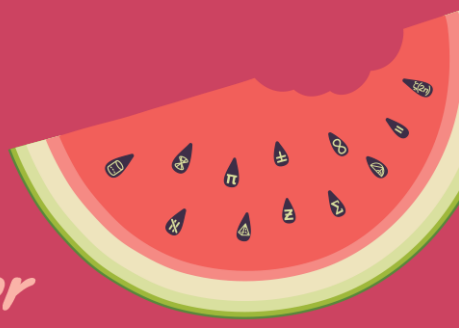
AMSI SUMMER RESEARCH SCHOLARSHIPS 2025–26

Get a taste for Research this Summer



Comparison of Regression-Adjusted ABC and Neural Methods for Simulation-Based Inference

Stephen Dang

Supervised by Prof. Christopher Drovandi and Dr. David Warne

Queensland University of Technology

Jane Street®

AMSI AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

Abstract

This project implements and benchmarks regression-adjusted rejection ABC (REJ ABC RA) within a standardised simulation-based inference evaluation pipeline, comparing it against plain rejection ABC and neural posterior estimation (NPE) across seven benchmark tasks and three simulation budgets (10^3–10^5). Regression adjustment consistently improves posterior quality over plain rejection ABC on tasks where the relationship between parameters and data summaries is approximately linear, in some cases halving the error at low budgets with negligible additional runtime. However, on tasks with multimodal or strongly nonlinear posteriors, a single linear correction can actively hurt performance at small budgets by miscorrecting accepted samples. NPE remains the most accurate method overall, but requires orders of magnitude more computation. These results clarify when regression adjustment is a reliable and essentially free upgrade to classical ABC, and when its assumptions break down.

1 Introduction

1.1 Motivation

Many problems in science and engineering involve models that are easy to simulate but hard to analyse mathematically. Given a set of parameters, such a model can generate synthetic data — but working backwards from data to parameters is difficult because the underlying probability of observing the data cannot be written down in a tractable form. Simulation-based inference (SBI) addresses this by using repeated simulations to approximate the answer instead (Lueckmann et al., 2021).

A large family of SBI methods exists, ranging from classical Approximate Bayesian Computation (ABC) — which accepts or rejects simulated parameters based on how closely they reproduce the observed data — to modern neural approaches that learn a model of the posterior directly from simulations (Lueckmann et al., 2021). Comparing these methods rigorously is challenging because results can depend heavily on the task, the simulation budget, and the choice of evaluation metric. Lueckmann et al. (2021) introduced SBIBM to address this, providing a standardised set of benchmark tasks, reference posteriors, and evaluation protocols that allow methods to be compared fairly.

Within classical ABC, a well-known issue is that accepted samples are rarely a perfect match to the observed data, introducing bias into the posterior approximation. Regression adjustment (RA) is a post-processing technique designed to correct for this, and Li and Fearnhead (2018) provide theoretical guarantees that it can improve posterior accuracy under appropriate conditions. This project implements regression-adjusted rejection ABC within the SBIBM framework and benchmarks it against plain rejection ABC and neural posterior estimation (Papamakarios & Murray, 2016; Greenberg et al., 2019), with the goal

of understanding when regression adjustment helps, when it fails, and how it compares to a neural alternative under matched simulation budgets.

1.2 Research Questions

This report addresses three research questions:

RQ1. Does regression adjustment consistently improve the posterior quality of rejection ABC across tasks and simulation budgets (10^3–10^5 simulations)?

RQ2. Which task characteristics predict when regression adjustment helps versus fails? In particular, how do properties such as local linearity of the parameter–summary relationship and multimodality of the posterior relate to RA performance?

RQ3. Under matched budgets, can regression adjustment make classical rejection ABC competitive with neural posterior estimation in accuracy and efficiency?

1.3 Contributions

The main contributions of this work are:

- **Implementation:** We implement regression-adjusted rejection ABC (REJ ABC RA) within the SBIBM-style evaluation pipeline, using the same protocol and budgets as baseline methods.
- **Controlled benchmarking:** We compare **REJ ABC**, **REJ ABC RA**, and **NPE** across multiple SBIBM tasks under matched simulation budgets (10^3–10^5) and repeated runs.
- **Evaluation:** We report posterior quality using **a sample metric** (MMD) and include runtime/efficiency summaries, aligning with SBIBM's emphasis that metric choice affects conclusions.
- **Interpretation via task geometry:** We organise tasks into two groups based on posterior/summary geometry (smooth/locally linear vs complex/multimodal) and use this grouping to explain when RA behaves as a strong "default upgrade" and when it can fail.
- **Documentation of scope:** We clearly document any excluded tasks and the reasons for exclusion, to preserve reproducibility and interpretability.

