & Murray, 2016; Glöckler et al., 2022) are not considered amortized. Similarly, neural likelihood estimation (NLE; Papamakarios & Murray, 2016) and neural ratio estimation (NRE) (Hermans et al., 2020) which depend on MCMC algorithms do not qualify as amortized. In contrast, recent transformer-based (Gloeckler et al., 2024) or context-aware methods (Elsemüller et al., 2024) clearly fall within the scope of amortized neural posterior estimation (NPE).

B EXPERIMENTAL DETAILS

Since the analytic posterior is only obtainable in Experiment 5.1, we measure performance relative to the data-generating parameters $\boldsymbol{\theta}^*$ to enable a direct comparison between the experiments. For likelihood misspecification settings, $\boldsymbol{\theta}^*$ is closely related to the posterior $p(\boldsymbol{\theta} \mid \tilde{\boldsymbol{x}}_{\text{obs}})$ based on adjusted (e.g., decontaminated) data $\tilde{\boldsymbol{x}}_{\text{obs}}$ (**Target 2**). Thus, the NPE-UDA posterior approximations being closer to $\boldsymbol{\theta}^*$ than the analytic posterior $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ in the contamination scenario of Experiment 5.1 indicates that NPE-UDA methods indeed focus **Target 2**.

In all experiments, we build upon the `BayesFlow` Python library for amortized Bayesian workflows using generative neural networks (Radev et al., 2023).

B.1 METHOD DETAILS

NNPE We implemented NNPE following the original implementation of Ward et al. (2022) at `https://github.com/danielward27/rnpe`, who used a spike scale of $\sigma = 0.01$ and a slab scale of $\tau = 0.25$ for all experiments. Whether spike (standard normal) or slab (standard Cauchy) noise is applied to a simulated data point is determined by sampling from a Bernoulli distribution with $p = 0.5$.

Sensitivities of NPE-UDA In both experiments, we found the typical UDA phenomenon of sensitivity to higher learning rates (Perone et al., 2019) in the form of unstable learning dynamics such as exploding gradients. We also found sensitivity to short training times, suggesting that finding a stable optimum for the two-component NPE-UDA loss in Eq. 3 requires more gradient updates than usual.

Computational Cost of NPE-UDA Since the NPE-UDA methods operate in the compressed summary space, the runtime increase during training is minimal compared to NPE. For example, despite the relatively large (32-dimensional) summary space in Experiment 5.2, NPE and NPE-MMD took 12s/epoch and NPE-DANN 13s/epoch during GPU training on a cluster.

B.2 METRICS

We compute multiple metrics that measure the performance based on the approximation performance of J data-generating parameters $\{\theta_j^*\}_{j=1}^J$ via S posterior samples (we forego the obs notation where possible for brevity here). Depending on the metric, results are averaged across the J parameters and/or N observed data sets.

Normalized root mean squared error (NRMSE):

$$\text{NRMSE} = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{J}\sum_{j=1}^{J}\frac{\sqrt{\frac{1}{S}\sum_{s=1}^{S}(\theta^*_{j,n} - \hat{\theta}^{(s)}_{j,n})^2}}{\max(\theta^*_j) - \min(\theta^*_j)}\right]. \tag{8}$$

Expected calibration error (ECE) via the fraction of ground-truth inliers for R linearly spaced α-confidence intervals in $[0.005, 0.995]$ (Ardizzone et al., 2018; Radev et al., 2020):

$$\text{ECE} = \frac{1}{J}\sum_{j=1}^{J}\text{median}_{r=1}^{R}\left(\left|\frac{1}{N}\sum_{n=1}^{N}\mathbb{I}\left\{Q_{\frac{1-\alpha_r}{2}}(\hat{\theta}^{(n)}_j) \leq \theta^*_j \leq Q_{1-\frac{1-\alpha_r}{2}}(\hat{\theta}^{(n)}_j)\right\} - \alpha_r\right|\right), \tag{9}$$

where $\text{median}_{r=1}^{R}$ represents the median fraction of inliers across the $R = 20$ credible intervals and $Q_k(\hat{\theta}^{(n)}_j)$ represents the k-th quantile of the posterior samples for the n-th data set. We estimate the ECE on all test data sets via the median calibration error of $R = 20$ linearly spaced credible intervals, averaged across J model parameters.

Posterior contraction (PC) relative to the prior distribution (Betancourt, 2018):

$$\text{PC} = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{J}\sum_{j=1}^{J}\left(1 - \frac{\text{Var}(\hat{\theta}^{(s)}_{j,n})}{\text{Var}(\theta^*_{j,n})}\right)\right]. \tag{10}$$

Posterior predictive distance (PPD):

$$\text{PPD} = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{S}\sum_{s=1}^{S} d\left(\boldsymbol{x}_n, \hat{\boldsymbol{x}}^{(s)}_n\right)\right]. \tag{11}$$

where $\hat{\boldsymbol{x}}^{(s)}$ represents a re-simulation based on a posterior sample of all parameters, $\hat{\boldsymbol{\theta}}^{(s)}$, and we use the MMD (Experiment 5.1) or NRMSE (Experiment 5.2) for $d(\cdot, \cdot)$.

The summary space domain distance (SSDD), which does not measure approximation performance but the degree of summary space alignment, is based on the biased sample-based $\widehat{\text{MMD}}^2$ estimator (Gretton et al., 2012):

$$\text{SSDD} = \frac{1}{N}\sum_{n=1}^{N}\widehat{\text{MMD}}^2\left[\{\phi(\boldsymbol{x}_n)\} \,||\, \{\phi(\boldsymbol{x}^{\text{obs}}_n)\}\right], \tag{12}$$

where $\{\phi(\boldsymbol{x}_n)\}$ and $\{\phi(\boldsymbol{x}^{\text{obs}}_n)\}$ are sets of summary statistics over which the expectations are approximated.

