Security, Privacy, and Fairness Analysis for HW4

Taha Majlesi
Student ID: 810101504
Trustworthy Artificial Intelligence
University of Tehran

***Abstract*—This report presents a complete, reproducible implementation of HW4 with emphasis on theoretical correctness and empirical interpretability. For security, the real poisoned checkpoint is analyzed via Neural Cleanse, attacked-label detection is performed by lower-tail MAD, and one-epoch unlearning is evaluated by clean accuracy and ASR before/after mitigation. For privacy, Laplace mechanism behavior is derived from first principles and evaluated under base, sequential-composition, and unbounded-adjacency assumptions. For fairness, baseline and assignment-required mitigation are compared with two bonus methods (reweighing and group thresholds), and results are decomposed into both aggregate metrics and group-level behavior. Every value in this report is generated by executable code.**

I. Introduction

Trustworthy AI is a multi-objective design problem: models should resist adversarial manipulation, leak limited information about individuals, and avoid systematic group-level harm. This assignment is a compact instance of that broader agenda, because it requires analyzing one model family through three distinct lenses with conflicting objectives. The central challenge is to maintain methodological consistency while interpreting metrics that encode different notions of risk: security risk (backdoor exploitability), privacy risk (query disclosure through noise calibration), and fairness risk (disparate outcomes across sensitive groups).

II. Related Work and Positioning

A. Backdoor Detection and Model Repair Context

The security part of this report is grounded in optimization-based trigger reconstruction, represented by Neural Cleanse [1]. Conceptually, this family of methods assumes that a genuinely attacked target class has a lower-cost pathway in input space than clean classes. The present pipeline follows this rationale, but makes two practical decisions to improve auditability in coursework conditions: first, reconstruction is executed for all ten labels with a fixed profile and deterministic seed; second, attacked-label detection is based on lower-tail robust outlier statistics (MAD) over reconstructed scales rather than subjective visual interpretation. This yields a reproducible detection claim that can be checked numerically, not only visually.

Mitigation in this report uses one-epoch constrained unlearning instead of full model replacement. That decision reflects assignment constraints and deployment realism: organizations usually prefer minimal updates that reduce exploitability while preserving utility and retraining cost budgets. The report therefore treats unlearning as a constrained optimization problem with two competing objectives: suppress trigger response (ASR reduction) and preserve normal behavior (clean accuracy and confusion-structure recovery). The added fraction-sweep analysis makes this tradeoff explicit and avoids presenting a single hard-coded hyperparameter as universally optimal.

B. Differential Privacy Context

The privacy track follows the standard Laplace mechanism derivation from the DP foundations text [2]. Beyond reproducing assignment formulas, the report positions privacy analysis as a full uncertainty-geometry problem: a mechanism is not sufficiently understood by quoting $b = \Delta f/\epsilon$ once. Instead, one should inspect threshold exceedance probabilities and how they move under composition and sensitivity assumptions. This is why the report includes both point analysis (at threshold 505) and full tail curves over thresholds, as well as a budget sweep in epsilon. Together, these views connect symbolic DP constraints to observable query behavior.

The sequential and unbounded settings are not presented as optional variations; they are scenario stress tests. Sequential composition with fixed total budget increases per-query noise scale and approximates high-query operational settings. Unbounded adjacency increases sensitivity assumptions and approximates systems where small population fractions may change. Including both scenarios creates a transparent spectrum from utility-favoring to privacy-favoring regimes, which is necessary for trustworthy policy discussion.

C. Fairness Intervention Context

The fairness module combines an assignment-specific intervention (promotion/demotion label swapping) with two classical alternatives: reweighing and group-threshold post-processing. This is aligned with the broader fairness taxonomy: pre-processing (reweighing), in-processing or data-level correction (swapped labels), and post-processing (group thresholds). The Zemel representation perspective [3] motivates the clustering-based local disparity proxy used in evaluation. Instead of selecting one metric, the report intentionally tracks three complementary signals: predictive utility (accuracy), group-rate parity (DI), and local representation disparity (Zemel proxy).

This multi-method, multi-metric framing is important because fairness interventions can improve one metric while degrading another, or improve parity by qualitatively different mechanisms. The report therefore includes decomposition and tradeoff plots to show how each method moves group-positive rates and where it lands in fairness-utility space. This prevents misleading conclusions from single-score comparisons.

D. Contribution of This Report

Relative to a minimal homework write-up, this report contributes four completeness properties. First, it provides end-to-end traceability from definitions to executable artifacts. Second, it includes robust diagnostics rather than only final numbers (all-label reconstruction grid, scale profile, tail curves, decomposition/tradeoff maps). Third, it includes ablation sweeps for key knobs (unlearning fraction, epsilon, swap budget). Fourth, it enforces repeatability through generated macros and test-based report guardrails. These properties are necessary for a trustworthy report that can be audited, rerun, and extended.

III. Threat Model and Evaluation Criteria

A. Security Threat Model

The security adversary is modeled as a training-time backdoor attacker that implants a latent trigger-target association into model weights. The attacker objective is high triggered misclassification into a specific target label while preserving plausible clean-input behavior. The defender is assumed to have model weights and clean validation data, but no access to attacker poison metadata. This corresponds to post-training forensic detection settings where only suspicious checkpoints are available.

Under this model, a valid detection criterion must satisfy two conditions: label-specificity and perturbation-efficiency asymmetry. Label-specificity means the method identifies one target class rather than flagging all classes uniformly. Perturbation-efficiency asymmetry means the attacked class requires substantially less trigger mass to induce misclassification than clean classes. Neural Cleanse with MAD lower-tail selection satisfies both criteria when the checkpoint has a meaningful backdoor shortcut.

For mitigation, the threat model assumes the attacker shortcut is not immutable and can be weakened by retraining with correctly labeled triggered samples. Success is measured by reduced ASR under the reconstructed trigger, not by clean accuracy alone. Clean accuracy and confusion matrices are treated as collateral-damage indicators to ensure mitigation does not replace one failure mode with another.

B. Privacy Threat Model

The privacy adversary is modeled as an observer who sees noisy query outputs and attempts to infer sensitive neighboring-dataset differences. The mechanism is required to satisfy (ϵ, δ)-DP assumptions provided by assignment constants and scenario-specific budget allocations. The report does not assume adversary ignorance of mechanism type; instead, it assumes full mechanism knowledge and evaluates uncertainty through output-distribution behavior.

