Decisions, criteria and justifications (baselines, hard removal and metrics) Reviewers sXia, VmCi

Hard removal: Hard-to-learn samples (hard) can negatively impact model performance by either:

- Increasing uncertainty in predictions. The model may predict correctly but with low confidence, leading to unreliable decision-making.
- Reinforcing incorrect patterns. The model may misclassify these samples with high confidence, making it more prone to overfitting on noise rather than learning meaningful patterns.

Removing these samples is a well-established machine learning technique, often called dataset cleansing or data curation. Studies have shown that filtering out mislabeled or overly ambiguous samples can improve model generalization (https://arxiv.org/abs/1911.00068, https://arxiv.org/abs/2310.16981). Formally, let $X = X_{\text{easy}} \cup X_{\text{hard}}$ be the dataset, where X_{easy} are well-confident samples, and X_{hard} are ambiguous or mislabeled samples. The model's expected loss can be decomposed as:

$$\mathbb{E}[\mathcal{L}(X)] = \mathbb{E}[\mathcal{L}(X_{\text{easy}})] + \mathbb{E}[\mathcal{L}(X_{\text{hard}})]. \tag{1}$$

Since X_{hard} contributes disproportionately to loss without meaningful learning, its removal reduces noise and enhances generalization. Empirically, prior works on curriculum learning and label noise filtering (https://arxiv.org/pdf/1712.05055) support this approach, demonstrating that excluding ambiguous samples can lead to more robust models. To avoid confusion, we will refine the introduction (where we commented about these samples) to emphasize that hard sample removal is not an arbitrary step but a widely used strategy for improving model robustness.

Baseline models: The baseline models we selected adhere to the following criteria: 1. They represent traditional techniques primarily used for generating synthetic data, allowing us to demonstrate the impact of preprocessing and postprocessing phases. 2. We specifically chose a preprocessing technique to highlight that relying solely on it may not be sufficient for optimal performance. The choice of baseline models (Traditional and PreProcess) was motivated by comparing our framework (Profile2Gen) with common and data-centric baseline approaches. The Traditional method represents the generation of synthetic data without any type of profiling. The PreProcess method represents a data-centric approach that uses profiling in the preprocessing step but not in the postprocessing. These comparisons allow us to isolate the impact of the different steps of Profile2Gen and demonstrate the effectiveness of our approach. Specifically regarding the preprocessing step, we intentionally selected a profiling framework optimized for each dataset. This framework is chosen through an optimization process to identify the most suitable preprocessing method for each specific dataset. We did not compare our approach to traditional methods like simple feature selection, as we believe these methods do not adequately capture the complexity and improvements provided by our methodology. Using our profiling framework as a baseline, we demonstrated the added value of combining it with a postprocessing phase, which significantly enhances performance. However, we acknowledge that the main text should include a more detailed justification for the choice of baseline models. Due to space limitations, we plan to expand on this discussion in the appendix, with an explicit mention in the main paper. This will allow us to keep the main text focused on the methodology and results while thoroughly explaining our baseline model selection process.

Metrics: We chose to use the Wasserstein distance for our evaluation. The Wasserstein distance quantifies how much energy is needed to transform one distribution into another, making it a robust metric for capturing the similarity or divergence between distributions. We used this to compare the synthetic data against real data and analyzed the diversity-generalization trade-off, supported by statistical metrics such as quartiles, mean, and standard deviation. While FID and MMD are similar in that they aim to measure the differences between probability distributions, they are based on different mathematical principles. FID is particularly suited for images, as it operates in the feature space of pretrained deep networks, which might not be easily adaptable to our context. In contrast, MMD is a more general metric that can be applied to tabular data, but it requires mapping the data to a feature space using a kernel, which might not be the most suitable for our use case. Nevertheless, we acknowledge that FID and MMD are valuable metrics and could be explored in future work on synthetic data evaluation.

Could you provide further insights into the trade-off between generalization and diversity during the postprocessing stage, and how the method ensures that critical edge-case information is not lost? - Reviewer BQrC

