Journal of Data Science 0 (0), 1–19
July 2026
DOI: 10.6339/26-JDS1237
Statistical Data Science

A Practical Guide to Differentially Private Deep Learning Using the Pseudo Posterior Mechanism

Alexander J.Preiss[1,*], Amanda Konet[1], Robert Chew[1], Matthew R. Williams[2], Elan A. Segarra[3], David H. Oh[3], Erin Boon[4], and Terrance D. Savitsky[4]

[1]*Research Triangle Park, NC 27709, RTI International, Department of AI, Data, and Product Engineering, USA*
[2]*Research Triangle Park, NC 27709, RTI International, Center for Official Statistics, USA*
[3]*Hillcrest Heights, MD 20746, U.S. Bureau of Labor Statistics, Office of Compensation and Working Conditions, USA*
[4]*Hillcrest Heights, MD 20746, U.S. Bureau of Labor Statistics, Office of Survey Methods Research, USA*

Abstract

Privacy-preserving machine learning methods seek to train useful models that do not disclose information about the data on which they were trained. Such methods are vital when organizations train neural networks on sensitive individual-level data and seek to release the models publicly. Their goal poses a trade-off between predictive performance (utility) and privacy protection. That trade-off makes privacy-preserving machine learning methods difficult to apply in practice, usually requiring extensive iteration and hyperparameter tuning. Yet, practitioners often have little guidance for navigating competing statistical, computational, and privacy demands. We present an implementation algorithm for the Stochastic Weight Averaging–Gaussian Pseudo Posterior Mechanism (SWAG-PPM), a Bayesian differentially private deep learning method. The implementation algorithm focuses on the joint tuning of two key hyperparameters whose interaction governs model convergence and the privacy–utility trade-off. We introduce novel diagnostic tools to evaluate convergence and guide hyperparameter adjustments. Using a transformer model for occupational injury classification, we demonstrate that diagnostic-guided tuning with SWAG-PPM can achieve strong privacy protection and utility. While our case study uses a specific dataset and model architecture, all methodological steps can apply to other settings where privacy risk is heterogeneously distributed.

Keywords *Bayesian deep learning; differential Privacy; imbalanced learning; official statistics; pseudo posterior distribution*

1 Introduction

Organizations handling sensitive individual-level data, such as government statistical agencies and technology firms, face increasing demands to produce public data products while rigorously protecting individual privacy. Formal privacy frameworks, particularly those offering mathematically provable guarantees (Dwork, 2006), are becoming essential in this context. For example, the US Census Bureau used a differential privacy framework to protect privacy in 2020 Census Results (U.S. Census Bureau, 2021).

*Corresponding author. Email: apreiss@rti.org.

Received September 9, 2025; Accepted June 2, 2026

Privacy protection is often considered when releasing statistical results and public-use data files. However, organizations also generate sensitive data products in the form of statistical and machine learning models. Deep learning models, particularly transformer-based neural networks (Vaswani et al., 2017), have become state-of-the-art tools for regression and classification tasks (Devlin et al., 2019). For example, the US Bureau of Labor Statistics (BLS) uses transformer models to automatically code injury and illness narratives into structured categories (U.S. Bureau of Labor Statistics, 2025). Organizations may seek to release such models publicly when they could be useful in other contexts. By "release a model", we mean publicizing a model's parameter values (also known as weights) such that other organizations or individuals can reuse the trained model for their own purposes.

Releasing models poses unique privacy risks. Adversarial actors with access to a model can attack it to extract information about the data on which the model was trained (Rigaki and Garcia, 2024). The most common attack is membership inference, in which an adversary tries to determine whether a given record was included in the model's training data (Shokri et al., 2017). To put this in practical terms, if the BLS released a workplace injury autocoder model trained on survey data provided by businesses, an adversary could try to determine whether a specific business contributed data. Such a disclosure could have real-world implications on survey respondents' confidence in their confidentiality and, therefore, on survey response rates.

In response, methodologies to privatize models (rather than data) have emerged, seeking to train models with strict privacy guarantees while minimizing utility reduction relative to non-private models (Abadi et al., 2016). Utility, in this context, typically refers to the predictive performance of the model (for example, the utility of the BLS autocoder models is defined by their accuracy at classifying injury codes based on narrative texts). These goals pose a trade-off. All else being equal, the stronger the privacy guarantee, the weaker the utility.

Our recent work introduced a Bayesian mechanism to privatize deep neural networks (Chew et al., 2025). This approach, Stochastic Weight Averaging–Gaussian Pseudo Posterior Mechanism (SWAG-PPM), demonstrated strong privacy guarantees with minimal utility loss, even under relatively strict privacy settings. See Section 2 for details on the SWAG-PPM methodology.

In practice, regardless of the methodology, training private deep learning models is challenging. Achieving both privacy and utility goals requires more hyperparameter tuning than non-private deep learning models require to achieve high utility. In the case of SWAG-PPM, achieving high-quality parameter estimation requires tuning the number of fine-tuning epochs so that the model parameters evolve near the global mode. Simultaneously, a set of risk-based weights must be calibrated to meet a target privacy guarantee, which in turn alters the shape of the parameter distribution. This introduces a non-trivial interaction: tuning the privacy mechanism affects the model's parameter trajectory, and vice versa.

In this paper, we present an implementation algorithm to coordinate these dual tuning processes, guiding the real-world application of the SWAG-PPM methodology to neural network model training. Our approach aims to help data curators achieve their desired privacy guarantees while maintaining the predictive performance necessary for practical deployment.

2 SWAG-PPM Methodology

SWAG-PPM adapts a formal privacy framework, the Pseudo Posterior Mechanism (PPM) (Savitsky et al., 2022), to the task of private deep learning. The PPM was initially applied to privatize data. It adjusts the likelihood contribution of each record in proportion to its disclosure risk.

By selectively downweighting high-risk observations often found in lightly populated tails of the data distribution, the PPM confines privacy-induced distortion to regions of heightened identification risk while preserving fidelity in modal (low-risk) regions. This selective approach allows for robust utility in synthetic data releases, under a formal privacy guarantee that strengthens asymptotically as sample sizes increase.

To apply the PPM to deep neural network training, another method is needed to approximate the multivariate posterior distribution of the model's parameters. Estimating the posterior using traditional Bayesian methods is intractable, because the parameter space of modern deep neural networks is large. Prior work has demonstrated that the stochastic gradient descent (SGD) algorithm used to train these models can, with appropriate modifications, elicit a large-sample Gaussian approximation to the posterior distribution (Mandt et al., 2017). A practical implementation of this idea, Stochastic Weight Averaging–Gaussian (SWAG) (Maddox et al., 2019), works by continuing SGD with a high constant learning rate near the posterior mode. By collecting parameter snapshots during this phase, SWAG fits a multivariate Gaussian distribution that serves as an approximate posterior. This distribution can then be sampled to generate posterior draws of the model parameters.

SWAG-PPM combines the risk-weighted pseudo-posterior of the PPM with the posterior approximation procedure of SWAG (Figure 1). Given observations $\{y_i\}_{i=1}^n$, model parameters $\theta \in \Theta$, and prior $\pi(\theta)$, the weighted pseudo-posterior is defined as

$$\pi_\lambda(\theta \mid \mathbf{y}) \;\propto\; \pi(\theta) \prod_{i=1}^{n} p(y_i \mid \theta)^{w_i(\lambda)}, \tag{1}$$

where $w_i(\lambda) \in [0, 1]$ is the disclosure-risk weight for record i computed from the risk model parameterized by λ. Lower weights correspond to higher identification risk. Typically, to release a public version of a model trained by SWAG-PPM under a differential privacy guarantee, a single set of parameters θ is drawn from the weighted pseudo-posterior distribution.