1.4 Statement of Authorship

All experiments, code implementation, and analyses presented in this report are my own work under the supervision of Professor Christopher Drovandi and Dr David Warne, except where explicitly acknowledged. Benchmark software (SBIBM / sbi) and third-party libraries are cited in the references.

2 Background and Related Work

2.1 Simulation-Based Inference (SBI)

In many scientific settings, the forward model is a stochastic simulator: given parameters θ, it generates synthetic data x, but the likelihood $p(x \mid \theta)$ is often intractable, making standard Bayesian or frequentist likelihood-based inference difficult. Cranmer et al. (2020) describe such models as *implicit* and emphasize that the core challenge is precisely the inability to evaluate the likelihood, motivating simulation-based inference as a more appropriate umbrella term than "likelihood-free inference."

SBI methods aim to approximate the posterior $p(\theta \mid x_o)$ using simulation. At a high level, the workflow is: specify a prior $p(\theta)$, generate simulations $x \sim p(\cdot \mid \theta)$, optionally reduce data to summaries $s(x)$, and then use an inference procedure to approximate the posterior. Cranmer et al. (2020) distinguish "simulator-in-the-loop" approaches (e.g., ABC) from surrogate-based approaches that train models such as neural conditional density estimators to amortise inference across observations.

2.2 Approximate Bayesian Computation (ABC)

Approximate Bayesian Computation (ABC) is a classical SBI approach that bypasses likelihood evaluation by comparing simulated and observed data (or summaries). As described by Sunnåker et al. (2013), all ABC-based methods approximate the likelihood function through simulation, comparing synthetic outputs against observed data. In the basic rejection ABC algorithm, parameters are sampled from the prior, data are simulated under those parameters, and the parameter draw is accepted if the simulated data are sufficiently close to the observed data under a distance function and tolerance ε. In rejection ABC, for each proposed parameter value θ, the simulator generates a synthetic dataset $\widehat{D}$, which is compared with the observed data D using a discrepancy measure $r(\widehat{D}, D)$. The parameter is accepted if this discrepancy is smaller than a tolerance threshold ε, that is, if $r(\widehat{D}, D) \leq \varepsilon$. Because exact matching is essentially impossible for nontrivial continuous data, ε must usually be strictly positive. As a result, ABC samples from an approximate posterior based on accepted simulations, $p(\theta \mid r(\widehat{D}, D) \leq \varepsilon)$, rather than the exact posterior $p(\theta \mid D)$.

A key practical issue is the curse of dimensionality: as Sunnåker et al. (2013) note, acceptance probability typically decreases as the dimensionality of the data increases, which substantially reduces computational efficiency. A common remedy is to replace the full data with a set of lower-dimensional summary statistics s(x) and accept based on distance in summary space. If summaries are sufficient with respect to the model parameters, this compression preserves all information about θ without introducing additional error. However, as Sunnåker et al. (2013) observe, sufficient low-dimensional statistics are typically unattainable for the complex models where ABC is most relevant, and using poorly chosen summaries can lead to

inflated credible intervals and additional posterior bias. The choice of summary statistics therefore becomes a major source of approximation error in practice.

2.3 Regression Adjustment (RA) for ABC

Regression adjustment (RA) is a widely used post-processing step designed to reduce ABC bias due to imperfect matching between simulated and observed summaries. Intuitively, even "accepted" simulations usually satisfy $s_i \approx s_{\text{obs}}$ rather than $s_i = s_{\text{obs}}$; RA attempts to correct accepted parameters $\theta_i$toward what would be expected if the summaries exactly equaled the observation.

Let s_{obs} denote the summary statistic computed from the observed data, let s_i be the summary statistic of the i-th accepted simulated dataset. Li and Fearnhead (2018) formalize the local linear regression adjustment by fitting a locally weighted linear regression of the accepted parameters θ_i on the summary discrepancy $(s_i - s_{\text{obs}})$. If $\hat{\beta}_\varepsilon$ denotes the estimated regression coefficient from this fit, where the subscript ε indicates dependence on the ABC tolerance and accepted sample set, then the adjusted sample is

$$\theta_i^* = \theta_i - \hat{\beta}_\varepsilon(s_i - s_{\text{obs}}).$$

Beyond this mechanism, Li and Fearnhead (2018) provide convergence results showing that, with appropriate bandwidth scaling, regression adjustment can yield a posterior that asymptotically correctly quantifies uncertainty, and they argue that the adjustment can be applied routinely when regression coefficients can be estimated accurately.

In practice, RA relies on assumptions that are often only approximately true: it is most effective when the parameter–summary relationship is locally close to linear and when the accepted sample cloud represents a single "local neighborhood" of the posterior. If the posterior is strongly nonlinear or multimodal, a single global linear correction can over- or under-correct.