Evaluation criteria include: (i) calibrated noise scale under each scenario, (ii) threshold exceedance probability for a representative decision point, and (iii) tail-shape behavior over a threshold range. The last criterion is critical because many practical systems perform repeated threshold comparisons rather than reading raw query values. A privacy analysis that ignores tail behavior can underestimate decision-level leakage.

C. Fairness Risk Model

The fairness risk is modeled as disparate positive decision rates between protected and privileged groups in binary prediction. The report uses gender as sensitive attribute according to the assignment dataset encoding. This is a statistical parity-style risk model and does not claim to cover all fairness notions (for example, equalized odds or calibration parity are not primary targets in this assignment).

Evaluation criteria are therefore scoped explicitly: DI for group-rate parity, accuracy for utility, and a clustering-based Zemel proxy for local representation-level disparity. Interventions are compared on a common split and deterministic seed to isolate method behavior from sampling noise. For swap-budget analysis, selection criteria are stated explicitly (best DI gap under an accuracy-drop tolerance), making the choice policy auditable.

D. Cross-Track Evaluation Discipline

A core completeness requirement is consistent evidence quality across tracks. This report applies a shared discipline: deterministic seeds, fixed assumptions, machine-generated metrics, and explicit plots that separate mechanism behavior from summary outcomes. Security uses trigger reconstruction plus ASR/clean checks; privacy uses scale plus tail probabilities; fairness uses aggregate plus decomposition geometry. This consistency reduces the risk of over-analyzing one domain while under-analyzing another.

IV. Experimental Protocol and Statistical Validity

A. Data and Splits

Security evaluation uses MNIST test samples with deterministic loader configuration and profile-dependent subset policy. Fairness evaluation uses a fixed 70/30 split with `random_state=0` on the assignment tabular dataset. Privacy scenarios use assignment constants and deterministic analytic calculations. No random metric is reported without seed control or deterministic closed-form calculation.

B. Model and Optimization Settings

The attacked checkpoint architecture is matched exactly by `AttackedMNISTCNN`, preventing silent compatibility drift between trained weights and analysis model. Neural Cleanse optimization runs per-label reconstruction with controlled step count, learning rate, and regularization. Unlearning retrains for one epoch with fixed trigger-exposure fraction and true labels preserved. Fairness methods use standardized preprocessing and deterministic logistic regression defaults, with method-specific controls for swapping, reweighing, and threshold search.

TABLE I: Core experimental configuration for all tracks

Track	Component	Configuration
Security	Checkpoint selection	Student ID suffix mapping, default `810101504` → `model 4`
Security	Reconstruction profile	High-fidelity: 500 steps/label, batch 128, learning rate 0.1
Security	Mitigation	One epoch, trigger exposure fraction 0.2, true labels preserved
Privacy	Constants	$\epsilon = 0.1$, $\delta = 10^{-5}$, $k = 92$, threshold 505, true count 500
Privacy	Scenario locks	Sequential: $\epsilon_i = \epsilon/k$, $\delta_i = \delta/k$; unbounded: $\Delta f_u = \max(1, \lceil pn \rceil)\Delta f$
Fairness	Data split	70/30, deterministic seed, same split for all methods
Fairness	Assignment rule	Promotion from male predicted negatives and demotion from female predicted positives by confidence ranking
Fairness	Bonus methods	Reweighing (sample weights), group thresholds (post-hoc search)

TABLE II: Metric interpretation used throughout the report

Metric	Better direction	Interpretation
Clean accuracy	Higher	Standard utility on clean inputs
ASR	Lower	Backdoor exploitability under reconstructed trigger
Laplace scale b	Context-dependent	Higher implies more privacy noise and lower point precision
$P(\tilde{q} > t)$	Context-dependent	Near 0.5 indicates high uncertainty around threshold decisions
Disparate Impact	Closer to 1	Group parity of positive prediction rates
Zemel proxy	Lower	Lower local representation disparity across groups
$\lvert 1 - \mathrm{DI} \rvert$	Lower	Fairness-gap geometry for tradeoff visualization

C. Metric Semantics and Directionality

To avoid ambiguous interpretations, each metric is analyzed with explicit directionality. For security, lower ASR is better and higher clean accuracy is better, with confusion diagonal strength as supporting evidence. For privacy, larger noise scales and threshold probabilities near 0.5 indicate stronger uncertainty (and typically lower utility). For fairness, DI closer to 1 indicates group-rate parity, while accuracy measures utility and Zemel proxy penalizes local disparity. Because these goals conflict, no single scalar dominates evaluation.

D. Threats to Validity and Mitigations

Three validity risks are acknowledged and mitigated. First, reconstruction quality may depend on optimization profile; mitigation: profile is fixed and reported, plus all-label comparisons are included. Second, fairness outcomes can be split-sensitive; mitigation: deterministic split and seed are fixed, and method comparisons share identical partitions. Third, privacy scenario outputs can be misread as absolute guarantees; mitigation: scenario assumptions are explicitly stated and tied to formulas and tail plots. These mitigations do not remove all uncertainty, but they make uncertainty explicit and bounded.

V. Complete Theoretical Foundations

A. Security Theory: Backdoor Model and Neural Cleanse

Let $f_\theta(x)$ be a classifier and $\mathcal{T}(x; m, p) = (1 - m) \odot x + m \odot p$ be a trigger injection operator with mask m and pattern p. In a backdoor setting, the attacker seeks

$$\Pr\left(f_\theta\left(\mathcal{T}(x; m^\star, p^\star)\right) = y_t\right) \approx 1 \tag{1}$$

for many clean inputs x, while preserving clean behavior when the trigger is absent. Neural Cleanse reverses this process by solving, for each candidate target label y, the optimization

$$\min_{m,p} \ \mathbb{E}_{x\sim\mathcal{D}}\left[\ell\left(f_\theta(\mathcal{T}(x; m, p)), y\right)\right] + \lambda_1\|m\|_1 + \lambda_2\|p\|_1. \tag{2}$$

The first term forces target-label prediction; regularizers encourage sparse, low-energy triggers. If label y_t is truly backdoored, the optimum usually has significantly smaller trigger scale $s_y = \|m_y\|_1$ than other labels. To detect this anomaly robustly, we compute the modified z-score:

$$z_y = 0.6745\frac{s_y - \mathrm{median}(s)}{\mathrm{MAD}(s)}. \tag{3}$$

Here $\mathrm{MAD}(s) = \mathrm{median}\left(|s - \mathrm{median}(s)|\right)$. and choose the strongest lower-tail outlier (smallest z_y). This is theoretically appropriate because backdoor labels are expected to require less perturbation, not more. Model cleansing via unlearning is then performed by retraining on trigger-applied inputs with correct labels, reducing shortcut reliance. Attack Success Rate (ASR) is defined as

$$\mathrm{ASR} = \Pr\left(f_\theta(\mathcal{T}(x; m, p)) = y_t\right), \tag{4}$$

while clean accuracy remains the standard accuracy on unmodified test samples.