The weighting scheme $w_i(\lambda)$ operates as a surgical intervention on the model distribution. Specifically, records that fall into "tails" of the predictive distribution (i.e., where disclosure risk is relatively high) are downweighted, while the majority of records in low-risk regions remain largely unperturbed. This localized downweighting achieves privacy guarantees with minimal sacrifice in model utility, since information from the safe regions of the distribution continues to contribute at near-full strength.

2.1 Sensitivity and Privacy Tuning

Differential privacy is defined in terms of neighboring datasets D and D' differing in a single record. A randomized mechanism $\mathcal{M}$ satisfies (ε, δ)-differential privacy if, for all measurable sets S,

$$\Pr[\mathcal{M}(D) \in S] \;\leqslant\; e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta. \tag{2}$$

Here, ε bounds the worst-case privacy loss, while δ quantifies the probability of a rare "catastrophic" disclosure event in which the guarantee is violated.

To connect this definition with the weighted pseudo-posterior in (1), we rewrite (2) as:

$$\pi_\lambda(\theta \mid \mathbf{y}) \leqslant e^{\varepsilon} \pi_\lambda(\theta \mid \mathbf{y}') + \delta, \tag{3}$$

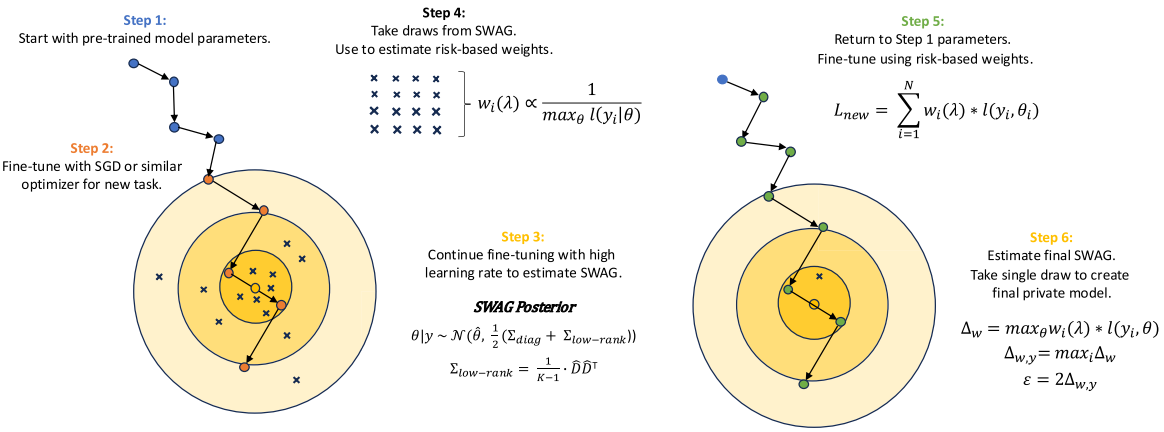

Figure 1: SWAG-PPM algorithm.

where the pseudo-posterior mechanism generates $\theta \in S$ and neighboring databases $\mathbf{y}$ and $\mathbf{y}'$ differ by a single record. We then introduce the *sensitivity* of the mechanism:

$$\Delta_{w,y} = \max_{y_i \in Y} \max_{\theta \in \Theta} |w_i(\lambda) \log p(y_i \mid \theta)|, \tag{4}$$

which characterizes the maximum contribution any individual record can make under the model. In finite samples, the effective privacy cost is bounded by this dataset-specific sensitivity,

$$\varepsilon = 2\Delta_{w,y}, \tag{5}$$

with δ absorbing the residual probability mass of tail events not fully controlled by the risk weights. Thus, both ε and δ are indirectly tuned through the disclosure-risk weighting scheme $w_i(\lambda)$.

To provide flexible tuning, constants (c, g) are introduced to scale and shift the weights:

$$w_i(\lambda) = \max\big(0,\ c \cdot (1 - \tilde{f}_i) + g\big), \tag{6}$$

where $\tilde{f}_i \in [0, 1]$ is the normalized risk score for record i, c is the slope parameter, and g is the intercept. This parameterization allows practitioners to calibrate the weighting function to achieve a target (ε, δ) privacy guarantee while preserving statistical efficiency.

2.2 SWAG Posterior Approximation

First, the neural network model is trained to convergence using standard stochastic gradient descent (SGD) or a suitable alternative. SGD is then run at a constant learning rate η in the vicinity of the posterior mode $\hat{\theta}$ to generate a sequence of parameter vectors $\{\theta_t\}_{t=1}^{T}$ targeting the distribution in (1). The SWAG procedure estimates the mean and covariance of these iterates as

$$\hat{\mu} = \frac{1}{T} \sum_{t=1}^{T} \theta_t, \tag{7}$$

$$\hat{\Sigma} = \frac{1}{T-1} \sum_{t=1}^{T} (\theta_t - \hat{\mu})(\theta_t - \hat{\mu})^{\top}. \tag{8}$$

This yields the Gaussian posterior approximation

$$\theta \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma}). \tag{9}$$

2.3 Asymptotic DP Guarantee

To release private models, we can draw M independent samples $\{\tilde{\theta}^{(m)}\}_{m=1}^{M}$ from the approximate posterior. While a single draw is sufficient for release, multiple draws can support uncertainty quantification.

In finite samples, the privacy analysis relies on local sensitivity. This dataset-specific sensitivity $\Delta_{w,y}$ determines the effective (ε, δ) bounds for that dataset (see Section 2.1). Here, ε bounds the worst-case privacy loss, while δ quantifies the probability of a rare "catastrophic" disclosure event in which the guarantee is violated. Because $\Delta_{w,y}$ depends on the observed data, the finite-sample privacy parameters are dataset-dependent.

As the sample size n increases, the local sensitivity converges to the global sensitivity:

$$\Delta_{w,y}^{(\text{local})} \to \Delta_{w}^{(\text{global})}, \quad \text{as } n \to \infty, \tag{10}$$

and the corresponding dataset-dependent parameters converge to global values:

$$(\varepsilon(n), \delta(n)) \to (\varepsilon, 0). \tag{11}$$

The convergence rate follows the stochastic approximation error,

$$\varepsilon(n) - \varepsilon = O\left(n^{-1/2}\right), \qquad \delta(n) = O\left(n^{-1/2}\right). \tag{12}$$

In practice, sample sizes exceeding 10,000 are sufficient for $\varepsilon(n)$ to be effectively indistinguishable from the global ε, with $\delta(n)$ near zero. Thus, privacy risk diminishes as data scale grows, while utility is preserved through the localized weighting strategy introduced in Section 2.1.

2.4 Key Hyperparameters in SWAG-PPM

While SWAG-PPM involves several tunable hyperparameters (Appendix Table 1), two play a central role in determining both privacy and utility outcomes: the number of weighted *fine-tuning* (FT) epochs and the disclosure-risk weight parameters c and g. Throughout, ***FT epochs*** refers to step 5 in the SWAG-PPM algorithm (Figure 1), i.e., the mode-finding step run after the initial (non-private) training pass used to compute per-record losses for PPM weights, and before the SWAG estimation phase. This step does not use the high constant learning rate employed by SWAG; instead, it follows the standard optimizer schedule (e.g., decays or early stopping) to relocate the parameters to the weighted posterior mode implied by the PPM objective. The subsequent SWAG estimation phase then runs at a constant learning rate to collect parameter snapshots for the Gaussian approximation.