2.4 Neural Posterior Estimation (NPE)

Neural Posterior Estimation (NPE) is a simulation-based inference approach that learns an approximation to the posterior distribution directly using a conditional density estimator. Rather than estimating the likelihood $p(x \mid \theta)$, NPE trains a neural network on simulated pairs (θ, x) to approximate the mapping from observations $x$to the posterior $p(\theta \mid x)$. This allows posterior inference to be amortized: once trained, the model can produce an approximate posterior for a new observation without running a separate inference procedure (Greenberg et al., 2019)

A key motivation for NPE is that it can work directly with simulator outputs, reducing reliance on hand-crafted summary statistics and leveraging neural networks to learn informative features from the data. Greenberg et al. (2019) note that posterior density estimation methods directly target $p(\theta \mid x)$, and therefore provide a natural way to perform amortized inference while taking advantage of flexible neural density estimators.

2.5 Benchmarks and metrics

Because SBI methods span many algorithmic families and are evaluated under different tasks, metrics, and budgets, comparing methods can be difficult. Lueckmann et al. (2021) argue that SBI comparisons have been fragmented—different studies use different tasks and metrics, comprehensive multi-task comparisons are rare, and metric choices can introduce bias—motivating the need for a shared benchmark framework. They introduce **SBIBM**, providing tasks, reference posteriors, metrics, plotting, and infrastructure tooling to enable rapid and reproducible comparisons. The framework is designed so that reference ("ground-truth") posterior samples are available for evaluation, enabling consistent two-sample testing.

SBIBM evaluates algorithms under fixed observations and varying **simulation budgets** (e.g., 1k to 100k simulations), and reports performance via multiple metrics and runtime.

The central posterior-quality metric is:

- **Maximum Mean Discrepancy (MMD):** a kernel-based two-sample test comparing samples from the inferred posterior to samples from the reference posterior. SBIBM notes that MMD can be sensitive to kernel hyperparameters, especially for complex or multimodal posteriors when using common heuristics such as the median heuristic for kernel length-scale selection.

SBIBM emphasizes that conclusions can depend on the evaluation metric used. Patterns that appear under the **Classifier Two-Sample Test (C2ST)**—a measure based on how well a classifier can distinguish posterior samples from reference samples—may not appear under the **Maximum Mean Discrepancy (MMD)**, depending on hyperparameter settings. The benchmark also notes that some metrics can be misleading, since strong point estimates do not necessarily imply a high-quality posterior approximation.
In this project, SBIBM provides the standardized evaluation setting and metrics needed to compare classical ABC variants (with and without regression adjustment) and NPE under matched simulation budgets.

3 Methods

This section describes the methods compared and the exact implementation choices used to ensure a fair comparison under matched simulation budgets.

3.1 Methods compared

All methods are evaluated on the same benchmark tasks, fixed observations, and simulation budgets $N \in \{10^3, 10^4, 10^5\}$.

Rejection ABC (REJ ABC).
Sample $\theta_i \sim p(\theta)$, simulate $x_i \sim p(x \mid \theta_i)$, compute a discrepancy d_i to the observed data x_{obs}, and retain the $k$smallest-distance particles. The accepted parameters approximate draws from the ABC posterior.

Rejection ABC with regression adjustment (REJ ABC RA).
Use the same accepted set as REJ ABC, then apply a local regression adjustment to reduce the bias induced by imperfect matching between simulated and observed outputs.

Neural Posterior Estimation (NPE).
Train a conditional density estimator $q_\phi(\theta \mid x)$ on simulated (θ, x) pairs using standard NPE with a normalising flow. Posterior samples are then generated by sampling from $q_\phi(\theta \mid x_{\text{obs}})$.

3.2 Regression adjustment for rejection ABC

Accepted set (top-k)

We use a **top-k** acceptance rule (rather than an ε-threshold). For a given budget N, we generate $\{(\theta_i, x_i)\}_{i=1}^{N}$ and compute distances

$$d_i = \| x_i - x_{\text{obs}} \|_2,$$

where simulator outputs and the observation are flattened into vectors. Let $\mathcal{A}$ denote the indices of the $k$smallest distances. The baseline REJ ABC posterior sample is $\{\theta_i\}_{i \in \mathcal{A}}$. We use $k = 100$by default and also test $k = 500$.

Local weighting

To emphasize locality around the observation, accepted particles are weighted using an Epanechnikov-style kernel based on their distance:

$$w_i = 1 - \left(\frac{d_i}{d_{\max}}\right)^2, d_{\max} = \max_{i \in \mathcal{A}} d_i, i \in \mathcal{A}.$$

Particles closer to x_{obs} receive higher weight.