B.3 Experiment 1 - 2D Gaussian Means

Misspecification Setting	**Prior**	**Likelihood**
Well-specified	$\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_2)$	$\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}_2)$
Prior location misspecification	$\boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\mu_0}, \boldsymbol{I}_2)$	$\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}_2)$
Prior scale misspecification	$\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}, \tau_0\boldsymbol{I}_2)$	$\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}_2)$
Likelihood scale misspecification	$\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_2)$	$\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{\mu}, \tau\boldsymbol{I}_2)$
Contamination misspecification	$\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_2)$	$\boldsymbol{x}_k \sim \frac{\epsilon}{2}\cdot\delta(\boldsymbol{x}-\boldsymbol{c}) + \frac{\epsilon}{2}\cdot\delta(\boldsymbol{x}+\boldsymbol{c}) + (1-\epsilon)\cdot\mathcal{N}(\mu, \boldsymbol{I}_2)$

Table B.1: **Experiment 1**: Overview of the model specifications in the different misspecification settings.

Table B.1 provides an overview of the well-specified setting and the different misspecification scenarios inspired by Schmitt et al. (2023)

B.3.1 Network Architecture

We use a deep set architecture (Zaheer et al., 2017) for the summary network ϕ, compressing the input to 4-dimensional summary statistics. For the generative inference network of the approximator q, we use an affine coupling flow architecture (Ardizzone et al., 2021; Kingma & Dhariwal, 2018) with 3 coupling layers.

For the domain classifier ψ in NPE-DANN, we use a standard feedforward network with 2 hidden layers of width 256. We do not use label smoothing or weight the gradient reversal balance.

B.3.2 Training and Evaluation Details

To rule out any overfitting effects, we use an online training approach where new data from the simulated and the observed domain is simulated at each training step, resulting in overall simulation budgets of $N = 49,920$ and $N_{\text{obs}} = 49,920$. Since we use a batch size of 32, also for the observed data in NPE-UDA methods, online training amounts to 1560 mini-batches and thus gradient updates. We use an Adam optimizer with an initial learning rate of $5 \cdot 10^{-4}$ and cosine decay.

We use $S = 100$ posterior samples per method to limit the computational cost of the PPD calculation, where an MMD distance is calculated for each re-simulated data set and thus posterior sample. While we did not observe different result patterns with higher values of S, we will increase S in the full version of this work.

B.3.3 Additional Results

We provide additional results iterating over three factors: (i) performance in the simulated vs. the observed domain, (ii) $\lambda = [0.1, 1, 10]$, and (iii) comparison of the posterior approximations to the analytic posterior instead of the data-generating parameters $\boldsymbol{\theta}^*$.

Performance in the Simulated Domain Figure B1, Figure B2, and Figure B3 show the performance in the simulated domain. Despite notable performance differences in the observed domain, all methods perform well in the simulated domain for the vast majority of settings, with the only exception being the failures of NPE-DANN for high regularization weights in Figure B3. NNPE performs worse in the simulated (noiseless) domain since it was optimized based on noisy training data. Besides the NPE-DANN failures, we mostly do not observe a trade-off of the summary space alignment of the NPE-UDA methods. Only in the high regularization setting $\lambda = 10$, the ECE is systematically higher compared to NPE.

Performance in the Observed Domain Figure B4 and Figure B5 confirm our finding of an application- and also method-specific λ optimum: Whereas the difference of the NPE-UDA methods to NPE is often small for $\lambda = 0.1$, $\lambda = 10$ still leads to performance improvements of NPE-MMD in likelihood misspecification scenarios but renders NPE-DANN highly unstable when large domain shifts are present.

Performance Compared to the Analytic Posterior Figure B6, Figure B7, and Figure B8 compare the different approximation algorithms to the analytic posterior under **Target 1**, also showing the clear separation between NPE and NNPE vs. the NPE-UDA methods as a function of λ. The inference network latent distance (INLD) to its base distribution, a proxy of approximation quality (Siahkoohi et al., 2023), is closely related to a methods performance (compare for example Figure 2 and Figure B7).

B.4 Experiment 2

Simulator (Noisy Camera Model) We adopt a noisy camera model similar to the one presented in Ramesh et al. (2022). First, the input image is clipped to the range $[-1, 1]$. Next, we use scikit-image (van der Walt et al., 2014) to add Poisson noise to the image, then filter it using a Gaussian filter from SciPy (Virtanen et al., 2020) with a standard deviation σ for the Gaussian kernel. The result is a blurred image with identical size as the input image.

Data Preparation For each data set, we normalize the images to the range $[-1, 1]$. The MNIST (Lecun et al., 1998) images are rescaled from 28×28 to 16×16 with anti-aliasing enabled. To produce the training data $\boldsymbol{x}$, the images are processed by the simulator, with $\sigma_0 = 1.4$. For NNPE, we then add noise to $\boldsymbol{x}$ using the spike-and-slab error model from Ward et al. (2022).