Fine-Tuning (FT) Epochs The number of FT epochs controls how long optimization is continued under the risk-weighted loss to reach (and stabilize around) the weighted posterior mode. Too few epochs can leave the parameters off-mode, biasing the mean and covariance estimated during SWAG. In contrast, too many FT epochs not only yield diminishing returns in terms of improved convergence but also increase the risk of overfitting. This overfitting may occur in two ways: (i) standard generalization overfitting to the training set, where the model begins to memorize noise or idiosyncrasies, and (ii) overfitting to the weighted training distribution defined by the PPM, leading to excessive specialization on the privacy-weighted objective and potentially narrowing the posterior to the point where SWAG samples exhibit limited variability.

Disclosure-Risk Weight Parameter (c) As outlined in Section 2.1, PPM has two privacy hyperparameters c and g which indirectly set the privacy guarantee through weights $w_i(\lambda)$. The slope c controls how quickly the weight declines with increasing disclosure risk. Larger c values make the decline steeper, so high-risk records are downweighted more relative to low-risk ones. However, because c also scales the maximum weight, larger values allow low-risk records to exert stronger influence, which can increase sensitivity and weaken privacy. In contrast, smaller c values flatten the decline and reduce all weights, lowering the maximum contribution of any record and thereby strengthening privacy, though at the cost of reducing effective sample size and potentially harming utility. The intercept g shifts all weights upward or downward: decreasing g uniformly reduces record contributions, further strengthening privacy but possibly reducing utility. Since c governs the trade-off between relative risk-based weighting and overall sensitivity, we focus on it as the primary privacy hyperparameter for the remainder of the paper.

Hyperparameter Interactions Crucially, c and *FT epochs* are *not* independent. Changing c alters both the curvature of the weighted loss and the location of the weighted posterior mode, requiring a re-run of the mode-finding phase to converge to the appropriate minimum. Larger c values create a steeper loss basin, which can make optimization appear to converge more quickly, but also increase minibatch gradient variance and instability. Smaller c values flatten the loss surface and reduce variance, which stabilizes optimization but typically requires more FT epochs to achieve convergence. In all cases, the FT epoch count must balance adequate convergence against the twin risks of underfitting (too few epochs) and overfitting (too many). This interaction motivates the structured *Implementation Algorithm* presented next, which coordinates the joint selection of c and *FT epochs* to achieve target privacy guarantees while preserving predictive performance.

3 Implementation Algorithm

Our recommended procedure for implementing SWAG-PPM is summarized at a high level in Figure 2. After completing an initial training run, the process begins by checking if the model meets established privacy (ε) and utility (e.g., classification accuracy) targets. These evaluations, described in Section 4.1, determine whether the model meets predefined privacy guarantees and utility targets. Given that training SWAG-PPM can be computationally intensive, it is often practical to stop when a model satisfies both criteria, rather than conduct extensive hyperparameter tuning in pursuit of incremental gains.

The next step is to assess model convergence. Convergence is critical for utility: models that are underfit have potential for improved predictive performance, while overfit models often exhibit degraded generalization. Tools for diagnosing convergence in SWAG-PPM are outlined in Section 4.2.

If privacy and utility requirements are not met after convergence is achieved, adjustment options are available. Section 4.3 describes strategies such as increasing the linear downweighting parameters in the PPM weighting function to strengthen privacy protection, or revisiting the initial requirements if utility targets remain unmet. This structured sequence ensures that decisions to accept, adjust, or retrain are informed by clear evaluation criteria and an explicit understanding of the privacy–utility trade-off.

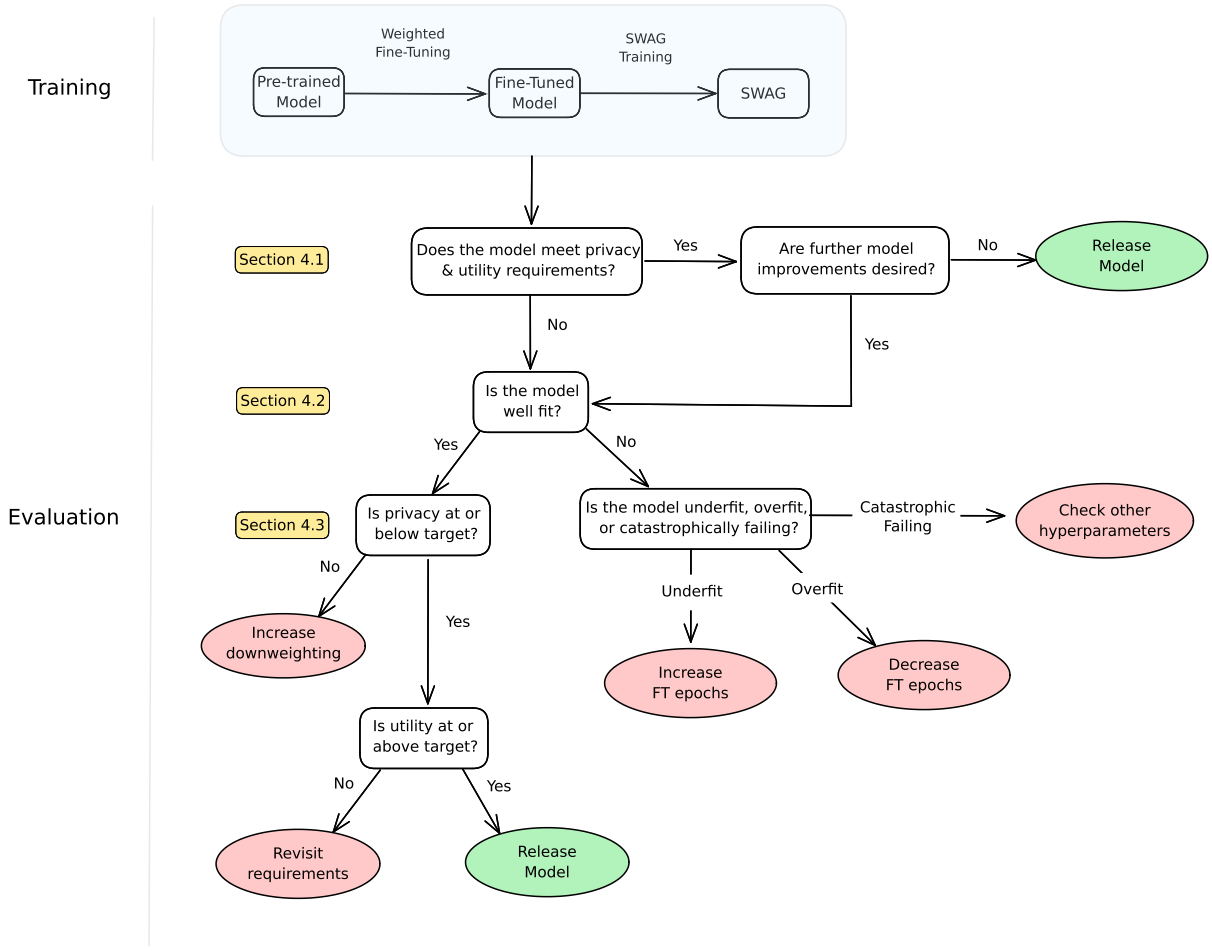

Figure 2: SWAG-PPM evaluation flowchart.

4 Application to OSHA Severe Injuries Dataset

To demonstrate this process in action, we use the publicly available Severe Injury Reports dataset (Chew, 2025) collected by the U.S. Occupational Safety and Health Administration (OSHA) from January 2015 through September 2023 under regulation 29 CFR 1904.39. Each record includes a "Final Narrative" free-text description of the incident and is coded using version 2.01 of the Occupational Injury and Illness Classification System (OIICS) into fields such as "Nature of Injury". In its raw form, the dataset contains 86,210 observations spanning 199 distinct OIICS "Nature of Injury" codes, exhibiting a pronounced class imbalance typical of occupational injury data (and much other real-world data).