The critical edge-case information depends on what you consider it. I mean, are low-probability events, which, despite being underrepresented, are important to have a synthetic representativity? If so, we demonstrated that Profile2Gen enhances model performance by preserving minority groups, which correspond to events with low representativeness in the dataset. However, the effectiveness of the technique varies depending on the generative model adopted. In cases where the technique fails to capture and represent minority data properly, a performance drop is observed. To understand why Profile2Gen better preserves these minority classes, we hypothesize that separating data groups during preprocessing allows the generative model to learn distinct distributions more effectively. In the context of Generative Adversarial Networks (GANs), let G_θ represent the generator with parameters θ, and D_ϕ the discriminator with parameters ϕ. The standard GAN optimization objective is given by: $\min_\theta \max_\phi \mathbb{E}_{x\sim p_{\text{real}}(x)} [\log D_\phi(x)] + \mathbb{E}_{x'\sim G_\theta(z)} [\log(1 - D_\phi(x'))]$ where $p_{\text{real}}(x)$ is the real data distribution, and $G_\theta(z)$ generates synthetic samples from a latent noise distribution $z \sim p(z)$. By training separate generative models for different subgroups, we effectively decompose $p_{\text{real}}(x)$ into **conditional distributions** $p(x|c)$, where c represents a specific subgroup. This modifies the objective function to: $\min_\theta \max_\phi \sum_c \mathbb{E}_{x\sim p(x|c)} [\log D_\phi(x)] + \mathbb{E}_{x'\sim G_\theta(z|c)} [\log(1 - D_\phi(x'))]$ This forces the generator to learn the distribution of each subgroup separately, reducing early collapse and improving representation for minority groups. However, synthetic data inherently introduce noise, and excessive divergence from $p(x|c)$ may result in synthetic distributions that fail to capture the underlying characteristics of minority classes. This is precisely why Profile2Gen proves more effective than PreProcessing (which lacks post-processing). When analyzing the histograms in Figure 8 in the main paper, we observe that the PreProcessing technique alone overrepresents certain classes. However, post-processing mitigates this overrepresentation. We hypothesize this correction occurs because overrepresented samples are categorized as hard samples during post-processing. This reclassification aligns the synthetic data distribution more closely with the real data while preserving meaningful diversity. Since this adjustment is based on model confidence scores, it ensures that the selected synthetic samples remain representative of the original dataset. Consequently, post-processing refines the synthetic data, ensuring it retains key structural properties while benefiting from the broader variations introduced by generative models.

"Given the variability in performance across different datasets, do you have insights into which characteristics of a dataset favor Profile2Gen over simpler preprocessing methods? Reviewer BQrC

We had not analyzed this aspect previously, but it could be included in our supplementary material. In general, Profile2Gen outperforms traditional preprocessing methods. When examining the protocols separately, we observe that in the TSTR scenario, Profile2Gen performs best in approximately 80% of the cases. However, the datasets where preprocessing outperformed Profile2Gen the most were Cholesterol and Diabetes, which also happen to be the two smallest datasets. Conversely, in the Parkinson dataset—the largest one—Profile2Gen demonstrated significantly better performance than preprocessing. This suggests that Profile2Gen handles larger datasets more effectively than preprocessing in a TSTR scenario. A similar pattern was observed in the augmentation scenario.

"How does error propagation from the early profiling stage affect the overall performance, and what measures can be taken to mitigate such issues?" Reviewer BQrC

In our context, an error could be the incorrect classification of a sample as a hard sample from the beginning when it should not be considered 'hard'. If this happens, the mislabeled sample will be included in the training process alongside the actual hard samples, and the synthetic data generated for it may be influenced by distributions that do not accurately reflect its true nature. However, since we apply a post-processing step, the impact of this error can be mitigated. Suppose the predictor's confidence level remains low even after training. In that case, the sample will continue to be classified as hard and re-evaluated in the post-processing stage. Thus, any initial misclassification can be corrected throughout the process, limiting its overall influence. The key point is that the decision to classify a sample as hard is based on the predictor's confidence level and a well-defined threshold, which minimizes the likelihood of significant errors. As a result, the potential impact of an initial misclassification should be assessed during post-processing, where necessary adjustments can be made.

A major weakness of the paper is the innovation in its method. As mentioned in the paper, the proposed method is largely based on existing traditional and preprocess approaches. The primary innovation lies in the filtering of the generated data after synthesis, which may limit the contribution's significance. Reviewer FkDs We appreciate your feedback and realize that the description of our contribution may not have been fully clear. Our approach is not a post-processing step but rather a data-centric technique that introduces a complete framework for improving synthetic data generation. Our work highlights the impact of the entire approach compared to using only a preprocessing strategy (PreProcess). While we include PreProcess for comparison with our framework, the steps adopted in preprocessing — such as applying flipping techniques and optimizing thresholds — are not typically followed in traditional preprocessing approaches to determine the optimal threshold. In fact, traditional preprocessing for regression tasks that rely on data profiling frameworks for preprocessing is not well established in the literature. However, other preprocessing methods, such as feature selection and dimensionality reduction are `https://www.europub.co.uk/articles/advancing-medical-diagnostics-with-deep-learning-and-data-preprocessing-A-744987`, and, more recently, LLM-based techniques, are emerging `https://arxiv.org/pdf/2402.17944`.s in our approach, the frameworks used for data profiling often rely on predictor confidence to categorize data into different groups. Due to this gap in existing methods, we adapted an existing framework to enable preprocessing specifically for regression tasks, making this another key contribution of our work.