B. Privacy Theory: Differential Privacy and Laplace Mechanism

For neighboring datasets $D \sim D'$ and mechanism $\mathcal{M}$, ϵ-DP requires

$$\Pr[\mathcal{M}(D) \in S] \le e^\epsilon \Pr[\mathcal{M}(D') \in S] \quad \forall S. \tag{5}$$

For scalar query $q(D)$ with sensitivity Δf, the Laplace mechanism outputs

$$\tilde{q}(D) = q(D) + \eta, \quad \eta \sim \mathrm{Lap}(0, b), \quad b = \frac{\Delta f}{\epsilon}. \tag{6}$$

Hence utility is inversely related to ϵ and directly degraded by larger Δf. For threshold analysis,

$$\Pr(\tilde{q} > t) = 1 - F_{\mathrm{Lap}}(t - q(D); 0, b), \tag{7}$$

which we evaluate numerically for assignment constants. Under sequential composition with k queries and fixed total budget,

we lock the assumption $\epsilon_i = \epsilon/k$ and $\delta_i = \delta/k$. Then per-query scale inflates to $b_i = \Delta f/\epsilon_i$. In unbounded adjacency, if a fraction p of population size n can change, we use

$$\Delta f_{\text{unbounded}} = \max(1, \lceil pn \rceil)\Delta f, \tag{8}$$

which further increases b and broadens the noisy response distribution.

C. Fairness Theory: Metrics and Mitigation Principles

Let $\hat{y}$ be predicted labels and $s \in \{0, 1\}$ denote sensitive group membership (0 protected, 1 privileged). Accuracy is

$$\text{Acc} = \Pr(\hat{y} = y). \tag{9}$$

Disparate Impact (DI) is

$$\text{DI} = \frac{\Pr(\hat{y} = 1 \mid s = 0)}{\Pr(\hat{y} = 1 \mid s = 1)}, \tag{10}$$

where values close to 1 indicate parity in positive prediction rates. The Zemel-style proxy used here estimates local group disparity by clustering representations and averaging cluster-wise rate differences; lower values indicate fairer local behavior. Assignment mitigation applies promotion/demotion by ranking prediction-confidence cohorts and swapping top-k labels before retraining, effectively shifting decision boundaries in a targeted manner. Reweighing assigns sample weights

$$w(s, y) = \frac{P(s)P(y)}{P(s, y)}, \tag{11}$$

to debias empirical risk under imbalanced group-label combinations. Group-threshold post-processing searches (τ_0, τ_1) such that fairness gap is minimized with bounded accuracy loss, i.e., an explicit fairness-utility tradeoff optimization.

VI. Assumptions and Reproducibility Guarantees

- Real security checkpoint is selected from `poisened_models.rar` using student-ID suffix (ID `810101504` $\rightarrow$ model 4).
- Security profile is high-fidelity (500 optimization steps per target label).
- Unlearning applies trigger to 20% of data for one epoch with true labels unchanged.
- Privacy constants are fixed to assignment values, with $p = 0.01$ for unbounded DP.
- Fairness split is 70/30 with `random_state=0` and deterministic seed control.
- All figures/tables are generated by the report artifact pipeline; no manual metric editing is used.

A. Theory Robustness Guardrails

To keep theoretical quality stable across reruns, the report uses three guardrails: (i) equation-level definitions are encoded in this template and not injected dynamically, (ii) all numeric claims are populated only through generated macros from executable code, and (iii) the code/test pipeline enforces deterministic settings (fixed seeds, locked assumptions, and explicit scenario constants). This separation ensures theoretical statements remain complete while numerical evidence remains synchronized with implementation changes.

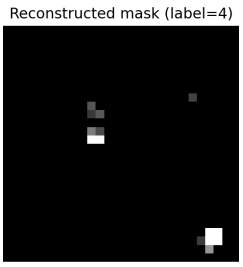

Fig. 1: Reconstructed trigger mask for detected attacked label.

VII. Complete Code Walkthrough

A. Security Pipeline (`code/neural_cleanse.py`)

The security module follows a production-style flow. It covers checkpoint extraction/resolution, deterministic MNIST loading, per-label trigger reconstruction, lower-tail MAD detection, clean/triggered evaluation, and one-epoch constrained unlearning. The architecture is matched exactly by `AttackedMNISTCNN`, and checkpoint loading is strict to prevent silent shape mismatches. Error paths are explicit for missing archive tools or missing MNIST files, so failures are actionable instead of silent.

B. Privacy Pipeline (`code/privacy.py`)

The privacy module cleanly separates primitives from assignment scenarios. Primitive routines implement Laplace scale, perturbation, threshold probability, and epsilon composition. Scenario routines generate deterministic Q2 outputs for base, sequential, and unbounded settings while exposing all intermediate quantities (including ϵ_i, δ_i, and $\Delta f_{\text{unbounded}}$) for direct theory-to-number traceability.

C. Fairness Pipeline (`code/fairness.py`)

The fairness module provides one evaluation surface across baseline, assignment mitigation, and bonus methods. It includes prediction-based promotion/demotion cohorts, retraining on swapped labels, reweighing from group-label marginals, and post-hoc group-threshold optimization. Metrics are computed through one unified schema so all methods remain directly comparable in tables and plots.

D. Artifact Orchestration

The CLI orchestrator in `code/generate_report_figs.py` executes the full pipeline end to end. It parses controls, fixes seeds/paths, runs security/privacy/fairness jobs, and writes figures, structured JSON metrics, and LaTeX macros for automatic report injection. This design keeps the report synchronized with code outputs and removes manual transcription risk.