For this demonstration, we aim to train a private machine learning model that predicts the "Nature of Injury" code based on the "Final Narrative" free-text description. This aim corresponds to the BLS survey autocoding use case mentioned in Section 1 (U.S. Bureau of Labor Statistics, 2025). We use a supervised transfer learning approach, i.e., we fine-tune a base model on "Nature of Injury"-"Final Narrative" pairs from the OSHA data, and we evaluate the model on a held-out set of pairs. Fine-tuning was performed starting from the weights of a pre-trained DistilRoBERTa transformer (Liu et al., 2021).

4.1 Evaluating Privacy and Utility

Our initial model was trained with key hyperparameters set to $c = 1$ and $FT\ epochs = 7$; for other default hyperparameter settings, see Appendix Table 1. Selecting effective starting values is important but challenging to do well *a priori* before any runs have begun. We began with $c = 1$ as a neutral baseline: this setting preserves the full weight of low-risk records while still reducing the influence of high-risk ones, providing a stable foundation for evaluating privacy–utility trade-offs. We subsequently adjusted c only if privacy diagnostics indicated that stronger downweighting of high-risk records was necessary. The starting number of $FT\ epochs$ was guided by prior experience with similar data but could also be chosen via early stopping criteria.

Privacy and Utility Requirements Organizations must establish clear privacy and utility requirements when training differentially private machine learning models. The privacy parameter, ε, is the upper bound on acceptable privacy loss during SWAG-PPM training and controls the privacy-utility trade-off. Smaller values of ε indicate a stricter upper bound on privacy loss, thus stronger privacy protections, at the expense of utility. Larger values of ε provide weaker privacy protections but afford greater model utility. However, unlike other differential privacy methods for machine learning such as DP-SGD (Abadi et al., 2016), users cannot directly set the target ε of SWAG-PPM. Instead, ε is the actual privacy cost given the data's risk distribution, choice of weights, and model parameters, making it an output of SWAG-PPM rather than a pre-specified input. While users cannot directly control ε, they can influence it indirectly by adjusting how aggressively the algorithm downweights risky records through the disclosure-risk weight parameters like c.

Selecting the appropriate target ε value is challenging, as prior work has emphasized that there are no established ε selection methods (Hsu et al., 2014; Dwork et al., 2019; Near et al., 2025). Furthermore, the data context and the disclosure risk for each practitioner's use case determine what makes an "acceptable" ε choice (Near et al., 2025). At $\varepsilon < 1$, privacy is strongly protected, which is ideal for use-cases with strict privacy requirements. Consequently, deep learning models trained with such low ε are likely to have poor utility. In the case of SWAG-PPM, low ε values are unlikely in practice for datasets with relatively risky observations. Larger ε values ($1 < \varepsilon < 20$) can still provide privacy and are more desirable for use-cases in which data are less-sensitive or high accuracy is important. However, whether those protections are meaningful is context-dependent (Near et al., 2025). Some evidence suggests potential vulnerabilities at higher values, as outliers may be leaked during attacks when $\varepsilon \geqslant 10$ (Nasr et al., 2021; Stadler et al., 2022). In high-risk scenarios, set the ε target to the smallest ε value that maintains acceptable utility levels. For lower-risk scenarios, larger ε can be tolerated in favor of better model utility.

Real-world implementations illustrate this variability in practice. For example, Apple uses $\varepsilon = 16$ for next-word prediction, a task in which model accuracy is critical and individual contributions are low risk, and a more conservative $\varepsilon = 2$ for health-related telemetry, reflecting greater data sensitivity (Apple Inc, 2017). Similarly, Google uses ε between 0.994 and 13.69 for their next-word prediction (Google Research, 2024) but opted for $\varepsilon = 2.64$ per user per day for their COVID-19 Community Mobility Reports, reflecting the sensitivity of location data and cumulative privacy loss from recurring data releases (Aktay et al., 2020). For temporal data or recurring data releases, ε targets should account for privacy budget accumulation and draw on experience with data of a similar class (Hu et al., 2022). In 2021, U.S. Census Bureau revealed that they revised their ε to 19.61, much higher than their previously reported ε of 12.2. The

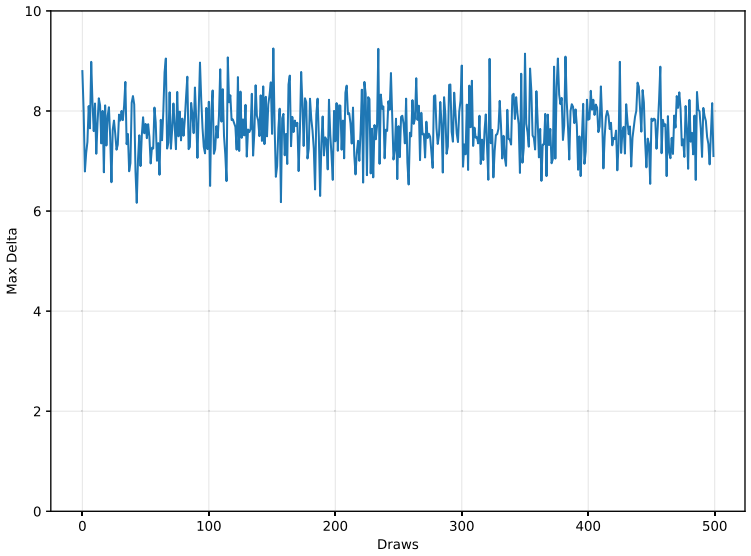

Figure 3: Max delta across 500 draws, c=1, 7 FT epochs.

Census Bureau's approach illustrates how organizational priorities can shift parameter choices, as they noted the privacy budget was increased due to data users' requests for improved accuracy for various race and ethnic groups at various geography levels (U.S. Census Bureau, 2021). A proposed community-driven differential privacy deployment registry would help foster norms related to ε, risk, and utility (Howarth et al., 2025).

Likewise, SWAG-PPM also does not allow users to fix δ, which can make model performance comparisons across different mechanisms challenging. However, prior work has demonstrated that for text classification tasks with large number of imbalanced classes like this case study, SWAG-PPM consistently outperformed alternatives like DP-SGD, even when they were allowed to train under extremely loose privacy parameters (up to $\delta = 0.99$) (Chew et al., 2025). Extending the PPM framework from local to global sensitivity in the finite-sample setting has been investigated by Hu et al. (2022), who used a clipped log-likelihood. Further extensions of SWAG-PPM using this approach remains an active area of research.

Utility requirements are context-dependent and should be informed by community norms and existing performance benchmarks. The adequacy of model performance depends on the standards established within specific domains and how well existing non-private models perform on the same or similar tasks. A practical approach for organizations to assess utility is to establish internal benchmarks by comparing the performance of the differentially private model against the performance of a non-private model trained on the same dataset.

For the OSHA data case study, we set a target ε of 10 and established a utility threshold of no more than 10% degradation in the F1 scores from the non-private baseline trained for 7 epochs (macro F1 = 0.43, weighted F1 = 0.92).

Max Delta Plots To evaluate privacy, we created max delta plots (Figure 3) to depict the highest per-record loss observed across all training examples for each posterior draw. The plot's x-axis represents a draw from the SWAG-PPM posterior and the y-axis indicates the highest weighted likelihood value across all observations for that draw. The SWAG-PPM privacy guarantee is defined as twice the "max delta" (i.e., the highest value across all the draws).