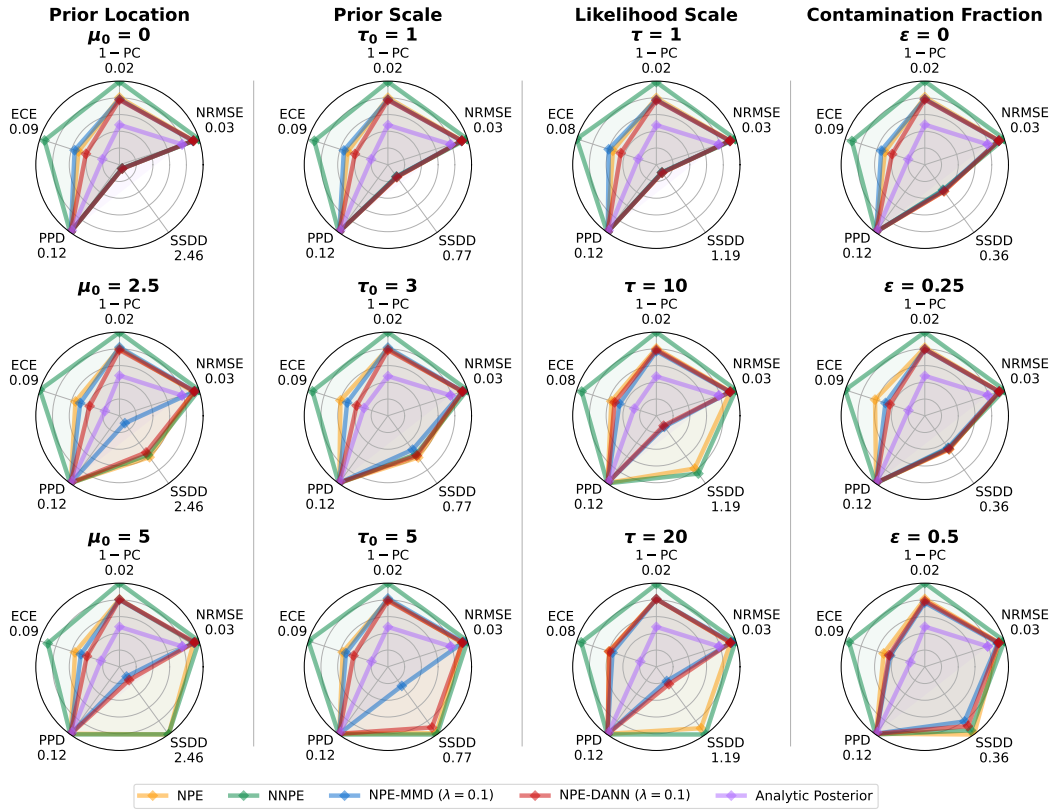


Figure B1: **Experiment 1**: Performance metrics of the methods in all misspecification scenarios (columns) **on simulated (i.e., well-specified) data for $\lambda = 0.1$ in NPE-MMD and NPE-DANN**, averaged across 3 separate runs. Lower values indicate better performance (for SSDD only for NPE-MMD and NPE-DANN). $1-$ PC $= 1-$ Posterior Contraction. NRMSE = Normalized Root Mean Squared Error. SSDD = Summary Space Domain Distance (MMD; not applicable for Analytic Posterior). PPD = Posterior Predictive Distance (MMD). ECE = Expected Calibration Error.

While the training data remains constant across scenarios, the observed data is generated in different ways. For the prior misspecification scenario, we use the USPS data set (Hull, 1994) instead of MNIST, but the parameters of the simulator remain identical (i.e., $\sigma = \sigma_0$). For the likelihood scale scenario, we use $\tilde{\sigma} = 1.25 \cdot \sigma_0$, leading to an increased blur. For the noise contamination scenario, we randomly set 10% of the pixels of each observation to black or white. For the row contamination scenario, we randomly set 2 rows of each observation (i.e., 12.5% of the pixels) to black. Refer to Table 2 for samples from each scenario.

Network Architecture For the summary network, we use a 4-layer convolutional neural network, which outputs 32 learned summary variables.

For the inference network, we use flow matching (Lipman et al., 2023; Wildberger et al., 2023) to convert a multivariate Gaussian distribution to the approximate posterior distribution. We use a U-Net architecture (Ronneberger et al., 2015) to learn the flow field conditional on the summary variables.

For NPE-DANN, we use a domain classifier ψ consisting of a standard feedforward network with 3 hidden layers of width 256, a gradient reversal layer (GRL) weight of 1, and no label smoothing.

Training and Evaluation Details We use an AdamW optimizer with an initial learning rate of $5 \cdot 10^{-4}$ and cosine decay. We use a batch size of 32 and train for 20 epochs, except for NPE-MMD, which required increasing the batch size to 128. To keep the number of gradient updates constant, we also increased the number of epochs to 80 for NPE-MMD. The training budget is $50\,000$ training images, and $1\,000$ observed images. Training one neural network takes approximately 10 minutes on a GPU.

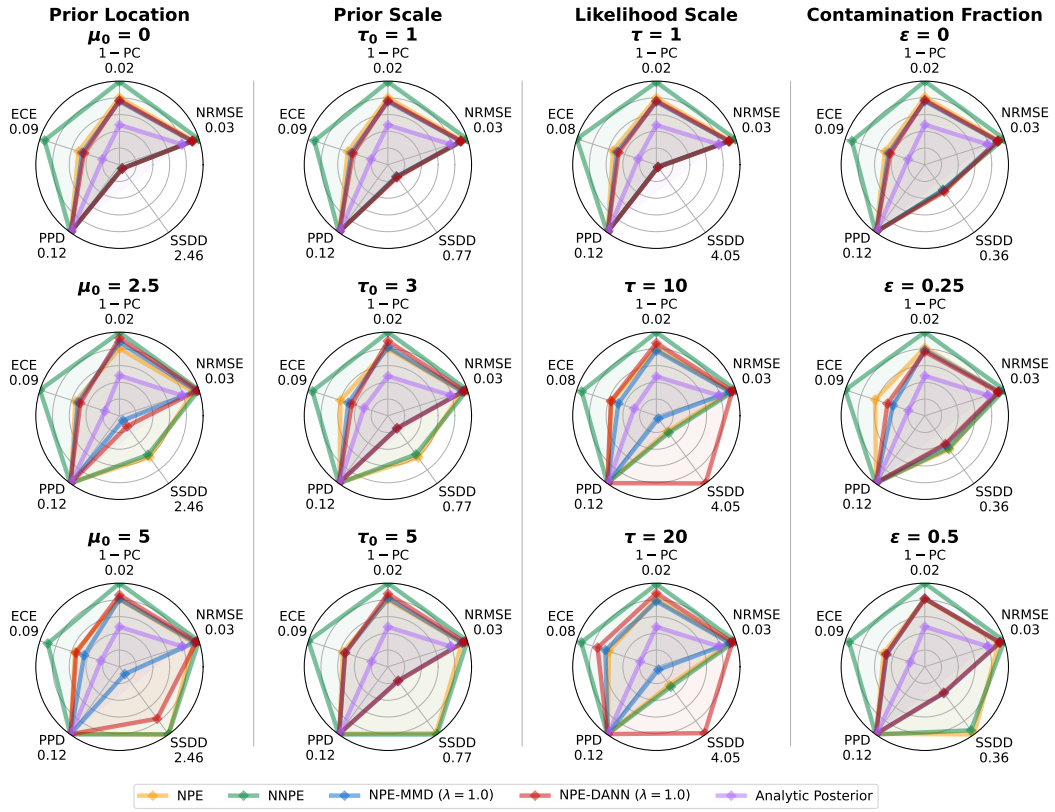


Figure B2: **Experiment 1**: Performance metrics of the methods in all misspecification scenarios (columns) **on simulated (i.e., well-specified) data for $\lambda = 1$ in NPE-MMD and NPE-DANN**, averaged across 3 separate runs. Lower values indicate better performance (for SSDD only for NPE-MMD and NPE-DANN). $1-$ PC $= 1-$ Posterior Contraction. NRMSE = Normalized Root Mean Squared Error. SSDD = Summary Space Domain Distance (MMD; not applicable for Analytic Posterior). PPD = Posterior Predictive Distance (MMD). ECE = Expected Calibration Error.