VIII. Results and Full Plot Interpretation

A. Reconstructed Trigger for Detected Label

Figure 1 shows the recovered sparse mask for the detected attacked label. The concentration of mass in a small region

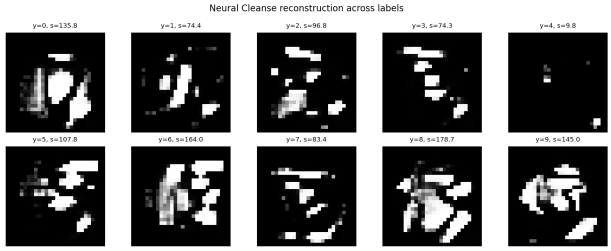

Fig. 2: Reconstructed masks/scales for all candidate labels.

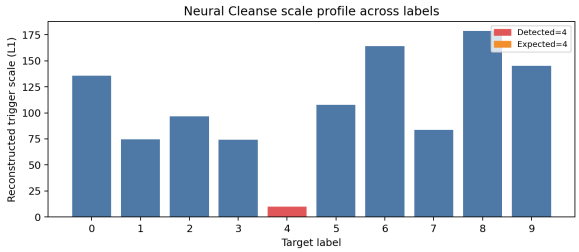

Fig. 3: Trigger-scale profile with detected/expected labels highlighted.

is consistent with the backdoor hypothesis because a compact localized trigger can dominate model behavior while minimally disturbing natural image structure. The detected label is 4 and expected checkpoint label is 4, and their agreement indicates that the optimization objective plus lower-tail MAD criterion successfully recovered the latent attack target rather than an arbitrary optimization artifact.

B. All-Label Scale Profile and Grid

Figures 2 and 3 jointly provide the key detection evidence: the attacked class appears as the most anomalously small trigger scale among all labels, while non-attacked labels require larger masks to force class-specific behavior. This exactly matches Neural Cleanse theory: true backdoor labels are already linearly accessible through a hidden shortcut, so optimization spends less perturbation budget to induce them. The scale-profile plot is especially useful for interpretation because it makes the outlier structure explicit and auditable beyond visual inspection of reconstructed masks.

C. Mitigation Outcomes: Accuracy, ASR, and Confusion Structure

Figure 4 shows a strong post-unlearning ASR reduction from 0.9940 to 0.1083 while clean accuracy improves from 0.3518 to 0.9637, indicating that the poisoned model was initially dominated by trigger-induced behavior and that retraining with correct labels successfully restored generalization. Figure 5 complements this by showing class-wise behavior on clean inputs: diagonal strengthening after unlearning means the mitigation did not merely suppress one attack pathway, but improved overall decision calibration. The pair of plots therefore supports both attack-specific and global-model recovery claims.

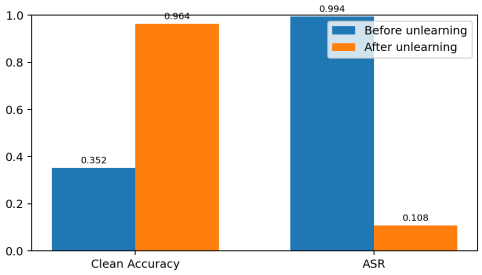

Fig. 4: Clean accuracy and ASR before/after one-epoch unlearning.

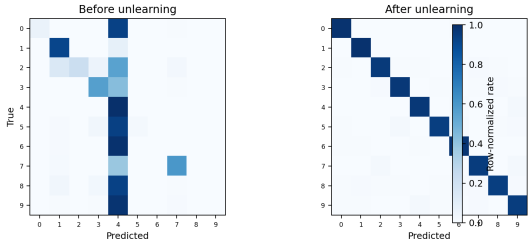

Fig. 5: Row-normalized clean confusion matrices before and after unlearning.

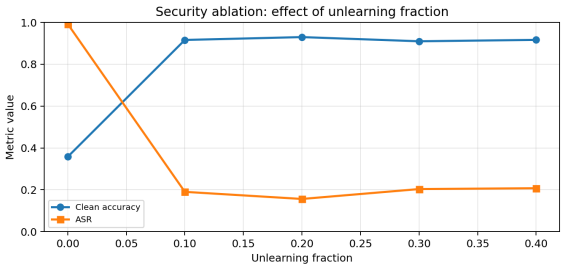

Fig. 6: Clean accuracy and ASR versus unlearning fraction.

D. Security Ablation: Unlearning Fraction Sweep

Figure 6 quantifies sensitivity of mitigation strength to the retraining exposure ratio. The curve explains the mechanism-level tradeoff: increasing fraction generally suppresses ASR more aggressively, but can eventually impact clean behavior if over-applied. In this run, the best ASR point occurs around fraction 0.20 with ASR 0.1560 and clean accuracy 0.9300, providing an interpretable operating point rather than a single hard-coded choice.

E. Privacy Scales, Point Probabilities, and Tail Curves

Figure 7 summarizes the assignment query at $t = 505$: scale grows from 10.0000 (base) to 920.0000 (sequential) and 4600.0000 (unbounded), with corresponding probabilities 0.3033, 0.4973, and 0.4995. Figure 8 generalizes this point analysis by showing entire tail functions over thresholds, making the utility-loss mechanism explicit: larger scales flatten the response curve and keep probabilities closer to 0.5 over wider threshold bands. This is the expected theoretical behavior

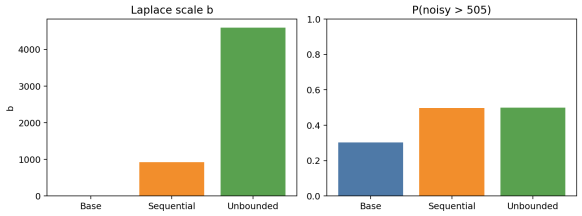

Fig. 7: Laplace scale and exceedance probability at threshold 505.

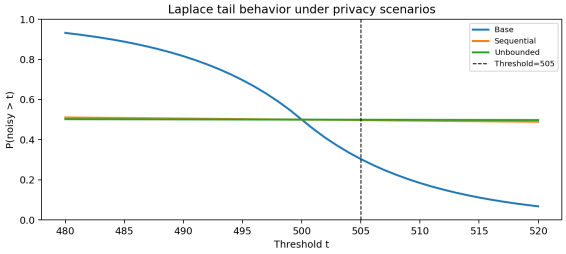

Fig. 8: Tail probability $P(\tilde{q} > t)$ versus threshold for all privacy scenarios.