These plots help identify whether high-risk records are being sufficiently downweighted, and can help spot patterns such as persistent outliers or instability across draws. Because it responds directly to changes in c, the plot also serves as a practical feedback tool when tuning hyperparameters, allowing practitioners to assess whether privacy improvements are being achieved without excessive utility loss.

A well-fitting model would typically exhibit low, stable max delta values across draws, with low variability and no large spikes, resulting in ε values that consistently remain below the target privacy budget. High, persistent variability across draws in the max delta plot could indicate underfitting, in the sense that the model's parameter samples are still moving through a broad, unstable region of the loss landscape. This can happen when mode-finding is incomplete or when the risk-based weights are relatively flat (low c), which reduces discrimination between high- and low-risk records and can slow convergence. In contrast, low variability punctuated by sudden spikes may signal overfitting or instability: most draws capture dominant patterns consistently, but occasional draws overemphasize high-loss outliers, producing spikes. Such behavior is especially likely when c is large, since steep weighting shrinks the effective sample size and amplifies the influence of rare, high-risk records. We discuss these dynamics more in Section 4.2.

After training the OSHA data for 7 FT epochs using $c = 1$, (Figure 3), we see moderate variability in the max delta values across 500 draws, with most values fluctuating between approximately 6.5 and 9.2. This suggests that the model is underfit and has not yet converged. To obtain ε, we multiply the max delta across the draws by two (Savitsky et al., 2022). For $c = 1$ and $FT\ epochs = 7$ using the OSHA data, we observe a $\varepsilon = 18.5$, higher than our target ε of 10. High ε, combined with evidence of underfitting in Figure 3, indicates that we should continue training more FT epochs.

Utility Plots To evaluate model utility, we relied on test set performance metrics, with a particular focus on the F1 score. We reported F1 scores using both macro and weighted averages to capture different aspects of performance under class imbalance. The macro F1 treats all classes equally, providing insight into performance across rare and common classes alike, while the weighted F1 gives proportionally more influence to frequent classes, reflecting their prevalence in the dataset.

The macro and weighted F1 scores were computed using test set predictions from 30 posterior draws of the SWAG-PPM model and visualized using violin plots. This visualization provides a clear view of both the central tendency and the dispersion of performance across draws, allowing us to assess the stability of the model's predictions under posterior uncertainty. Large variability in these distributions can indicate sensitivity to parameter draws, whereas narrow, high-density regions suggest more stable performance. A well-fitting model should exhibit stability in the macro and weighted F1 scores across draws.

Figure 4 presents the violin plots for the model trained using $c = 1$ and FT epochs $= 7$. The macro F1 (left) and weighted F1 (right) distributions are tightly concentrated with minimal variance, centered around 0.43 and 0.92, respectively. The stability of these distributions indicates little sensitivity to parameter draws for either score. The substantial difference between the macro and weighted F1 scores reflects the class imbalance of the OSHA data, where strong performance on frequent classes drives the higher weighted F1 score in all draws. Though the privacy requirement was not met, the resulting model using $c = 1$ and $FT\ epochs = 7$ met our utility requirements.

In a different data context users may find that the distribution of utility across parameter draws has more variation, which may lead to the possibility of drawing a set of parameters with

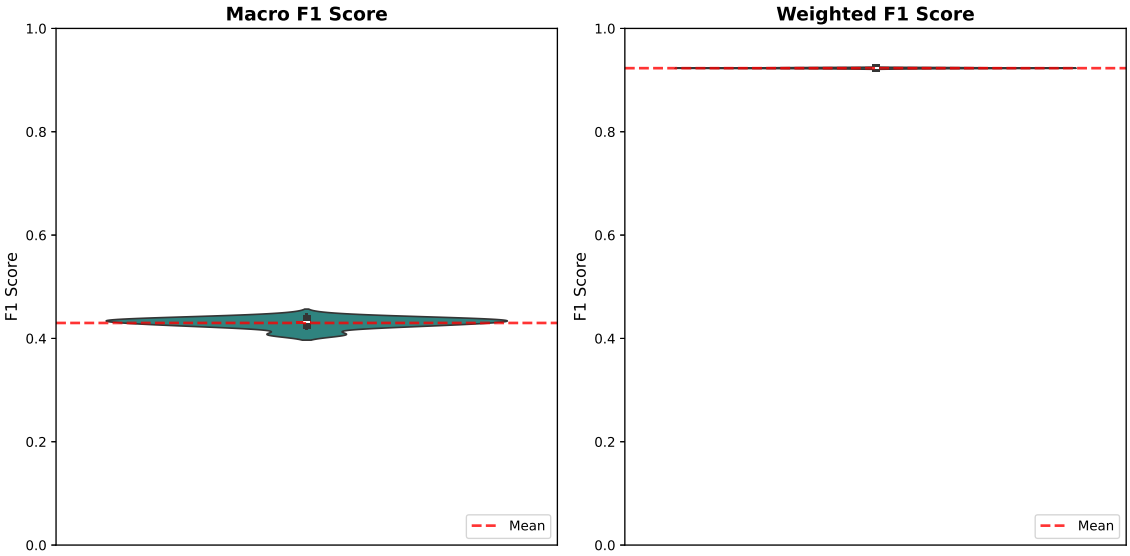

Figure 4: F1 scores, c=1, 7 FT epochs.

poor F1 scores. A private selection mechanism similar to that adopted in Hod and Canetti (2025) can alleviate this potential problem. This mechanism consists of repeatedly sampling from the posterior distribution until a draw exceeds a previously set utility threshold. Since only the final draw is utilized, the privacy expenditure of this mechanism has been shown to be tighter than simply adding together the expenditures of each draw. See Liu and Talwar (2019) for further details on the privacy accounting as well as other private selection mechanisms that may be employed to allow for utility improvements.

To further interpret these results, class-specific metrics and the confusion matrix can offer valuable complementary perspectives. They help identify specific classes for which the model performs well or struggles, guiding targeted refinements in model architecture, training, or hyperparameter tuning. Together, these diagnostics provide a comprehensive understanding of the model's utility while acknowledging uncertainty across posterior samples.

Further Tuning vs. Timeline and Budget Even when a model meets predefined privacy and utility requirements, additional hyperparameter tuning may still be worthwhile to push performance or protection further. For example, privacy could be improved by increasing the aggressiveness of risk-based downweighting (Section 4.3), or utility could be enhanced by refining mode-finding epochs or other training parameters (Section 4.2). These refinements can yield incremental gains that may be valuable for specific deployment contexts or policy targets. To determine how much hyperparameter tuning is worthwhile, we suggest a two-step process. First, get a sense for the magnitude of change in privacy and utility when hyperparameters are changed. If it took more than one iteration to meet privacy and utility requirements, this may already be evident. Second, estimate the cost of a disclosure or prediction error. This informs the value of a given increase in privacy or utility. Together, these estimates can produce an expected return on tuning time, which can be considered alongside timeline and budget. Practically, further tuning after meeting privacy and utility requirements is most likely to be worthwhile when organizations release models infrequently.

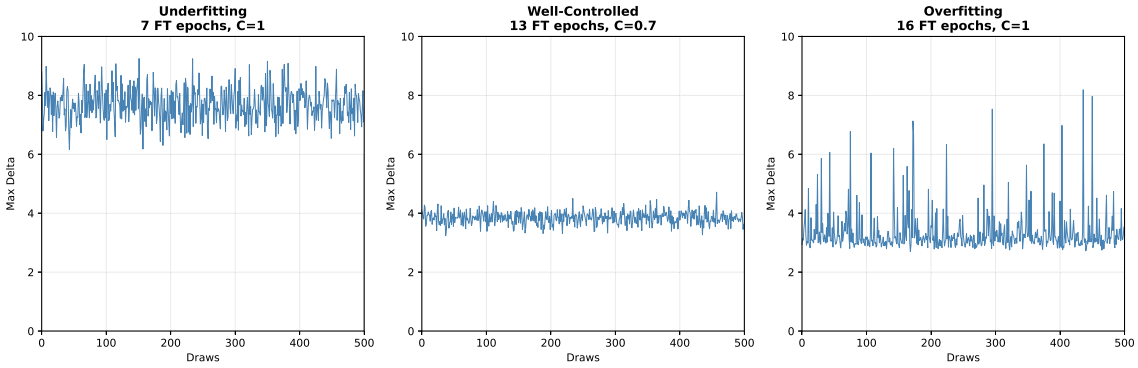

Figure 5: Assessing model fit.