Similar to Experiment 5.1, we use a relatively low number of posterior samples (here: $S = 10$) per method to limit the computational cost of the experiment, allowing for a broader exploration of hyperparameters and the variance between multiple runs. While we observe a low variance of posterior samples and additionally average over observations and samples, we will increase S in the full version of this work.

Additional Metrics Table B.2 displays the performance on a held-out in-distribution data set, to assess the influence on the loss on the in-domain observations. Table B.3 displays the same data as Table 1, but with uncertainty indicators (standard deviation).

Additional Figures Figure B9 shows the plots corresponding to Figure 1 for the remaining three scenarios.

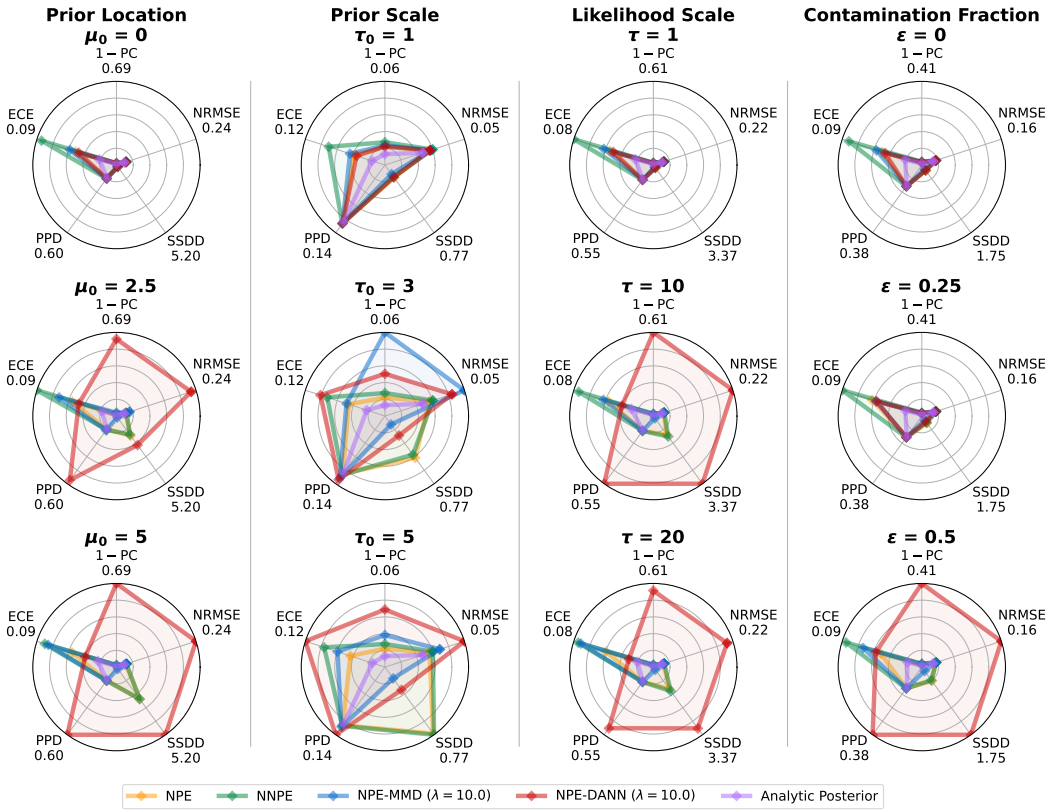


Figure B3: **Experiment 1**: Performance metrics of the methods in all misspecification scenarios (columns) **on simulated (i.e., well-specified) data for $\lambda = 10$ in NPE-MMD and NPE-DANN**, averaged across 3 separate runs. Lower values indicate better performance (for SSDD only for NPE-MMD and NPE-DANN). $1-$ PC $= 1-$ Posterior Contraction. NRMSE = Normalized Root Mean Squared Error. SSDD = Summary Space Domain Distance (MMD; not applicable for Analytic Posterior). PPD = Posterior Predictive Distance (MMD). ECE = Expected Calibration Error.