Covariate preprocessing

We regress on the residual outputs $\Delta x_i = x_i - x_{\text{obs}}$. These residual vectors are standardised (z-scored) using the accepted set (with weighted mean and variance). To improve stability in high-dimensional outputs, we optionally apply PCA to obtain a reduced representation.

$$\tilde{x}_i \in \mathbb{R}^m, m \leq 50,$$

with an additional cap on m when k is small to reduce overfitting risk.

Regression model and adjustment

We fit a local linear model on the accepted set:

$$\theta_i = \beta_0 + \tilde{x}_i^\top B + \varepsilon_i, i \in \mathcal{A},$$

using weights w_i. We use either ordinary least squares or ridge regression (with ℓ_2 penalty) to stabilise estimation when the covariates are high-dimensional.

The regression-adjusted parameters are then computed as

$$\theta_i^* = \theta_i - (\tilde{x}_i - \tilde{x}_{\text{obs}})^\top \hat{B}, i \in \mathcal{A},$$

where $\tilde{x}_{\text{obs}}$ is the transformed representation of x_{obs} under the same preprocessing, and $\hat{B}$ is the fitted coefficient matrix. This subtracts the estimated local linear dependence of θ on the residual discrepancy, yielding parameters corresponding to the hypothetical case where the simulated output equals the observation. The output of REJ ABC RA is the set of $k$adjusted particles $\{\theta_i^*\}_{i \in \mathcal{A}}$, returned directly.

3.3 Neural posterior estimation setup

Neural Posterior Estimation (NPE) is trained directly on simulator outputs using paired samples (θ, x) generated from the prior predictive distribution. Inputs x and parameters θ are standardised independently. The posterior model is a conditional **Neural Spline Flow (NSF)** with a fixed architecture and fixed training hyperparameters across all tasks.

For a total simulation budget N, all N simulations are generated in a single stage from the prior, and the conditional density estimator is trained once on the resulting dataset.

After training, we draw $10{,}000$ posterior samples from the learned approximation,

$$\theta^{(j)} \sim q_\phi(\theta \mid x_{\text{obs}}), j = 1, \ldots, 10{,}000.$$

All neural hyperparameters are held constant across tasks and budgets; only the task, observation, simulation budget, and random seed vary between runs.

3.4 Computational setup and reproducibility

Experiments use:

- **Tasks:** 7 benchmark tasks
- **Budgets:** 10^3 to 10^5 simulations
- **Repeats:** 10 independent runs (random seeds) per method–task–budget setting
- **Hardware split:** REJ ABC and REJ ABC RA run on CPU; NPE runs on GPU.

For each setting, we aggregate results across seeds using the **mean** of each evaluation metric.

Runtime is measured as **end-to-end wall-clock time** required to produce posterior samples under a given simulation budget: for ABC methods, this includes simulation and top-k selection (and regression adjustment where applicable), and for NPE, this includes simulation, training across rounds, and posterior sampling.

4 Experimental Design

4.1 Tasks and evaluation protocol

We evaluate methods on **seven SBIBM benchmark tasks**, using SBIBM's standardised protocol: for each task, SBIBM defines multiple fixed observations and provides **reference posterior samples** for each

observation; algorithms are run at simulation budgets from 10^3 **to** 10^5 simulations and evaluated by comparing approximate posterior samples to the reference posterior samples.

To interpret when regression adjustment is expected to help, we group tasks by posterior / data–parameter geometry:

Group	Tasks	What to notice (task properties relevant to RA)
A — Smooth / locally near-linear	Gaussian linear; Gaussian linear uniform; Bernoulli GLM; Bernoulli GLM raw	**Gaussian linear / uniform** are simple 10-dimensional Gaussian models where $\theta$is the mean and covariance is fixed; the uniform-prior variant introduces truncated support. Bernoulli GLM has 10 parameters and is provided both with sufficient statistics (10-d) and raw data (100-d). The "raw" version increases feature dimension and can make local linear adjustment less stable (more parameters to fit from a small accepted set).
B — Complex / multimodal	SLCP; Two moons; Gaussian mixture	**SLCP** is constructed to have a simple likelihood but a complex posterior: uniform prior over five parameters, data are four 2D points, and the posterior has four symmetric modes with additional structure. **Two moons** is 2D with both global bimodality and local crescent-shaped structure, explicitly testing multimodality. **Gaussian mixture** is a mixture of two 2D Gaussians (one much broader than the other) and is common in the ABC literature. In these settings, accepted samples may come from multiple regions; a single local linear correction can over/under-correct when the accepted set spans distinct modes.

Across tasks, reference posterior samples are used as the evaluation target (SBIBM generates 10k reference posterior samples per observation in their setup).

