

# MAL-BAYES-III: Hamiltonian Monte Carlo Posterior Neural-Head Inference for Calibrated Obfuscated Malware Detection in Memory Forensics

William Atuahene Agyei

[waatuahene@aamusted.edu.gh](mailto:waatuahene@aamusted.edu.gh)

Kwame Nkrumah University of Science and Technology

Michael Asante

Kwame Nkrumah University of Science and Technology

Kwabena Owusu-Agyemang

Kwame Nkrumah University of Science and Technology

Kate Takyi

Kwame Nkrumah University of Science and Technology

---

## Research Article

**Keywords:** Hamiltonian Monte Carlo, Bayesian neural networks, volatile-memory forensics, obfuscated malware, uncertainty quantification, selective prediction

**Posted Date:** August 25th, 2026

**DOI:** <https://doi.org/10.21203/rs.3.rs-10668316/v1>

**License:**   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

**Additional Declarations:** No competing interests reported.

---

# MAL-BAYES-III: Hamiltonian Monte Carlo Posterior Neural-Head Inference for Calibrated Obfuscated Malware Detection in Memory Forensics

## Abstract

**Objective:** This study developed MAL-BAYES-III, a partial-Bayesian neural framework that applies Hamiltonian Monte Carlo (HMC) only to a compact final classification head for calibrated obfuscated-malware detection from volatile-memory artefacts. **Methods:** After duplicate and conflicting-label controls, 58,018 CIC-MalMem-2022 observations with 52 predictors were partitioned into training, validation, and locked-test sets. Separate 52-128-64-24 neural encoders were learned for binary and four-category tasks, frozen, and coupled to HMC-sampled Bayesian heads using four chains with 1,200 warm-up and 1,200 retained iterations per chain. **Results:** On the locked test, HMC achieved multiclass accuracy of 0.821527, macro-F1 of 0.730271, and the lowest ECE of 0.007615, although Histogram Gradient Boosting remained more accurate. Predictive entropy identified multiclass errors with AUROC 0.889710, while credible-interval width achieved 0.849388. Binary HMC reached 0.999914 accuracy, 1.000000 malware recall, and ECE 0.000074. Both posterior heads satisfied the declared convergence gates with zero divergences. **Conclusion:** MAL-BAYES-III does not establish universal predictive superiority; it contributes diagnostically validated posterior inference, calibrated probability evidence, and uncertainty-aware selective review for memory-forensic malware classification.

**Keywords:** Hamiltonian Monte Carlo; Bayesian neural networks; volatile-memory forensics; obfuscated malware; uncertainty quantification; selective prediction

## 1. Introduction

Obfuscated malware complicates digital-forensic analysis because packing, encryption, anti-analysis techniques, and memory-resident execution can conceal behaviour that is difficult to recover from static inspection alone. Volatile-memory analysis provides a complementary evidence source by preserving runtime artefacts associated with processes, services, injected code, loaded modules, and other execution activity. CIC-MalMem-2022 was developed to support this form of analysis and has become a widely used benchmark for binary and multiclass obfuscated-malware detection using memory-derived features (Carrier et al., 2022; Dehfouli & Habibi Lashkari, 2025). Studies using the benchmark have reported strong classification performance with conventional machine learning, deep learning, and ensemble methods, confirming the discriminative value of memory evidence for malware detection (Dener et al., 2022; Hossain & Islam, 2024).

Forensic use, however, requires more than a correct class label. A model may discriminate well on a benchmark while providing confidence estimates that are difficult to interpret when observations are ambiguous, shifted, or unlike the training distribution. Recent cybersecurity research has therefore moved toward uncertainty-aware prediction, showing that Bayesian and ensemble-based uncertainty estimates can help distinguish reliable from questionable decisions and can support rejection or escalation under dataset shift (Doan et al., 2023, 2024; Yang et al., 2024; Yumlembam et al., 2025). This is especially relevant to memory-forensic malware analysis, where the practical question is not only whether an observation is predicted as benign or malicious, but also how much evidential confidence should be attached to that prediction and to the inferred malware category.

Hamiltonian Monte Carlo (HMC) provides a principled route to posterior inference because it uses gradient-informed dynamics to explore continuous parameter spaces, but its application to entire neural networks is computationally demanding. Full-batch HMC studies have highlighted both the value and the cost of posterior sampling in modern Bayesian neural networks, motivating designs in which only selected parts of a trained network remain Bayesian (Izmailov et al., 2021; Andrade & Sato, 2025; Thiagarajan et al., 2025). Related cybersecurity work has also demonstrated the feasibility of explicit MCMC-based uncertainty modelling: GIBBON uses Gibbs-based Bayesian refinement for graph-oriented IoT intrusion detection, while

NUTSGRAPH applies No-U-Turn Sampling to a graph-based Bayesian intrusion detector (Osei-Wusu et al., 2026a, 2026b). These approaches establish useful precedents for MCMC-driven cybersecurity inference, but they address graph-structured network intrusion rather than memory-forensic malware classification.

Within the reviewed 2021-2026 literature, an unresolved methodological gap remains at the intersection of memory-forensic malware detection and diagnostically audited HMC posterior inference. Existing CIC-MalMem-2022 studies principally emphasize deterministic or ensemble discrimination, efficiency, interpretability, or temporal generalisation (Roy et al., 2023; Maniriho et al., 2024; Madamidola et al., 2025), whereas uncertainty-aware malware studies have largely examined other data representations, Bayesian approximations, ensembles, or conformal decision mechanisms (Doan et al., 2023, 2024; Yumlembam et al., 2025). Among the reviewed studies, no framework was identified that combines a learned memory-forensic neural representation with standard HMC sampling restricted to a compact Bayesian classification head, evaluates both benign-versus-malware and four-category prediction, and audits posterior convergence, calibration, credible intervals, uncertainty-error association, and selective prediction within one leakage-controlled protocol (Carrier et al., 2022; Maniriho et al., 2024; Doan et al., 2024; Osei-Wusu et al., 2026a, 2026b).