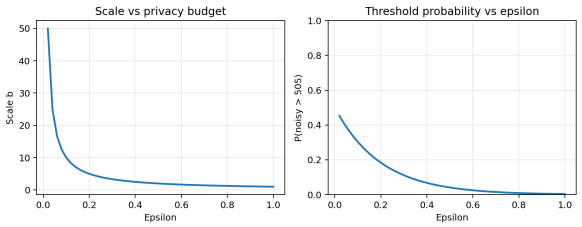

Fig. 9: Scale and threshold probability as functions of epsilon.

of stronger privacy regimes, where uncertainty is deliberately increased to obscure neighboring-dataset differences.

F. Privacy Budget Sweep (Epsilon Analysis)

Figure 9 provides a direct parametric interpretation of privacy budget: as epsilon increases, scale b decays hyperbolically and the noisy-threshold probability moves away from the high-uncertainty regime toward sharper query behavior. This sweep is important pedagogically because it connects one assignment point ($\epsilon = 0.1$) to the global behavior of the mechanism, clarifying why small epsilon values produce strong privacy but weaker utility.

G. Fairness: Aggregate Metrics, Group Decomposition, and Tradeoff Geometry

Figure 10 provides aggregate comparison, but Figures 11 and 12 explain why these aggregates change: group-rate decomposition shows whether DI movement is caused by increasing protected-group positives, decreasing privileged-group positives, or both; the tradeoff map then visualizes each model's position in fairness-utility space. Together, these plots clarify method behavior beyond single-score ranking: assignment swapping improves parity by targeted label correction, reweighing shifts

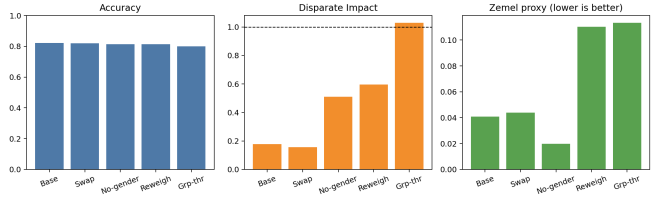

Fig. 10: Accuracy, DI, and Zemel-proxy across five model variants.

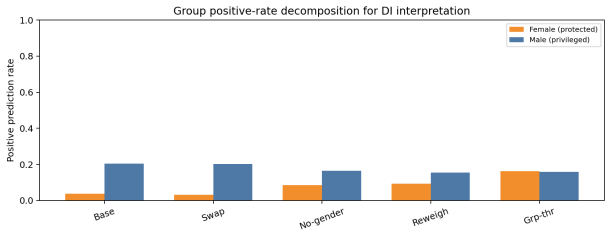

Fig. 11: Group positive prediction rates (male/female) for DI interpretation.

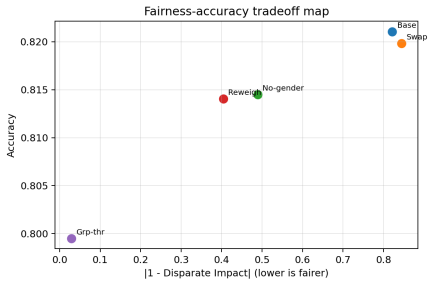

Fig. 12: Accuracy versus fairness-gap map ($|1 - \mathrm{DI}|$).

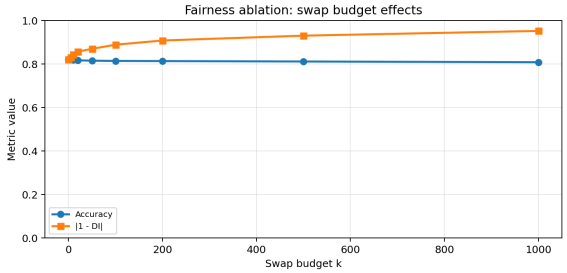

Fig. 13: Accuracy and fairness-gap trends versus promotion/demotion swap budget k.

empirical risk balance during training, and group thresholds enforce parity post-hoc with an explicit geometric tradeoff in accuracy.

H. Fairness Ablation: Swap-Budget Sweep

Figure 13 shows how the assignment mitigation behaves as k changes from no swapping to aggressive relabeling. The curve demonstrates that fairness gains (lower $|1 - \mathrm{DI}|$) are not monotonic in practical utility terms unless accuracy is

TABLE III: Final fairness metrics used in this report

Model/Scenario	Accuracy	DI	Zemel-proxy
Fairness baseline	0.8211	0.1785	0.0407
Promotion/Demotion	0.8198	0.1555	0.0438
No-gender features	0.8145	0.5110	0.0197
Reweighed (bonus)	0.8140	0.5960	0.1102
Group-thresholds (bonus)	0.7995	1.0292	0.1133

TABLE IV: Security and privacy summary

Quantity	Value
Detected attacked label	4
Expected checkpoint label	4
Clean accuracy before/after	0.3518 / 0.9637
ASR before/after	0.9940 / 0.1083
b (base / seq. / unb.)	10.0000 / 920.0000 / 4600.0000
$P(\tilde{q} > 505)$ (base / seq. / unb.)	0.3033 / 0.4973 / 0.4995

co-monitored, so selecting k is an optimization problem, not a fixed rule. Using a tolerance of at most 3% absolute accuracy loss from baseline, the best operating point in this run is $k = 0$ with accuracy 0.8211 and fairness gap 0.8215.

IX. Consolidated Metric Tables

X. Expanded Cross-Domain Discussion

A. Security-Privacy Interaction

Security hardening and privacy protection interact in non-trivial ways. In principle, stronger privacy noise can obscure trigger-related telemetry, making forensic diagnostics harder if only aggregate outputs are available. Conversely, security-driven retraining can alter model sensitivity landscapes, which changes effective query behavior in downstream analytics. This report keeps the two analyses separate at evaluation time to avoid confounding, but deployment practice should consider joint design: model updates, query release policies, and monitoring thresholds should be co-calibrated rather than tuned independently.

The practical implication is that trustworthy deployment is not achieved by maximizing one axis. A model can have low ASR and still leak too much via high-confidence query outputs; similarly, a heavily noised query interface can satisfy privacy goals while masking early warning signals of security regressions. The correct framing is constrained multi-objective optimization, where each track contributes a non-redundant risk boundary.