4.2 Metric

We evaluate posterior quality using **two-sample tests** between approximate posterior samples $\{\theta_i\}_{i=1}^{n} \sim q(\theta \mid x_{\text{obs}})$ and reference posterior samples $\{\theta'_j\}_{j=1}^{m} \sim p(\theta \mid x_{\text{obs}})$, and we also report the runtime.

Maximum Mean Discrepancy (MMD).
MMD measures the distance between distributions by comparing their mean embeddings in a reproducing kernel Hilbert space. With kernel $k(\cdot,\cdot)$, the population quantity is

$$\mathrm{MMD}^2(p,q) = \mathbb{E}_{p,p}[k(\theta,\theta')] + \mathbb{E}_{q,q}[k(\theta,\theta')] - 2\mathbb{E}_{p,q}[k(\theta,\theta')].$$

In practice, this is estimated from posterior samples using the usual unbiased/biased sample estimators. In SBIBM, MMD is reported as MMD^2, where 0 is best. A key practical point is that MMD depends on kernel hyperparameters (e.g., RBF length-scale). SBIBM notes MMD can be sensitive to kernel choice, and that the commonly used median heuristic can struggle on multimodal posteriors such as Two Moons, making it harder to distinguish clearly different posteriors unless hyperparameters are adapted.

5 Result

5.1 Task-by-Task Breakdown

Group A — Smooth / Locally Near-Linear Tasks

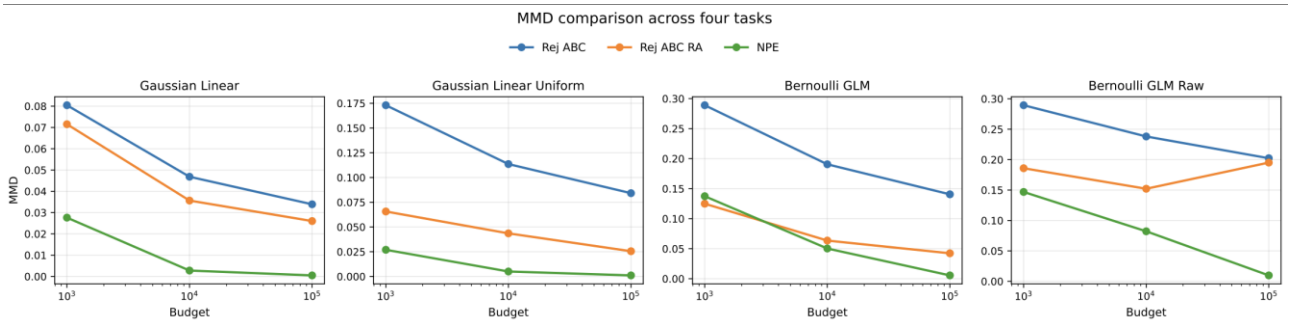


Figure 1: Group A performance: MMD vs Budget

Gaussian linear and Gaussian linear uniform are the clearest success cases for RA. On Gaussian linear, the three-way ordering (NPE best, REJ ABC RA middle, REJ ABC worst) holds cleanly across all budgets. Notably, at 10^5 REJ ABC RA (~0.026) nearly matches NPE (~0.0004), suggesting that on this smooth, well-behaved task, linear regression adjustment can largely close the gap with the neural method at sufficient budget. The uniform variant tells a similar story: RA provides a large relative gain over plain rejection at 10^3 (~0.065 vs ~0.175), maintains this advantage throughout, and again converges close to NPE by 10^5. These results confirm the theoretical prediction of Li and Fearnhead (2018): when the parameter–summary relationship is approximately linear, the local linear correction is well-specified and reliable.

Bernoulli GLM reinforces this picture. At 10^3, REJ ABC RA (~0.125) and NPE (~0.130) start at almost identical MMD — a striking result showing that RA is competitive with the neural method at low budget on this task. From 10^4 onward NPE pulls ahead and converges to near-zero MMD at 10^5, while REJ ABC RA levels off around 0.044. The improvement of RA over plain rejection (~0.29 at 10^3) is nonetheless consistent and substantial across all budgets.

Bernoulli GLM raw is the most instructive failure case within Group A. Both NPE and REJ ABC RA start at a similarly elevated MMD at 10^3 (~0.148 and ~0.188 respectively), reflecting the genuine difficulty of the 100-dimensional raw output under a limited simulation budget. From there their trajectories diverge sharply: NPE improves steeply and reaches near-zero MMD by 10^5, while REJ ABC RA improves marginally to 10^4 before rising back up at 10^5, crossing above plain rejection. This is a direct consequence of regression instability in high-dimensional covariate spaces — with 100-dimensional residual outputs, the local linear model becomes ill-conditioned even as more simulations become available, causing the correction to worsen rather than improve. This single task demonstrates that even within Group A, high-dimensional raw summaries can break RA, and the failure is intrinsic to the linear adjustment rather than a product of the comparison method.