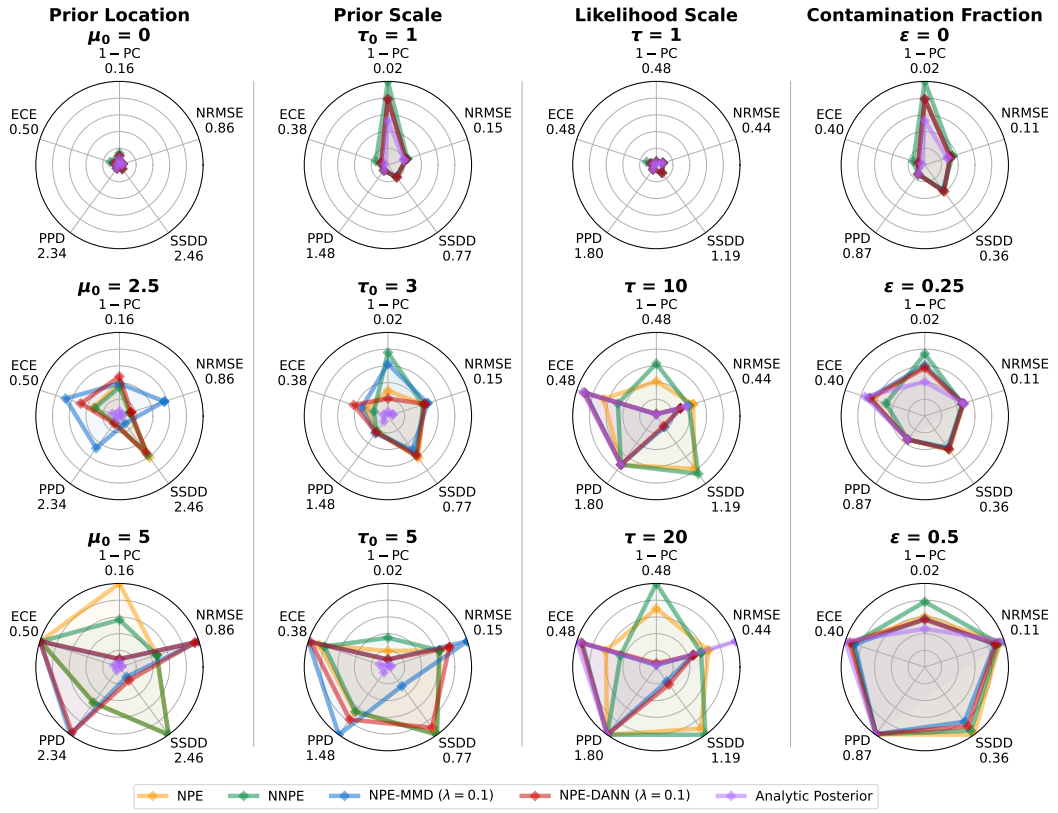


Figure B4: **Experiment 1**: Performance metrics of the methods in all misspecification scenarios (columns) **for $\lambda = 0.1$ in NPE-MMD and NPE-DANN**, averaged across 3 separate runs. Lower values indicate better performance (for SSDD only for NPE-MMD and NPE-DANN). $1-$ PC $=$ $1-$ Posterior Contraction. NRMSE = Normalized Root Mean Squared Error. SSDD = Summary Space Domain Distance (MMD; not applicable for Analytic Posterior). PPD = Posterior Predictive Distance (MMD). ECE = Expected Calibration Error.

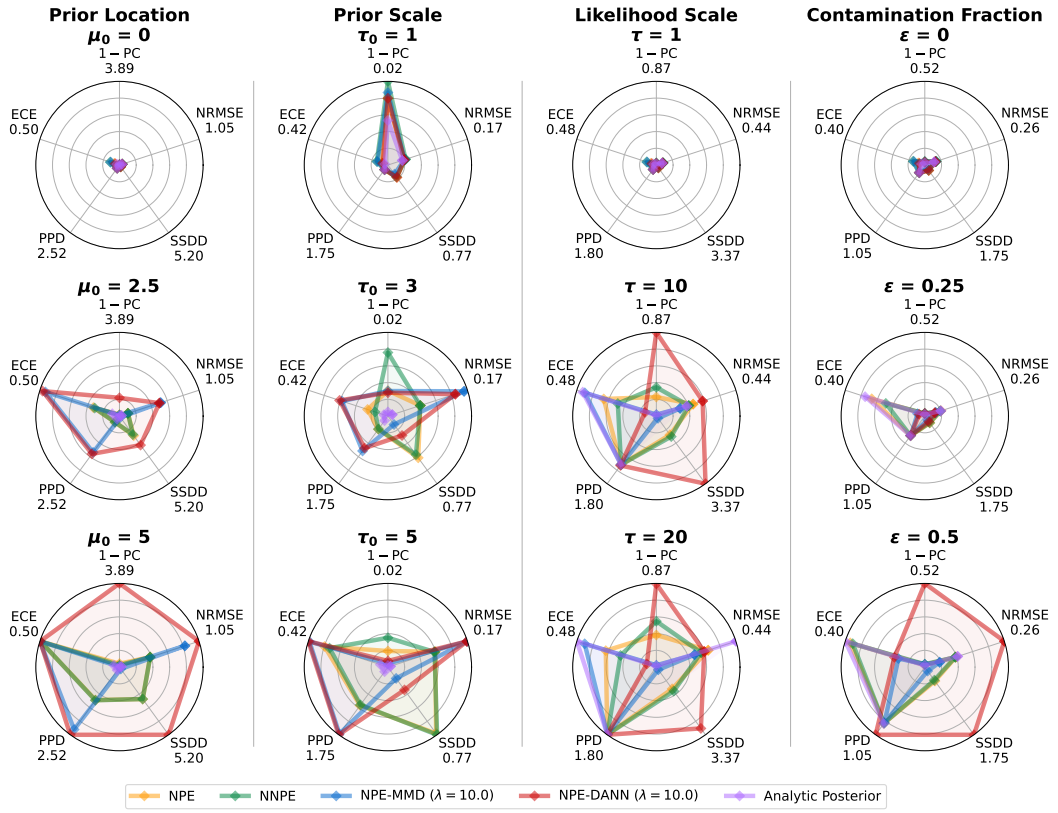


Figure B5: **Experiment 1**: Performance metrics of the methods in all misspecification scenarios (columns) **for $\lambda = 10$ in NPE-MMD and NPE-DANN**, averaged across 3 separate runs. Lower values indicate better performance (for SSDD only for NPE-MMD and NPE-DANN). $1-$ PC $=$ $1-$ Posterior Contraction. NRMSE = Normalized Root Mean Squared Error. SSDD = Summary Space Domain Distance (MMD; not applicable for Analytic Posterior). PPD = Posterior Predictive Distance (MMD). ECE = Expected Calibration Error.