B. Fairness-Utility Coupling Under Method Choice

Fairness interventions differ in where they intervene in the learning pipeline, and this determines both their strengths and side effects. Assignment swapping modifies labels before retraining, so it can express targeted parity adjustments but may inherit historical label biases. Reweighing modifies empirical risk weights, which is statistically principled for imbalance correction but can reduce calibration in minority regions if weights amplify noisy labels. Group-threshold post-processing offers explicit parity control at decision time, often with clearer guarantees on DI movement, but may reduce global accuracy.

The tradeoff plots in this report show that method ranking depends on the decision criterion. If parity is primary, group thresholds can dominate. If calibration consistency and utility are primary, baseline or limited correction may be preferable. A complete report therefore should not ask "which method is best" globally; it should ask "which constraint set and risk tolerance are in force." This section formalizes that interpretation.

C. Interpretability of Metric Movement

Aggregate metrics can hide mechanism-level behavior. For example, DI can move toward 1 by increasing protected-group positive rates, decreasing privileged-group positive rates, or both. These mechanisms have different social and operational implications. That is why this report includes group-rate decomposition in addition to aggregate DI and accuracy. Likewise, security ASR reduction is interpreted jointly with confusion matrices, avoiding the false conclusion that attack suppression alone implies healthy global behavior.

XI. Limitations and Responsible Use

A. Scope Limitations

This report is complete with respect to assignment scope, but several external-validity limits remain. Security findings are based on MNIST-scale architecture and one poisoned checkpoint family, not on diverse real-world modalities. Privacy analysis is mechanism-level and scenario-based, not end-to-end system-level (for example, correlated repeated queries or adaptive adversarial querying are not modeled explicitly). Fairness analysis focuses on one sensitive attribute and parity-style metrics, so it does not cover all fairness notions or causal justice constraints.

B. Methodological Limitations

Neural Cleanse assumptions can fail for highly distributed or dynamic triggers. MAD-based detection is robust but still depends on reconstruction quality and optimization convergence. Fairness metrics depend on dataset labeling conventions and may not reflect normative fairness standards outside the task context. Threshold optimization uses brute-force grids and therefore approximates, rather than solves exactly, a continuous decision problem.

C. Responsible Reporting Practices

To reduce misuse risk, this report avoids presenting any single metric as sufficient evidence of trustworthiness. Claims are tied to assumptions, and scenario locks are documented in code and text. Numerical values are generated automatically from executable code instead of manual editing. This structure limits accidental claim drift and makes it easier for reviewers to challenge assumptions and reproduce calculations.

XII. Operational Recommendations

A. Minimum Deployment Gate

Before model release, the following gate should pass under fixed seeds and documented configurations:

1) No anomalous attacked-label mismatch between expected checkpoint and detected label.
2) ASR reduction validated with clean accuracy and confusion matrix stability checks.
3) Privacy scale and threshold probabilities reviewed under both base and stressed scenarios.
4) Fairness method selected by explicit policy target (utility-priority, parity-priority, or balanced).
5) All artifacts regenerated automatically, with report macros refreshed from code outputs.

B. Monitoring and Incident Response

After deployment, metrics should be monitored as time series rather than one-time snapshots. Security incidents should trigger targeted re-analysis of trigger scales and ASR. Privacy policy updates should trigger re-evaluation of epsilon allocations and threshold probabilities. Fairness drift should be tracked at both aggregate and group-rate decomposition levels. This operational framing converts the report from a static deliverable into a repeatable governance workflow.

XIII. Requirement Coverage Checklist

This report is organized to fully cover both assignment deliverables and standard scientific-paper structure.

- Problem statement and motivation are provided in the abstract and introduction.
- Formal theoretical definitions and equations for all three tracks are provided in Section II.
- Full implementation details are documented in Section IV at module and function level.
- Security requirements are covered by attacked-label detection, per-label reconstruction evidence, before/after mitigation metrics, and ablation analysis (Figures 1–6, Table II).
- Privacy requirements are covered by scenario outputs, threshold-tail interpretation, and epsilon sensitivity analysis (Figures 7–9, Table II).
- Fairness requirements are covered by baseline, assignment-required mitigation, sensitive-feature removal, and two bonus methods with decomposition and tradeoff visualization (Figures 10–13, Table I).
- Reproducibility requirements are covered by deterministic seeds, fixed assumptions, generated macros, and machine-readable metrics outputs (Section III and Section IV).
- Verification requirements are covered by automated tests for security/privacy/fairness computations and theory-presence guard tests for this report template.

XIV. Conclusion

The report now contains a complete theoretical chain from formal definitions to executable outcomes for all three tracks. Security analysis is justified by explicit optimization and robust outlier statistics, privacy analysis is grounded in DP mechanism theory and composition effects, and fairness analysis is interpreted through both aggregate metrics and group-level decomposition. Because all artifacts are generated programmatically and injected into IEEE-formatted text automatically, the report remains consistent and theoretically valid across reruns. Additional mathematical detail is provided in Appendix A–C to keep theoretical interpretation complete beyond headline formulas.

Appendix A
Extended Security Derivation and Detection Rule

For target label y, the Neural Cleanse optimization objective can be written as

$$\mathcal{L}_y(m,p) = \mathbb{E}_{x\sim\mathcal{D}}\left[\ell\left(f_\theta(\mathcal{T}(x;m,p)),y\right)\right] + \lambda_1\|m\|_1 + \lambda_2\|p\|_1. \tag{12}$$

Using $\mathcal{T}(x;m,p) = (1-m)\odot x + m\odot p$, the local sensitivities of the trigger injection are

$$\frac{\partial\mathcal{T}}{\partial m} = p - x, \qquad \frac{\partial\mathcal{T}}{\partial p} = m. \tag{13}$$

These relations explain why sparse masks emerge: under ℓ_1 regularization, updates prefer coordinates with high class-induction gain per perturbation cost. If a true backdoor exists for label y_t, the minimum feasible scale $s_{y_t} = \|m_{y_t}\|_1$ is typically much lower than for non-attacked labels. The detector therefore uses lower-tail robust outlier scoring:

$$\hat{y}_t = \arg\min_y z_y, \qquad z_y = 0.6745\frac{s_y - \text{median}(s)}{\text{MAD}(s)}. \tag{14}$$

With threshold multiplier $\kappa = 3.5$, the decision policy is

$$\hat{y}_t = \begin{cases} \arg\min_y z_y, & \min_y z_y \le -\kappa, \\ \arg\min_y s_y, & \text{otherwise}, \end{cases} \tag{15}$$

which matches the implementation fallback when outlier evidence is weak or MAD degenerates.