Group B — Complex / Multimodal Tasks

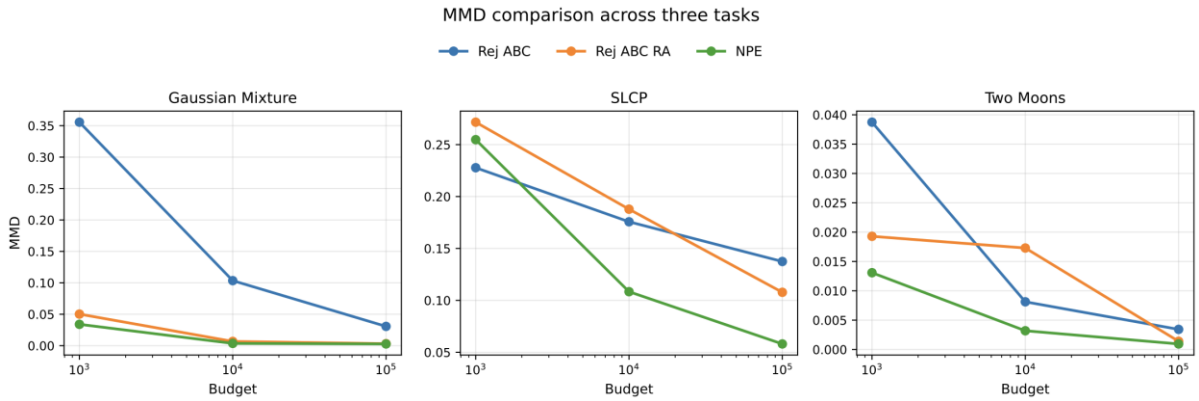


Figure 2: Group B performance: MMD vs budget

Gaussian mixture is a surprising result within Group B. Despite its multimodal structure, REJ ABC RA (~0.05) performs dramatically better than plain rejection (~0.356) at 10^3, and both converge to near-zero by 10^4–10^5. At 10^3, REJ ABC RA (~0.05) is worse than NPE (~0.033. Both methods converge to near-identical near-zero MMD by 10^5. The strong RA performance here is likely because the Gaussian mixture is only 2D, meaning accepted samples at low budget tend to concentrate within a single mode, making the local linear correction serviceable despite the global mixture structure. Plain rejection is severely hampered at small budget, making RA's relative gain over plain rejection the largest of any task.

SLCP is the clearest case where RA actively hurts. At 10^3, REJ ABC RA (~0.27) is noticeably worse than plain rejection (~0.23), confirming that the four-mode posterior structure causes accepted particles to span distinct regions, making a single global linear correction counterproductive. This underperformance persists through 10^4 where RA (~0.19) remains above plain rejection (~0.18), with RA only pulling slightly ahead at 10^5 (~0.11 vs ~0.135). NPE improves steeply from ~0.254 at 10^3 to ~0.060 at 10^5, while REJ ABC RA remains above 0.10 even at the largest budget, never closing the gap meaningfully. SLCP is therefore the task where the failure of linear RA is most sustained and the gap to NPE the most pronounced.

Two moons similarly shows RA underperforming plain rejection at moderate budgets. In Figure 2, REJ ABC RA (~0.019) starts better than plain rejection (~0.039) at 10^3, but at 10^4, RA (~0.017) performs worse than REJ (~0.008), with plain rejection actually crossing below RA. The crescent-shaped bimodal structure causes the linear adjustment to misalign accepted samples across the two modes. By 10^5 both methods converge closely (~0.002–0.003), with RA slightly better. NPE maintains a consistent and substantial advantage throughout, starting at ~0.013 at 10^3 and reaching ~0.001 by 10^5, while REJ ABC RA only approaches NPE-level accuracy at 10^5 after lagging considerably at smaller budgets.