Convergence and Privacy Lastly, it is important to note that evaluating privacy before a model has fully converged provides only an approximation to the final posterior mode. The risk profile observed at this stage may not match that of the final converged model, since parameter trajectories can shift as training continues. If this approximation is unsatisfactory for your application (e.g., in cases where high-stakes decisions require precise privacy diagnostics) it may be prudent to proceed directly to the next stage of the procedure, ensuring the model has converged before conducting definitive privacy assessments.

4.2 Diagnosing Model Convergence

Model convergence is a critical consideration when implementing SWAG-PPM because both privacy and utility outcomes depend heavily on the stability of the model's parameter estimates. As with other forms of deep neural network training, convergence is not guaranteed. A model that has not converged may still be traversing the parameter space, leading to suboptimal predictive performance and unstable privacy diagnostics. Conversely, an overtrained model can overfit to idiosyncratic patterns in the training data, increasing the risk of memorizing individual records and thereby compromising privacy. Striking the right balance in training duration is therefore essential for ensuring that the SWAG-PPM posterior draws faithfully represent the intended privacy–utility trade-off.

Underfitting vs. Overfitting Unlike more established Bayesian procedures, SWAG does not yet offer a mature set of diagnostic tools for formally assessing convergence. To address this gap, the max delta plot introduced in the last Section can also be used as an initial diagnostic aimed at identifying convergence-related issues. This plot, which tracks the maximum per-record loss across posterior draws, provides an interpretable signal for detecting both underfitting (persistent high variability) and overfitting (stable performance punctuated by spikes linked to outlier records). Understanding these patterns is not only valuable for evaluating the current model but also for informing how to structure future runs, such as adjusting mode-finding epochs or refining privacy tuning parameters.

Figure 5 presents a panel of outputs illustrating models that are underfit, well-fit, and overfit. In the third plot where $FT\ epochs = 16$ and $c = 1$, the max delta plot exhibits pronounced spikiness characteristic of instability due to overfitting. This instability arises from misclassification of records in the distributional tails, which produce large negative log-likelihood contributions.

The effect is amplified by the strong influence of majority categories, which dominate the loss landscape under weaker downweighting. As c is reduced below 1, the risk-based weighting more strongly downweights high-loss examples, tempering the model distribution. Smaller c values effectively flatten the loss surface and reduce variance, which stabilizes optimization but typically requires more FT epochs to achieve convergence. This adjustment reduces mode dominance, stabilizes the treatment of tail records, and yields smoother max-Δ behavior with more balanced privacy-utility outcomes.

Catastrophic Failure Catastrophic failure refers to a breakdown in model training where the predictive utility is unacceptably low, often rendering the model unusable. In the context of SWAG-PPM, this failure mode typically occurs when the optimization process becomes trapped in an unfavorable region of the loss landscape from which recovery is difficult or impossible without restarting training. The result is a model that fails to capture meaningful patterns in the data, leading to poor generalization and ineffective privacy–utility trade-offs.

One common cause of catastrophic failure is an excessively high learning rate during mode-finding. When the step size is too large, gradient updates can overshoot the weighted posterior mode, propelling the parameters into a region of the loss surface with steep, unstable gradients or flat plateaus. In such cases, the optimizer may oscillate without making meaningful progress or may descend into a suboptimal basin where further improvements are blocked. Because SWAG-PPM relies on stable convergence to approximate the weighted posterior, these early missteps can prevent the collection of useful parameter snapshots for posterior construction.

This phenomenon is particularly problematic in SWAG-PPM due to the interaction between the learning rate and risk-based weights. If high-risk observations are sharply downweighted (large c), the effective sample size shrinks, increasing the variance of stochastic gradient estimates and making overshooting even more likely. Conversely, a poorly chosen c in combination with a high learning rate can amplify the influence of dominant classes, pulling the model away from tail-region structure needed for balanced performance.

To guard against catastrophic failure, careful tuning of the learning rate schedule, early monitoring of training accuracy, and diagnostic checks such as loss-trajectory plots or max-delta plots are essential. If signs of persistent low accuracy and unstable loss behavior appear, it is often more efficient to restart training with adjusted hyperparameters rather than continue a run that is unlikely to recover.

4.3 Hyperparameter Search

To visually summarize the application of this process to the OSHA data, Figure 6 shows a grid of different runs where we varied the hyperparameters c and $FT\ epochs$ to find an acceptable model. As described in Section 4.1, we began by training a model on OSHA data with $c = 1$ and $FT\ epochs = 7$. We then iterated on the SWAG-PPM evaluation flow chart (Figure 2) until a model met all privacy, utility, and fit criteria. The initial model met the utility requirement but not the privacy requirement. The max delta plot showed signs of underfitting, with high and highly variable max deltas. Following the flowchart, we set $FT\ epochs = 10$ for the next iteration. The second iteration still did not meet the privacy requirement, this time with max delta spikes indicating overfitting. The transition from underfitting to overfitting in only three epochs suggested that we were unlikely to meet our privacy target with $c = 1$. The third iteration used $c = 0.85$ and $FT\ epochs = 7$ and showed signs of underfitting. We then added epochs for two more iterations, until we began to overfit again without meeting the privacy requirement.