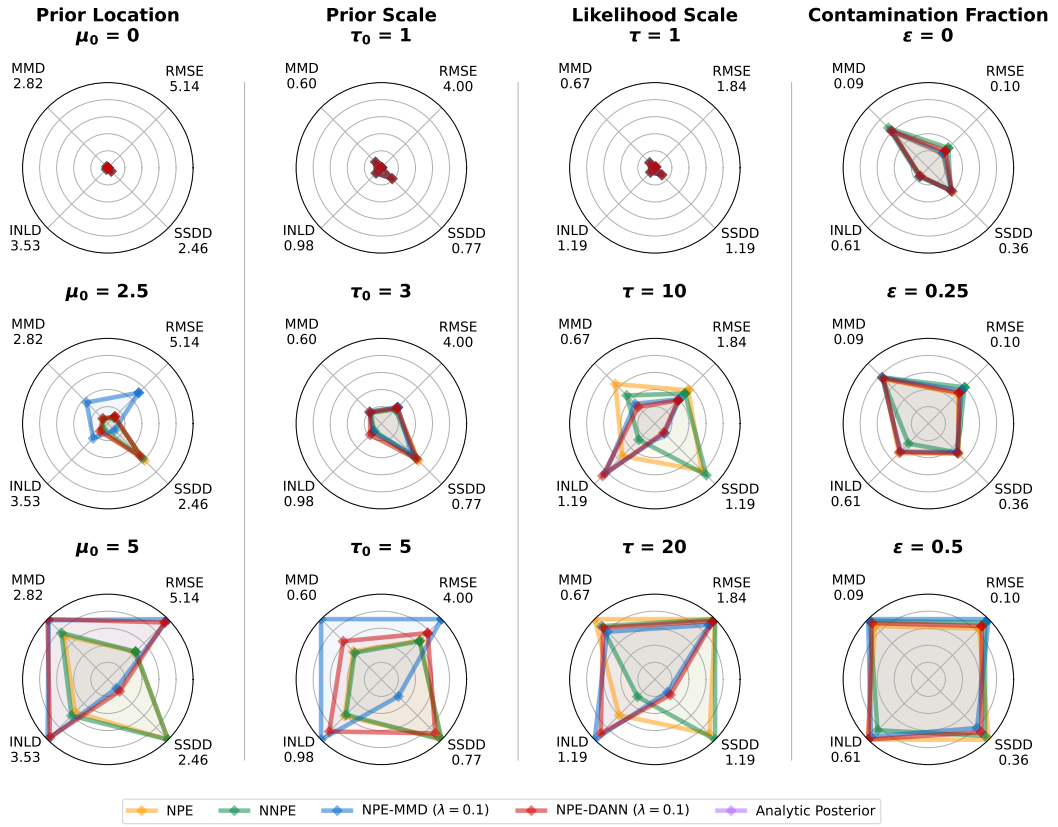


Figure B6: **Experiment 1**: Performance metrics of the methods in all misspecification scenarios (columns) **compared to the analytic posterior for $\lambda = 0.1$ in NPE-MMD and NPE-DANN**, averaged across 3 separate runs. Lower values indicate better performance (for SSDD only for NPE-MMD and NPE-DANN). MMD = Maximum Mean Discrepancy to analytic posterior. RMSE = Root Mean Squared Error to analytic posterior. SSDD = Summary Space Domain Distance (MMD; not applicable for Analytic Posterior). INLD = Inference Network Latent Distance (MMD) to base distribution.

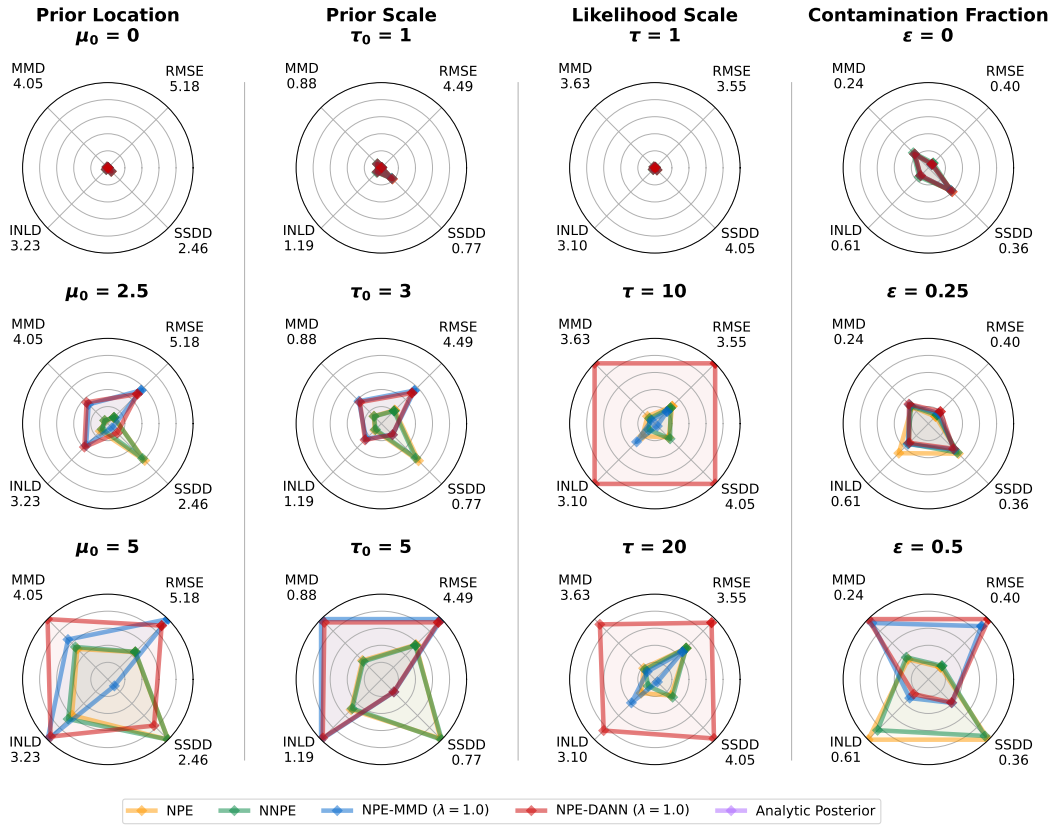


Figure B7: **Experiment 1**: Performance metrics of the methods in all misspecification scenarios (columns) **compared to the analytic posterior for $\lambda = 1$ in NPE-MMD and NPE-DANN**, averaged across 3 separate runs. Lower values indicate better performance (for SSDD only for NPE-MMD and NPE-DANN). MMD = Maximum Mean Discrepancy to analytic posterior. RMSE = Root Mean Squared Error to analytic posterior. SSDD = Summary Space Domain Distance (MMD; not applicable for Analytic Posterior). INLD = Inference Network Latent Distance (MMD) to base distribution.