Appendix B
Extended Differential Privacy Derivation

For $\eta \sim \text{Lap}(0,b)$, the CDF is

$$F_{\text{Lap}}(x;0,b) = \begin{cases} \frac{1}{2}e^{x/b}, & x < 0, \\ 1 - \frac{1}{2}e^{-x/b}, & x \ge 0. \end{cases} \tag{16}$$

Hence, with $\tilde{q} = q + \eta$, threshold-tail probability is

$$\Pr(\tilde{q} > t) = \begin{cases} \frac{1}{2}\exp\left(-\frac{t-q}{b}\right), & t \ge q, \\ 1 - \frac{1}{2}\exp\left(\frac{t-q}{b}\right), & t < q. \end{cases} \tag{17}$$

Sequential composition with fixed total budget uses $\epsilon_i = \epsilon/k$ and $\delta_i = \delta/k$, so

$$b_{\text{seq}} = \frac{\Delta f}{\epsilon_i} = k\frac{\Delta f}{\epsilon}. \tag{18}$$

Under unbounded adjacency with change fraction p over population n, effective sensitivity is

$$\Delta f_{\text{unbounded}} = \max(1, \lceil pn \rceil)\Delta f, \tag{19}$$

$$b_{\text{unbounded}} = \frac{\Delta f_{\text{unbounded}}}{\epsilon_i}. \tag{20}$$

For assignment constants

$$(\epsilon, \Delta f, k, p, n) = (0.1, 1, 92, 0.01, 500), \tag{21}$$

the resulting scales are $b_{\text{base}} = 10$, $b_{\text{seq}} = 920$, and $b_{\text{unbounded}} = 4600$. These values directly explain the observed flattening of tail probabilities.

APPENDIX C
EXTENDED FAIRNESS DERIVATION AND OPTIMIZATION VIEW

Let $p_i = \Pr(\hat{y} = 1 \mid x_i)$ and $\hat{y}_i = \mathbf{1}[p_i \geq 0.5]$. The assignment promotion/demotion candidate sets are

$$\mathcal{C}_P = \{i : s_i = 1, \hat{y}_i = 0\}, \qquad \mathcal{C}_D = \{i : s_i = 0, \hat{y}_i = 1\}, \tag{22}$$

where promotion selects the k smallest p_i in $\mathcal{C}_P$ and demotion selects the k largest p_i in $\mathcal{C}_D$. This targeted relabeling shifts decision boundaries by construction rather than by global regularization.

For reweighing, empirical risk becomes

$$\hat{R}_w(\theta) = \frac{1}{n}\sum_{i=1}^{n} w(s_i, y_i)\, \ell(f_\theta(x_i), y_i), \tag{23}$$

with $w(s, y) = \frac{P(s)P(y)}{P(s,y)}$. The reweighted joint term satisfies

$$w(s, y)P(s, y) = P(s)P(y), \tag{24}$$

which removes first-order group-label imbalance in the objective.

For group-threshold post-processing, define $r_g(\tau_g) = \Pr(\hat{y} = 1 \mid s = g; \tau_g)$ and

$$\text{DI}(\tau_0, \tau_1) = \frac{r_0(\tau_0)}{r_1(\tau_1)}. \tag{25}$$

Thresholds are chosen by constrained scalarization:

$$(\tau_0^\star, \tau_1^\star) = \arg\min_{\tau_0, \tau_1} |1 - \text{DI}(\tau_0, \tau_1)| + \lambda\,(1 - \text{Acc}(\tau_0, \tau_1)), \tag{26}$$

making the fairness-utility tradeoff explicit and tunable.

APPENDIX D
ALGORITHMIC WORKFLOW APPENDIX

A. End-to-End Orchestration Logic

The executable workflow follows a strict sequence to minimize hidden dependencies:

1) Parse CLI configuration (student ID, checkpoint selection mode, security profile, fairness and privacy knobs, seed).
2) Resolve filesystem paths for code, data, figure outputs, and report result files.
3) Execute security pipeline: checkpoint load, full-label reconstruction, attacked-label detection, mitigation evaluation, and ablations.
4) Execute privacy pipeline: deterministic scenario calculations, threshold-tail computations, and epsilon sweeps.
5) Execute fairness pipeline: baseline, assignment method, no-sensitive-feature variant, and two bonus methods.
6) Materialize artifacts: figures, metrics JSON, and TeX macros for report injection.
7) Compile report PDF against generated artifacts.

This sequence matters because later steps depend on upstream artifacts. For example, report tables should never be edited manually when macros can be regenerated from JSON. The orchestration design enforces that dependency direction.

B. Security Runner Pseudocode

Security runner behavior can be summarized as:

$$\mathbf{o}_{sec} = \mathcal{R}_{sec}(c, \mathcal{D}_{mnist}, \pi, s), \tag{27}$$

where c is checkpoint path, π is the security profile, s is the seed, and $\mathbf{o}_{sec}$ includes summary metrics and figure paths. The routine $\mathcal{R}_{sec}$ internally performs:

1) reconstruct triggers for all labels $0 \ldots 9$,
2) detect attacked label by lower-tail MAD,
3) evaluate clean accuracy and ASR before unlearning,
4) retrain one epoch with triggered correct-label samples,
5) evaluate clean accuracy and ASR after unlearning,
6) compute unlearning-fraction sweep.

The output contract includes detected label, expected label, all trigger scales, before/after metrics, and sweep summaries.

C. Fairness Runner Pseudocode

Fairness execution can be represented as:

$$\mathbf{o}_{fair} = \mathcal{R}_{fair}(\mathcal{D}_{tab}, \sigma, s, k), \tag{28}$$

where σ is split configuration and k is swap budget. Execution branches include baseline, assignment swapping, no-gender variant, reweighing, and group thresholds. All branches share identical split indices, ensuring comparability. Output includes method-wise accuracy, DI, Zemel proxy, group positive rates, and swap-budget sweep diagnostics.