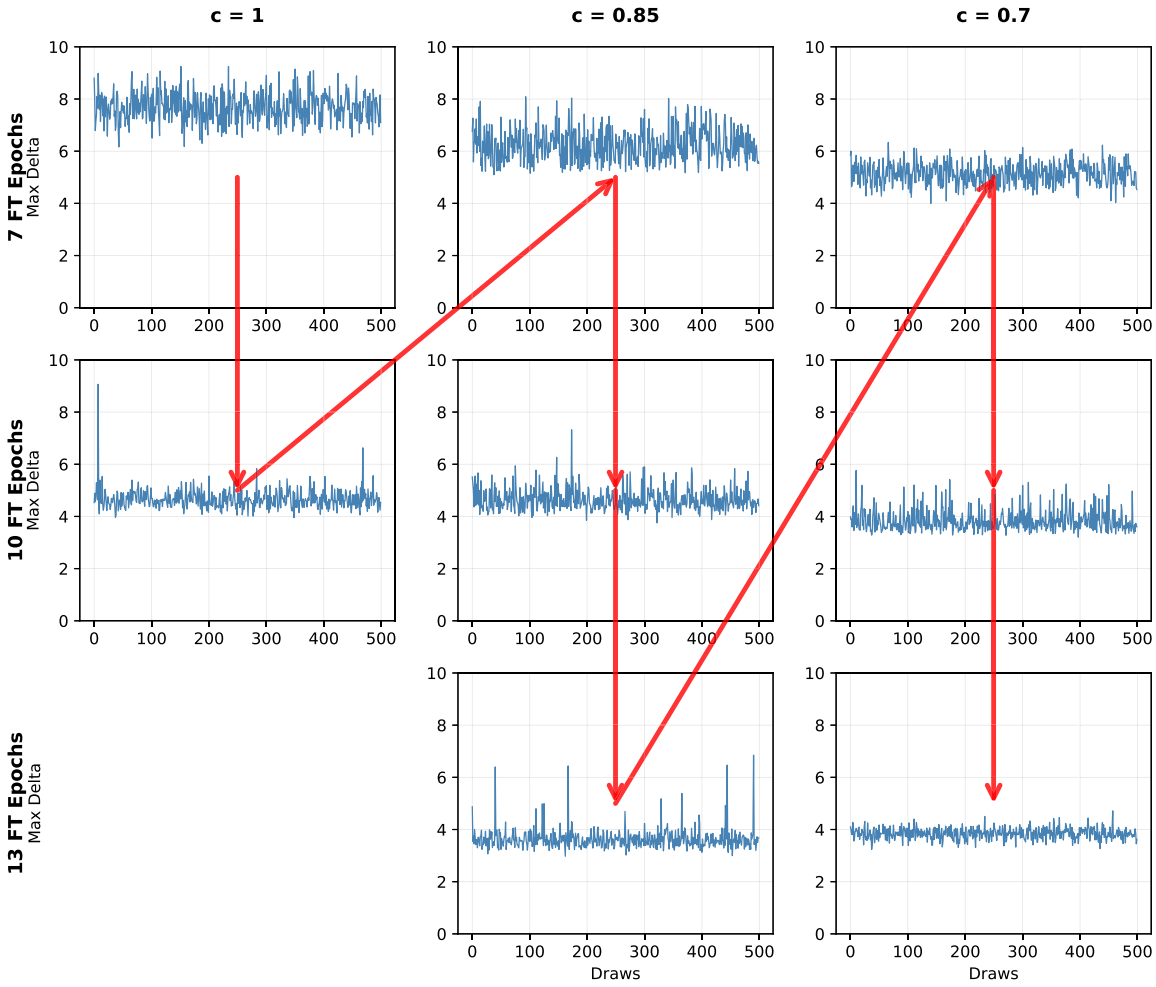

Figure 6: Hyperparameter grid search.

The sixth iteration used $c = 0.7$ and $FT\ epochs = 7$. The max delta plot showed signs of underfitting, but less so than the runs with $FT\ epochs = 7$ and higher c values. We added epochs for two more iterations, until the model with $c = 0.7$ and $FT\ epochs = 13$ met both privacy and utility requirements.

In this example, all model iterations had very similar utility, well above the utility requirement. This focused the search process on finding hyperparameter values that met the privacy requirement. Other use cases may entail a stronger privacy-utility trade-off, making the search process more challenging. If meeting one requirement requires failing at the other, two options are available. First, the private selection mechanism referenced in Section 4.1 may increase utility with only a small additional privacy expenditure. Second, the requirements may be reconsidered. In a context with a strong privacy-utility trade-off, releasing a model with worse privacy or utility than originally planned may be better than not releasing a model at all.

This example hyperparameter search illustrates the additional time and care required for the practical use of SWAG-PPM, relative to non-private model training. A single SWAG-PPM model fit is more computationally expensive than a non-private model fit, and the additional tuning required due to the interaction of c and $FT\ epochs$ compounds that difference. Hyperparameter

optimization (HPO) techniques (Bischl et al., 2023) could reduce the amount of human input needed, but would likely require more iterations to find a solution. HPO would also require scoring functions to replace the qualitative assessment of max delta plots in our human-in-the-loop process.

5 Discussion

In this paper, we provided a procedural framework for SWAG-PPM that helps practitioners make decisions about training and privacy-utility trade-offs. Based on prior experience, we focus on the interplay between two main hyperparameters, the disclosure-risk slope parameter c and the number of fine-tuning (FT) epochs, and introduce novel diagnostic plots and strategies (e.g., max delta plot) to help navigate issues such as model convergence and hyperparameter tuning.

We also presented a case study, in which we followed our procedural framework to conduct hyperparameter search for an example SWAG-PPM use case. In the case study, we used OSHA data to train a model to predict occupational injury codes using free-text incident descriptions. We set a privacy requirement of $\varepsilon < 10$ and a utility requirement of no more than 10 percent degradation in the F1 scores from the non-private baseline trained for 7 epochs (macro F1 = 0.43, weighted F1 = 0.92). We followed the SWAG-PPM evaluation flow chart (Figure 2) for eight iterations until we found a combination of c and $FT\ epochs$ that met both privacy and utility requirements.

Although our case study used a pre-trained transformer model and a specific occupational injury dataset, the procedural insights generalize to other architectures and domains where disclosure risk is heterogeneously distributed. In particular, the joint tuning of privacy-weight parameters and convergence-critical training epochs should be seen as a core component of applying Bayesian privacy mechanisms in deep learning contexts. Additionally, our experience suggests that starting from stable, moderately conservative hyperparameters and then adapting based on diagnostic feedback, offers a practical balance between computational cost and final model quality.

Future work could refine SWAG-PPM in several directions. On the methodological side, adaptive HPO schemes to jointly update c and $FT\ epochs$ during training could further automate the tuning process, reducing manual iteration. On the theoretical side, extending the asymptotic privacy guarantee proofs to finite-sample settings would strengthen confidence in practical deployments. Finally, from an applied perspective, evaluating SWAG-PPM on domains with different class imbalances, text lengths, or feature modalities would test its robustness and reveal domain-specific tuning patterns.

Overall, SWAG-PPM provides a promising bridge between formal privacy protection and modern neural network modeling. By addressing both the statistical and computational challenges of integrating risk-based pseudo-posteriors into SGD-trained models, our implementation algorithm moves toward a practical, reproducible workflow for releasing high-utility deep learning models with strong privacy guarantees.

Supplementary Material

Complete classification report with per-class metrics

Appendix

Table 1: Hyperparameters for SWAG-PPM.

Hyperparameter	Importance	Description
Training Epochs	Primary	The number of passes over the entire training data set (Default: 7). Unlike linear models that have a convex likelihood function that can be optimized directly, deep learning models are highly non-convex and require iterative gradient-based optimization methods for training. Several full passes over the data may be required for the model to converge to a local minimum due to the high dimensionality of the parameter space and complex loss landscape. Too few epochs can result in underfitting whereas too many epochs can result in overfitting. For SWAG-PPM, the number of training epochs comes into play for both the fine-tuning and SWAG estimation steps. For SWAG training, more epochs generally result in more stable posterior estimation though it takes longer to train and can cause memory issues if too many intermediate parameter sets are saved on the GPU. For SWAG estimation, we've found the guidance in the original paper of 20 training epochs is generally a good starting place (Maddox et al., 2019). For fine-tuning, see the guidance in Section 3 for thorough coverage.
Risk-based Weighting Terms (`c`, `g`)	Primary	The slope (`c`) and intercept (`g`) terms of the linear risk-based weights (Default: `c`=1 and `g`=0). The risk-based weights downweight observations based on their loss (higher loss, higher downweighting). Modifying the `c` and `g` hyperparameters allow for scaling and shifting the weight distribution respectively, allowing users to balance privacy vs. utility trade-offs. Lower values of `c` result in all weights being scaled down (e.g., `c`=0.8 means the highest weight is 0.8). Lower values of `g` result in the weight distribution becoming increasingly right-skewed (e.g., `g`=-0.4 will have more weights at zero than `g`=0). Lower values of `c` and `g` will usually result in better privacy protection at the expense of worse utility.
Base Model	Secondary	The set of model parameters we begin training from (Default: `distilRoBERTa`). Instead of starting from constant or randomly initialized parameter values, it's common to start from a model pre-trained on a task like the one you're interested in (i.e., "transfer learning"). This can greatly improve the efficiency and stability of training and generally translates to improved utility. While larger models with more parameters tend to result in better predictive performance, there is normally a trade-off in increased training and inference time. Additionally, all else equal, larger models tend to have a greater capacity to overfit and may be relatively more likely to leak sensitive data (Carlini et al., 2022).