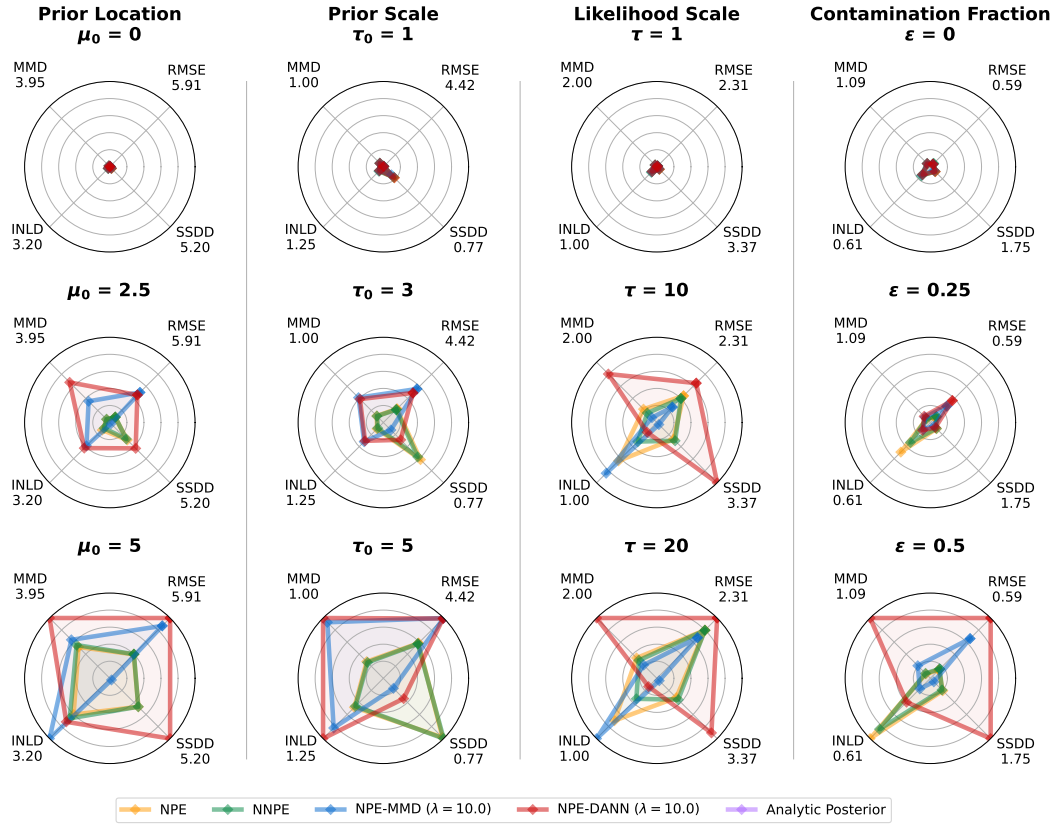
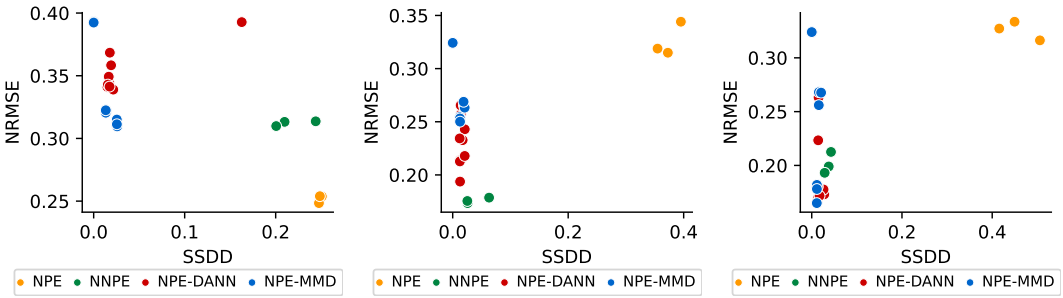


Figure B8: **Experiment 1**: Performance metrics of the methods in all misspecification scenarios (columns) **compared to the analytic posterior for $\lambda = 10$ in NPE-MMD and NPE-DANN**, averaged across 3 separate runs. Lower values indicate better performance (for SSDD only for NPE-MMD and NPE-DANN). MMD = Maximum Mean Discrepancy to analytic posterior. RMSE = Root Mean Squared Error to analytic posterior. SSDD = Summary Space Domain Distance (MMD; not applicable for Analytic Posterior). INLD = Inference Network Latent Distance (MMD) to base distribution.



(a) Prior (MNIST → USPS) (b) Contamination (Noise) (c) Contamination (Rows)

Figure B9: **Experiment 2**: Relationship of summary space domain distance (SSDD) and normalized root mean squared error (NRMSE, lower is better). For a) we see that despite the reduced SSDD, there is no gain in performance. For b) and c), we observe a sweet spot at a low SSDD value, before performance drops again when approaching zero. Refer to Table 1 for numerical values.

Method	λ	Prior (MNIST → USPS) NRMSE ↓	Prior (MNIST → USPS) PPD ↓	Likelihood Scale NRMSE ↓	Likelihood Scale PPD ↓
NPE	-	0.076 (9.3e-04)	0.009 (8.4e-05)	0.077 (4.4e-04)	0.009 (1.1e-04)
NNPE	-	0.178 (2.0e-03)	0.028 (9.6e-04)	0.179 (2.5e-03)	0.027 (4.0e-04)
NPE-DANN	0.01	0.092 (3.2e-03)	0.011 (6.4e-04)	0.081 (1.0e-04)	0.009 (1.2e-04)
NPE-DANN	0.10	0.144 (1.3e-02)	0.020 (2.1e-03)	0.096 (2.2e-03)	0.012 (2.8e-04)
NPE-DANN	1.00	0.286 (2.9e-02)	0.064 (1.6e-02)	0.126 (5.8e-03)	0.016 (8.1e-04)
NPE-MMD	0.01	**0.064 (1.1e-03)**	**0.007 (7.4e-05)**	**0.064 (3.7e-04)**	**0.007 (5.3e-05)**
NPE-MMD	0.10	0.065 (1.1e-03)	0.008 (1.1e-04)	0.324 (4.8e-04)	0.085 (1.8e-04)
NPE-MMD	1.00	0.324 (5.1e-04)	0.085 (1.6e-04)	0.324 (3.1e-04)	0.085 (1.1e-04)