5.2 Accuracy–Runtime Tradeoff

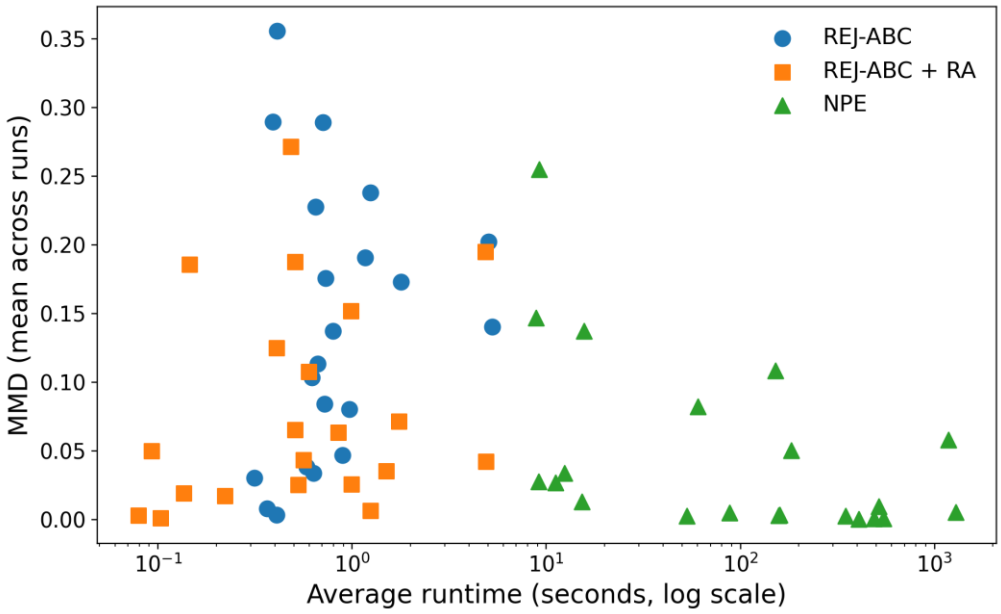


Figure 3: Accuracy-Runtime plot across task and budget

The scatter plot highlights a strong computational separation between the rejection-based ABC methods and NPE. REJ-ABC and REJ-ABC + RA both operate in a very low-runtime regime (approximately 0.08-5 seconds on average), whereas NPE requires substantially longer runtimes (approximately 9-1,284 seconds), with the gap widening at larger budgets. At comparable runtime scales, REJ-ABC + RA almost uniformly improves upon plain REJ-ABC, achieving lower MMD without any meaningful increase in computational cost, which indicates that the regression-adjustment step is practically negligible relative to simulation time. NPE remains the most accurate method overall, reaching the smallest MMD values across many task-budget settings, but these gains come at much higher computational expense. Hence, the practical trade-off is clear: REJ-ABC + RA is an attractive low-cost improvement over standard rejection ABC, while NPE is the preferred option when inference quality is prioritized over runtime

6 Discussion and Conclusion

6.1 Summary of Findings

This project implemented and benchmarked regression-adjusted rejection ABC (REJ ABC RA) within the SBIBM evaluation framework, comparing it against plain rejection ABC and neural posterior estimation across seven tasks and three simulation budgets.

Three main findings emerge from the experiments.

First, regression adjustment is a consistent and essentially free upgrade over plain rejection ABC when the task geometry is suitable. On smooth Group A tasks — Gaussian linear, Gaussian linear uniform, and Bernoulli GLM — RA reduces MMD substantially at every budget, in some cases halving the error at 10^3 simulations, while adding negligible runtime overhead. On Bernoulli GLM, RA at 10^3 achieves nearly identical MMD to NPE (~0.125 vs ~0.130), demonstrating that a well-specified linear correction can match a neural method at low budget on suitable tasks. These results confirm the theoretical motivation of Li and

Fearnhead (2018): when the parameter–summary relationship is locally well-approximated by a linear map, the correction is well-specified and reliable.

Second, the benefits of RA are not universal and task geometry is the key predictor of when it succeeds or fails. On multimodal tasks — SLCP and two moons — RA actively hurts performance at low to moderate simulation budgets, producing higher MMD than plain rejection. On SLCP this underperformance persists across all three budgets, with RA only marginally improving over plain rejection at 10^5 while remaining far behind NPE. On two moons, plain rejection crosses below RA at 10^4 before both converge at 10^5. In both cases the failure arises because accepted particles span multiple posterior modes, making a single global linear correction ill-defined. A different failure mode emerges on Bernoulli GLM raw: despite belonging to the smooth Group A, its 100-dimensional raw output makes the regression ill-conditioned, and RA's performance degrades at higher budgets rather than improving — crossing above plain rejection at 10^5. Together these cases illustrate that RA's assumptions — local linearity and a well-concentrated, unimodal accepted set — are genuine constraints, not just theoretical caveats.

Third, RA cannot close the gap to NPE under matched simulation budgets overall, though the gap is narrower than it might appear on smooth tasks. On Gaussian linear and Gaussian linear uniform, REJ ABC RA converges close to NPE by 10^5, and on Bernoulli GLM it matches NPE at 10^3. However, on multimodal tasks NPE maintains a substantial and persistent advantage at all budgets, and on Bernoulli GLM raw NPE handles the high-dimensional input cleanly while RA degrades. NPE also requires 3–5 orders of magnitude more computation than REJ ABC RA. In resource-constrained settings with smooth task geometry, REJ ABC RA therefore represents a practical and well-motivated choice; where task geometry is complex or posteriors are multimodal, NPE is the more robust option.