Continued on next page

Hyperparameter	Importance for Tuning	Description
Batch Size	Secondary	The number of observations used in a single gradient-based parameter update (Default: 8). Due to the huge number of parameters and observations used with deep learning models, it's often computationally infeasible to use the entire training set when performing gradient descent (memory constraints, slow updates, etc.). Instead, partitions of the full training set, called minibatches, are commonly used. The batch size is a less sensitive hyperparameter for SWAG-PPM than for other differentially private methods (e.g., DP-SGD). Practitioners should follow the same batch size recommendations as for non-private deep learning (smaller batches of 8 or 16 usually preferred).
Learning Rate	Secondary	The step size of parameter updates during training (Default: $fine\ tuning = 5e^{-5}$ and $posterior\ estimation = 0.01$). The learning rate determines how much the model's parameters are adjusted in response to the computed gradient. A larger learning rate means the parameters can change more each update whereas a smaller learning rate means parameters change less each update. For fine-tuning, practitioners can use the same intuition as in non-private settings. Weighted fine-tuning may require slightly higher learning rates to help with convergence, since downweighting makes the effective gradient updates smaller on average. For SWAG estimation, the learning rate should be higher than typically used for training or fine-tuning to help explore the space around the local minimum discovered during fine-tuning.
L2 Regularization (Weight Decay)	Secondary	The penalty to the likelihood function, discouraging large parameter values (Default: $fine\ tuning = 0.01$ and $posterior\ estimation = 0$). L2 regularization shrinks parameter estimates toward zero, helping with training stability and preventing overfitting. In Bayesian models (like SWAG), it corresponds to a Gaussian prior over parameters: smaller L2 values = weaker prior, larger L2 values = stronger prior. For fine-tuning, use the same intuition as in non-private deep learning. Weighted fine-tuning and SWAG estimation may require lower L2 regularization since downweighted observations contribute less to the likelihood, amplifying the effect of shrinkage.
Posterior Draws	Secondary	The number of draws taken from the posterior distribution (Default: 500). A draw is a set of model parameters. In SWAG-PPM, SWAG draws are initially used to generate risk-based weights, and later, used to calculate privacy (ϵ) and utility. A higher number of posterior draws is desirable to assess privacy ($\geqslant 500$) and utility ($\geqslant 30$), though more draws increase run time.

References

Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, ..., Zhang L (2016). Deep learning with differential privacy. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (E Weippl, S Katzenbeisser, C Kruegel, A Myers and S Halevi, eds.), 308–318.

Aktay A, Bavadekar S, Cossoul G, Davis J, Desfontaines D, ..., Wilson RJ (2020). Google COVID-19 community mobility reports: Anonymization process description (version 1.1).

Apple Inc (2017). *Differential Privacy Overview. White Paper*. Apple Inc., Cupertino, CA.

Bischl B, Binder M, Lang M, Pielok T, Richter J, ..., Lindauer M (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(2): e1484.

Carlini N, Ippolito D, Jagielski M, Lee K, Tramer F, Zhang C (2022). Quantifying memorization across neural language models. In: *The Eleventh International Conference on Learning Representations*.

Chew R (2025). *OSHA Severe Injury Reports: Jan 2015 - Sep 2023*. https://doi.org/10.6084/m9.figshare.28669604.v1

Chew R, Williams MR, Segarra EA, Preiss AJ, Konet A, Savitsky TD (2025). Bayesian pseudo posterior mechanism for differentially private machine learning. arXiv preprint: arXiv:2503.21528.

Devlin J, Chang MW, Lee K, Toutanova K (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of NAACL-HLT* (J Burstein, C Doran and T Solorio, eds.),

Dwork C (2006). Differential privacy. In: *International Colloquium on Automata, Languages, and Programming* (M Bugliesi, B Preneel, V Sassone and I Wegener, eds.), 1–12. Springer.

Dwork C, Kohli N, Mulligan D (2019). Differential privacy in practice: Expose your epsilons! *Journal of Privacy and Confidentiality*, 9(2). https://doi.org/10.29012/jpc.689

Google Research (2024). *Advances in Private Training for Production on-Device Language Models. Blog Post*. Google Research, Mountain View, CA.

Hod S, Canetti R (2025). Differentially private release of Israel's national registry of live births. In: *2025 IEEE Symposium on Security and Privacy (SP)*, 3912–3930. IEEE.

Howarth G, Altman M, Ayalde S, Ghazi E, McCallum C, ..., Near J (2025). A community-driven differential privacy deployment registry, *Technical Report NIST Internal or Interagency Report (NISTIR) 8588*, (Draft), National Institute of Standards and Technology.

Hsu J, Gaboardi M, Haeberlen A, Khanna S, Narayan A, ..., Roth A (2014). Differential privacy: An economic method for choosing epsilon. In: *2014 IEEE 27th Computer Security Foundations Symposium*, 398–410.

Hu J, Williams MR, Savitsky TD (2022). Mechanisms for global differential privacy under bayesian data synthesis. arXiv preprint: arXiv:2205.05003.

Liu J, Talwar K (2019). Private selection from private candidates. In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (M Charikar and E Cohen, eds.), 298–309.

Liu Z, Lin W, Shi Y, Zhao J (2021). A robustly optimized bert pre-training approach with post-training. In: *China National Conference on Chinese Computational Linguistics* (S Li, M Sun, Y Liu, H Wu, K Liu, W Che, S He and G Rao, eds.), 471–484. Springer.

Maddox WJ, Izmailov P, Garipov T, Vetrov DP, Wilson AG (2019). A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32: 13153–13164.

Mandt S, Hoffman MD, Blei DM (2017). Stochastic gradient descent as approximate Bayesian inference. *Journal of Machine Learning Research*, 18(134): 1–35.

Nasr M, Songi S, Thakurta A, Papernot N, Carlin N (2021). Adversary instantiation: Lower bounds for differentially private machine learning. In: *2021 IEEE Symposium on Security and*

Privacy (SP), 866–882.

Near J, Darais D, Lefkovitz N, Howarth G (2025). Guidelines for evaluating differential privacy guarantees, *Technical Report NIST.SP.800-226*, National Institute of Standards and Technology.

Rigaki M, Garcia S (2024). A survey of privacy attacks in machine learning. *ACM Computing Surveys*, 56(4): 1–34. https://doi.org/10.1145/3624010

Savitsky TD, Williams MR, Hu J (2022). Bayesian pseudo posterior mechanism under asymptotic differential privacy. *Journal of Machine Learning Research*, 23(55): 1–37.

Shokri R, Stronati M, Song C, Shmatikov V (2017). Membership inference attacks against machine learning models. In: *2017 IEEE Symposium on Security and Privacy (SP)*, 3–18.

Stadler T, Oprisanu B, Troncoso C (2022). Synthetic data – anonymisation groundhog day. In: *31st USENIX Security Symposium (USENIX Security*, volume 22, 1451–1468. USENIX Association, Boston, MA.

US Bureau of Labor Statistics (2025). Automated coding of injury and illness data. White paper.

US Census Bureau (2021). *Census Bureau Sets Key Parameters to Protect Privacy in 2020 Census Results Press Release*. U.S. Census Bureau.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, ..., Polosukhin I (2017). Attention is all you need. In: *Advances in Neural Information Processing Systems* (I Guyon, U Von Luxburg, S Benigo, H Wallach, R Fergus, S Viswanathan and R Garnett, eds.), volume 30.