Synthetic Data Proportions: The results show that adding 68.2% synthetic data degrades performance, but the exact reason is not deeply analyzed. Reviewer fkDs We did not explore this aspect in depth because performance degradation at high proportions of synthetic data is a well-documented behavior in data augmentation. Studies such as `https://www.sciencedirect.com/science/article/pii/S1877050923009687`Souza et al. (2023) and `https://arxiv.org/abs/2410.13098`Ashok & May (2024) discuss the existence of a saturation point, beyond which adding more synthetic data no longer improves performance and may even degrade it. Of course, no universal rule or established recommendation defines the exact proportion at which this saturation occurs, as it varies depending on the dataset and domain. As previously discussed, this effect can arise when synthetic data lacks diversity or novelty, effectively introducing redundancy or even noise into the training process. However, our experiments consistently showed that at approximately 68.2% of synthetic data, many models began to diverge significantly from the baseline, suggesting a critical point where performance starts to degrade. The proportions of synthetic data were randomly selected between 0.1 and 1. At higher levels, such as 90.9%, performance degradation became even more pronounced, likely due to the model overloading with non-generalizable information, particularly in the 'Traditional' and 'PreProcess' stages.

Model-Specific Performance: Some generative models (e.g., CTGAN) show high variability, but the authors do not discuss why. Reviewer fkDs

In machine learning, there is no silver bullet for solving problems. This is especially true for generative models, whose diversity reflects different trade-offs and applications. Each model has unique characteristics that make it more suitable for specific scenarios, and our experiments highlight this variability. Our primary goal was to analyze whether, despite these differences, our approach remains effective. Variability is an inherent characteristic of generative models due to their distinct underlying mechanisms. CTGAN, for instance, relies on deep neural networks, which are highly sensitive to training conditions, hyperparameters, and data distribution. This sensitivity can lead to fluctuations in performance (Xu et al., 2019, https://arxiv.org/pdf/1907.00503. In contrast, Bayesian Networks explicitly model dependencies between variables, making them less prone to such fluctuations (Heckerman, 2022, https://arxiv.org/abs/2002.00269). Our results confirm this pattern: lower RMSE values were observed in Bayesian Networks and CTGAN, particularly in TSTR, when trained with smaller synthetic datasets. This suggests that these models can generalize well to real data, as their generated distributions align closely with real distributions. However, the observed variation in performance across datasets indicates that factors such as dataset complexity, feature dependencies, and data impurity also play a role in model stability. Both models perform better with structured and cleaner data, as their ability to capture distributions improves in less noisy scenarios. Additionally, as detailed in Appendix D, we selected a medium-sized dataset for parameter tuning, considering dataset lengths and computational feasibility. We conducted an optimized parameter search using Optuna to minimize errors. However, this tuning was not applied to every dataset due to the large number of experiments and to avoid overfitting. Therefore, the choice of hyperparameters may have also influenced model variability. In summary, variability in performance is an expected outcome in machine learning models, where small parameter changes can lead to significant differences in results. Our study highlights both the most variable and the most stable models. Additionally, when evaluating model performance, we observed greater consistency in tree-based models such as Random Forest and Extra Trees, which split nodes based on impurity indices (`https://link.springer.com/book/10.1007/978-1-0716-1418-1`James et al. 2013) This characteristic may explain their robustness, particularly in the Profile2Gen experiments. To corroborate our hypothesis that the model's nature and dataset characteristics contribute to variability, we conducted an additional experiment where fine-tuning was performed on the same dataset used for training. This reduces external influences and allows us to analyze whether hyperparameter optimization is the main factor affecting performance. We randomly selected two datasets (Fat and Cholesterol) and evaluated CTGAN in two scenarios: Fine-tuning was performed on the same dataset used for training. Fine-tuning performed on an intermediary dataset - the same used in the main paper. The Figure in `https://anonymous.4open.science/r/icml2025-F9C4/table.png`

(table.png)

We observed that in Profile2Gen, variability was lower when the tuning was performed on the same dataset than when using an intermediary dataset. However, for the other approaches, the variability increased, suggesting that although hyperparameter tuning has an impact, it is not the primary factor driving performance fluctuations. Instead, our results indicate that the model's intrinsic nature and dataset complexity contribute more significantly to variability.We believe that we can enrich the work by creating a section in the appendix providing this discussion.