6.2 Limitations

Several limitations should be noted when interpreting these results.

Three SBIBM benchmark tasks — Lotka-Volterra (lv), SIR, and SLCP with distractors — were excluded from the evaluation due to compatibility issues with the experimental pipeline. These are among the more complex and higher-dimensional tasks in SBIBM, and their exclusion means the conclusions are drawn from a subset of the benchmark. In particular, the excluded tasks may be precisely the ones where RA would struggle most, potentially making the aggregate results more favourable to RA than a full benchmark would show.

The regression adjustment implemented here is restricted to local linear regression. While this is the most widely studied form of RA and is supported by the theoretical results of Li and Fearnhead (2018), the linear assumption is a genuine constraint. Nonlinear adjustments may perform better in tasks where the parameter–summary relationship is smooth but not linear — for instance, Blum and François (2010) proposed using a feed-forward neural network in place of linear regression for the adjustment step, and showed it can improve posterior approximation in such settings.

Finally, this work evaluates RA only in the context of rejection ABC. A natural extension would be to apply regression adjustment within sequential Monte Carlo ABC (SMC-ABC), where the accepted particles are already more concentrated around the posterior. In that setting, the local linearity assumption is more likely to hold even for complex tasks, potentially improving RA's performance in the multimodal regime.

6.3 Future Work

Three directions are most immediate. First, resolving the pipeline compatibility issues to include the three excluded tasks (lv, sir, and slcp distractors) would complete the SBIBM evaluation and provide a more comprehensive picture of RA's behaviour, particularly on high-dimensional and computationally expensive simulators.

Second, exploring richer regression adjustment strategies beyond local linear correction is a natural methodological extension. Non-linear adjustments — such as local polynomial, Gaussian process, or neural network-based post-processing — could better accommodate the curved parameter–summary relationships seen in multimodal tasks, potentially recovering performance in settings where linear RA currently fails.

Third, integrating regression adjustment into an SMC-ABC pipeline is a promising direction. Because SMC-ABC produces a more focused and localised accepted set through sequential tempering, the conditions under which linear RA is well-specified are more likely to be satisfied, even for complex posteriors. Benchmarking RA within SMC-ABC under the same SBIBM protocol would provide a more complete picture of how classical ABC can be strengthened.

6.4 Conclusion

This project demonstrates that regression-adjusted rejection ABC is a simple, computationally negligible enhancement that reliably improves posterior quality when the task geometry is smooth and locally linear. At the same time, the results clarify the conditions under which it fails: multimodal posteriors and high-dimensional raw summaries both break the linear correction assumption, and in these settings RA can be worse than doing nothing at small budgets. NPE remains the more robust and ultimately more accurate option across the full task suite, but at substantially greater computational cost. Regression adjustment occupies a clear and useful niche — as a low-cost default improvement for classical ABC in suitable problems — and the task grouping framework developed here provides a principled way to identify when that niche applies.

References

Lueckmann, J.-M., Boelts, J., Greenberg, D. S., Gonçalves, P. J., & Macke, J. H. (2021). Benchmarking simulation-based inference. *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, *130*, 1578–1589

Papamakarios, G., & Murray, I. (2016). Fast ε-free inference of simulation models with bayesian conditional density estimation. *Advances in neural information processing systems*, *29*.

Cranmer, K., Brehmer, J., & Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, *117*(48), 30055–30062. https://doi.org/10.1073/pnas.1912789117

Li, W., & Fearnhead, P. (2018). Convergence of regression-adjusted approximate Bayesian computation. *Biometrika*, *105*(2), 301–318 . https://doi.org/10.1093/biomet/asx081

Greenberg, D. S., Nonnenmacher, M., & Macke, J. H. (2019). Automatic posterior transformation for likelihood-free inference. *Proceedings of the 36th International Conference on Machine Learning*, *97*, 2404–2414. https://doi.org/10.48550/arXiv.1905.07488

Sunnåker, M., Busetto, A. G., Numminen, E., Corander, J., Foll, M., & Dessimoz, C. (2013). Approximate Bayesian Computation. *PLOS Computational Biology*, 9(1), e1002803. https://doi.org/10.1371/journal.pcbi.1002803

Blum, M. G. B., & François, O. (2010). Non-linear regression models for Approximate Bayesian Computation. *Statistics and Computing*, 20(1), 63–73. https://doi.org/10.1007/s11222-009-9116-0