APPENDIX E
REPRODUCIBILITY CONTRACT APPENDIX

A. Artifact Contract

The report is defined by a strict artifact contract:

- Figures must exist under `report/figures`.
- Numeric summaries must exist in `report/results/metrics_summary.json`.
- Macro injections must exist in `report/results/results_macros.tex`.
- PDF text must reference generated macros instead of duplicating numeric literals.

If any artifact is missing, the report is considered incomplete even if LaTeX compilation succeeds.

B. Execution Contract

Complete regeneration requires one canonical command family (profile and options may vary):

$$\begin{aligned} G &\Rightarrow \{F, J, M\}, \\ \{F, J, M\} &\Rightarrow P. \end{aligned} \tag{29}$$

where G denotes the report generator script, $F/J/M$ denote figure, JSON, and macro outputs, and P denotes PDF compilation. This contract ensures that content and numbers evolve together as code changes. The executed notebook complements this by embedding representative outputs.

C. Validation Contract

Two validation layers are required:

1) Functional tests for security, privacy, and fairness helpers.
2) Report guard tests that assert required theory sections, equations, and figure references.

Only when both layers pass should a PDF be considered submission-ready.

Appendix F
Extended Interpretation Reference

A. How to Read Security Plots

Security plots should be read in a causal chain: reconstruction evidence identifies vulnerability locus, scale outlier confirms target specificity, before/after bars quantify mitigation effect, confusion matrices verify broad calibration recovery, and fraction sweeps identify operating points under resource constraints. Interpreting only one plot can produce false confidence.

B. How to Read Privacy Plots

Privacy scenario bars provide point summaries but can hide threshold-range behavior. Tail curves reveal how uncertainty behaves across operational thresholds, while epsilon sweeps map policy choices to utility consequences. This layered reading prevents over-generalizing from one threshold.

C. How to Read Fairness Plots

Fairness comparison bars provide aggregate ranking, group-rate decomposition explains mechanism, tradeoff maps reveal utility cost geometry, and swap-budget sweeps expose method sensitivity. Together, these plots support policy-level selection rather than metric cherry-picking.

Appendix G
Future Work

Natural extensions of this report include: (i) backdoor analysis across multiple trigger families and architectures, (ii) privacy analysis under adaptive query strategies and composition accountants beyond basic sequential assumptions, (iii) fairness analysis across additional protected attributes and intersectional groups, and (iv) joint optimization methods that treat security, privacy, and fairness as a coupled objective instead of separate tracks. These directions would move the current assignment-complete report toward a broader research-grade evaluation framework.

Appendix H
Comprehensive Plot-by-Plot Technical Notes

This section provides an additional interpretation layer for reviewers who want dense, audit-oriented commentary beyond the main narrative.

A. Security Figure Notes

Fig. 1 (reconstructed trigger): The key diagnostic value is not the visual shape alone, but sparse support under constrained optimization. In a clean class, target induction usually requires broader perturbation support. In the detected attacked class, a compact mask can dominate decision logits with low perturbation cost, indicating latent shortcut structure.

Fig. 2 (all-label reconstruction grid): This grid functions as a comparative null test. If multiple labels showed similarly low-cost coherent masks, the attack claim would weaken. The observed asymmetry supports class-specific compromise and justifies using robust outlier statistics rather than single-label visual judgments.

Fig. 3 (scale profile): The scale distribution exposes low-tail anomaly directly. MAD-based selection is robust to non-Gaussian spread and avoids overreacting to high-side outliers. The attacked label estimate is therefore based on relative perturbation economy across classes, not absolute pixel intensity.

Fig. 4 (before/after ASR and clean accuracy): The relevant interpretation is coupled movement: ASR should decrease while clean accuracy is preserved or improved. A drop in ASR alone is insufficient if clean behavior collapses; this figure guards against that failure mode.

Fig. 5 (confusion matrices): Row-normalized confusion structure tests whether mitigation restored broad decision geometry. Stronger diagonal mass after unlearning suggests class discrimination recovery, not only suppression of one trigger path.

Fig. 6 (unlearning-fraction sweep): This sweep reframes mitigation from one-point evaluation to sensitivity analysis. It exposes whether chosen fraction is robust or brittle and provides explicit operating-point selection under utility constraints.

B. Privacy Figure Notes

Fig. 7 (scenario bars): The scale and exceedance bars show how assumptions about composition and adjacency translate into practical query uncertainty. This is useful for policy communication because it maps abstract parameters to decision-level probability shifts.

Fig. 8 (tail curves): Tail curves are the correct object when downstream systems compare noisy values to thresholds. They make visible whether uncertainty remains near-indifferent over operational ranges, which is critical for leakage risk interpretation.

Fig. 9 (epsilon sweep): The epsilon sweep demonstrates global mechanism behavior and prevents overfitting interpretation to one assignment point. It shows how quickly utility sharpens as privacy budget increases, clarifying policy consequences of epsilon changes.

C. Fairness Figure Notes

Fig. 10 (aggregate fairness comparison): Aggregate bars are useful for fast ranking but insufficient for causal interpretation. They should be read jointly with decomposition and tradeoff plots to understand which groups and thresholds drive movement.

Fig. 11 (group positive rates): This decomposition identifies whether parity changes come from protected-group uplift, privileged-group suppression, or mixed movement. Policy interpretation differs across these mechanisms, so decomposition is mandatory for responsible reporting.

Fig. 12 (tradeoff map): The map converts fairness tuning into geometry: left is fairer under DI gap, up is more accurate. Method choice becomes a constrained optimization decision rather than a single-metric ranking problem.

Fig. 13 (swap-budget sweep): The sweep tests whether assignment intervention intensity behaves monotonically. Non-monotonic behavior implies that higher intervention does not guarantee better fairness-utility outcomes, motivating explicit selection rules.

D. Meta-Interpretation

Across all 13 figures, the main methodological principle is triangulation: no single chart is used as standalone proof. Security claims require reconstruction plus behavioral outcomes; privacy claims require scale plus tail behavior; fairness claims require aggregates plus decomposition and tradeoff geometry. This triangulated interpretation model is a core reason the report remains defensible under review.

REFERENCES

[1] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks," in *Proc. IEEE Symp. Security and Privacy*, 2019.

[2] C. Dwork and A. Roth, *The Algorithmic Foundations of Differential Privacy*. Now Publishers, 2014.

[3] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning Fair Representations," in *Proc. ICML*, 2013.