Method	λ	Contamination (Noise) NRMSE ↓	Contamination (Noise) PPD ↓	Contamination (Rows) NRMSE ↓	Contamination (Rows) PPD ↓
NPE	-	0.077 (4.0e-04)	0.009 (6.4e-05)	0.076 (5.6e-04)	0.009 (1.5e-04)
NNPE	-	0.179 (4.0e-03)	0.027 (1.2e-03)	0.176 (3.5e-03)	0.027 (1.6e-03)
NPE-DANN	0.01	0.094 (4.8e-03)	0.011 (7.7e-04)	0.087 (5.4e-04)	0.010 (1.2e-04)
NPE-DANN	0.10	0.127 (4.2e-03)	0.015 (8.0e-04)	0.122 (9.7e-03)	0.015 (1.3e-03)
NPE-DANN	1.00	0.197 (1.2e-02)	0.029 (3.1e-03)	0.193 (2.5e-02)	0.029 (7.5e-03)
NPE-MMD	0.01	**0.065 (6.5e-04)**	**0.007 (4.4e-05)**	0.070 (8.2e-03)	0.008 (7.6e-04)
NPE-MMD	0.10	0.067 (1.1e-03)	0.008 (2.1e-04)	**0.065 (4.3e-04)**	**0.008 (9.7e-05)**
NPE-MMD	1.00	0.324 (5.2e-04)	0.086 (2.0e-04)	0.325 (6.4e-04)	0.086 (2.9e-04)

Table B.2: **Experiment 2**: Overview of the metrics on a held-out validation data set from the training distribution (mean and standard deviation of three runs). For NNPE and NPE-DANN we see reduced performance on the training distribution. For NPE-MMD, we see that for successful runs, the performance on the training distribution improves. For settings with vanishing SSDD (compare Table 1) the performance drops massively, for both training distribution and observed distribution. This supports the notion that no meaningful information is learned in the summary space.

Method	λ	Prior (MNIST → USPS) NRMSE ↓	Prior (MNIST → USPS) PPD ↓	Prior (MNIST → USPS) SSDD	Likelihood Scale NRMSE ↓	Likelihood Scale PPD ↓	Likelihood Scale SSDD
NPE	-	**0.252 (2.6e-03)**	**0.081 (2.8e-03)**	0.249 (1.3e-03)	0.169 (3.1e-03)	0.019 (5.8e-04)	0.089 (1.3e-02)
NNPE	-	0.312 (1.7e-03)	0.106 (7.4e-04)	0.218 (1.9e-02)	0.186 (2.0e-03)	0.027 (5.0e-04)	0.043 (9.5e-03)
NPE-DANN	0.01	0.342 (2.3e-03)	0.150 (5.3e-04)	0.019 (2.1e-03)	**0.109 (1.2e-02)**	0.015 (3.0e-03)	0.020 (5.5e-03)
NPE-DANN	0.10	0.344 (3.5e-03)	0.152 (2.9e-03)	0.016 (6.6e-04)	0.110 (1.5e-03)	0.014 (1.9e-04)	0.012 (4.4e-04)
NPE-DANN	1.00	0.373 (1.4e-02)	0.169 (1.2e-02)	0.067 (6.8e-02)	0.135 (5.1e-03)	0.017 (7.3e-04)	0.011 (3.0e-04)
NPE-MMD	0.01	0.312 (2.3e-03)	0.134 (1.0e-03)	0.026 (2.0e-04)	0.134 (1.9e-03)	**0.012 (1.2e-04)**	0.018 (3.1e-04)
NPE-MMD	0.10	0.322 (9.9e-04)	0.141 (4.9e-04)	0.013 (9.7e-05)	0.323 (4.6e-04)	0.085 (1.8e-04)	0.000 (1.8e-06)
NPE-MMD	1.00	0.393 (2.9e-04)	0.185 (2.0e-04)	-0.000 (9.0e-07)	0.325 (3.1e-04)	0.085 (3.7e-05)	0.000 (1.6e-06)

Method	λ	Contamination (Noise) NRMSE ↓	Contamination (Noise) PPD ↓	Contamination (Noise) SSDD	Contamination (Rows) NRMSE ↓	Contamination (Rows) PPD ↓	Contamination (Rows) SSDD
NPE	-	0.326 (1.3e-02)	0.090 (8.9e-03)	0.374 (1.7e-02)	0.326 (7.2e-03)	0.090 (5.7e-03)	0.457 (3.7e-02)
NNPE	-	**0.176 (2.1e-03)**	**0.025 (5.8e-04)**	0.038 (1.8e-02)	0.202 (8.1e-03)	0.032 (2.4e-03)	0.036 (5.8e-03)
NPE-DANN	0.01	0.231 (1.0e-02)	0.045 (6.3e-03)	0.020 (1.7e-03)	0.174 (2.5e-03)	0.033 (2.5e-03)	0.024 (4.6e-03)
NPE-DANN	0.10	0.207 (9.2e-03)	0.034 (2.6e-03)	0.013 (1.7e-04)	**0.173 (5.4e-03)**	0.028 (6.9e-04)	0.014 (8.7e-04)
NPE-DANN	1.00	0.252 (1.3e-02)	0.047 (4.4e-03)	0.013 (8.3e-04)	0.223 (3.3e-02)	0.039 (1.0e-02)	0.014 (1.2e-03)
NPE-MMD	0.01	0.266 (2.3e-03)	0.053 (9.3e-04)	0.020 (7.8e-04)	0.264 (5.6e-03)	0.054 (2.5e-03)	0.017 (2.4e-03)
NPE-MMD	0.10	0.253 (1.8e-03)	0.048 (1.3e-03)	0.013 (9.2e-05)	0.175 (7.2e-03)	**0.026 (1.5e-03)**	0.011 (1.0e-04)
NPE-MMD	1.00	0.324 (1.2e-04)	0.085 (1.6e-04)	0.000 (3.9e-06)	0.324 (1.4e-04)	0.085 (1.5e-04)	0.000 (2.4e-06)

Table B.3: **Experiment 2**: Overview of the metrics in the different misspecification scenarios (mean and standard deviation of three runs). Please refer to Table 1 for a detailed description. Note that each standard deviation is given for a constant set of hyperparameters, so it only covers the computational uncertainty for a given setting. As shown by the performance changes when changing λ, hyperparameters have a large influence on the results, and different hyperparameter choices might lead to qualitative changes in the results.