This study addresses that gap through MAL-BAYES-III, a posterior neural-head framework for calibrated obfuscated-malware detection in memory forensics. The contribution introduces a computationally restricted Bayesian neural architecture in which a deterministic encoder learns the nonlinear memory representation and HMC is applied only to the final probabilistic decision head, reducing posterior dimensionality relative to full-network HMC while retaining posterior predictive inference. It also evaluates the same forensic records under complementary binary and four-category tasks within a training-validation-locked-test design that isolates calibration and model-selection decisions from final evaluation. It integrates predictive probability with uncertainty quantities including predictive entropy, mutual information, and posterior credible-interval width, allowing uncertainty to be assessed for error identification and selective prediction rather than reported only descriptively. It couples posterior prediction with explicit HMC convergence diagnostics, comparator models, paired statistical evaluation, and controlled feature-shift analysis, thereby treating numerical reliability, predictive discrimination, calibration, and uncertainty as separate but connected properties of the detector.

## 2. Related Works

Recent memory-forensic malware studies demonstrate that CIC-MalMem-2022 supports strong binary and multiclass discrimination across several modelling strategies. Carrier et al. (2022) established the benchmark using memory-feature engineering, while Dener et al. (2022) and Hossain and Islam (2024) reported strong results with conventional and neural classifiers. Subsequent work has focused on improving efficiency or multiclass performance: MalHyStack combines correlation-based feature reduction with stacked ensemble learning, Shafin et al. (2023) developed compact deep models for resource-constrained settings, and Naeem et al. (2023) transformed process-memory evidence into image representations for deep stacked classification (Roy et al., 2023; Shafin et al., 2023; Naeem et al., 2023). Across these approaches, the primary objective remains discriminative performance, with model outputs evaluated mainly through accuracy-oriented or classwise metrics rather than posterior uncertainty.

A related strand has examined generalisation beyond conventional random-split evaluation. Maniriho et al. (2024) introduced temporal evaluation to memory-analysis malware detection and used deep autoencoders with stacked ensembles to study concept drift, while Madamidola et al. (2025) emphasized lightweight and interpretable detection of malware subtypes that were absent from training. These studies strengthen the evaluation of memory-based detectors by considering temporal change, unseen subtypes, efficiency, and interpretability. Their uncertainty treatment, however, is not based on posterior sampling of a Bayesian neural

classifier, so they do not directly address how parameter uncertainty propagates into individual category decisions or credible intervals.

Uncertainty-aware cybersecurity research provides complementary evidence. Yang et al. (2024) used Bayesian deep learning to quantify uncertainty in anomaly detection, and Doan et al. (2023, 2024) showed that Bayesian formulations can support robustness analysis and uncertainty-based identification of adversarial or distributionally unfamiliar malware. Yumlembam et al. (2025) combined ensemble-derived uncertainty with conformal decision mechanisms under substantial Windows PE dataset shift, demonstrating that uncertainty can be used operationally to reduce incorrect acceptance. These studies move beyond confidence scores, but their data domains and inference mechanisms differ from HMC posterior sampling over a memory-forensic neural decision layer.

The Bayesian-neural literature clarifies why this distinction matters. Full-batch HMC can represent posterior predictive distributions that differ from cheaper approximate methods, but the high dimensionality of neural parameter spaces makes full-network sampling expensive (Izmailov et al., 2021). Partially deterministic Bayesian networks and parameter-restricted HMC have therefore been investigated as computationally more tractable alternatives (Andrade & Sato, 2025; Thiagarajan et al., 2025). At the same time, Bayesian inference should not be equated automatically with robustness; Feng et al. (2024) demonstrated that even HMC-based Bayesian neural networks can remain vulnerable to adversarial perturbations. In cybersecurity specifically, GIBBON and NUTSGRAPH extend explicit MCMC inference to graph-based IoT intrusion detection using Gibbs sampling and NUTS, respectively (Osei-Wusu et al., 2026a, 2026b). Taken together, the reviewed literature establishes strong memory-based classifiers, increasingly mature uncertainty methods, and viable MCMC-based security models, but leaves unresolved whether a compact HMC Bayesian head built on a learned memory representation can deliver diagnostically reliable posterior uncertainty for both binary and malware-category classification under a strictly separated evaluation protocol.

### **3. Methodology & Experimentation**

#### **3.1 Experimental Design, Dataset, Task Definition, and Model Architecture**

MAL-BAYES-III employs a unified tabular neural architecture across both analytical tasks. As illustrated in Figure 1, the same tabular neural backbone was used for both analytical tasks, with only the output dimension changing between multiclass and binary prediction. Each 52-dimensional preprocessed memory-forensic vector entered a deterministic encoder comprising a 128-unit linear layer, LayerNorm, ReLU activation, and dropout of 0.15, followed by an analogous 64-unit block. A final 64-to-24 linear transformation with LayerNorm and ReLU produced the compact latent representation. During deterministic representation learning, this 24-dimensional embedding was connected to a linear task head producing four logits for the category-level task or a single logit for the binary task. The encoder was trained separately for each task; epoch selection used a stratified internal 15% subset of the training data, followed by refitting on the complete training partition for the selected number of epochs.

The MAL-BAYES-III model architecture was defined before the task-specific mathematical formulation and partition calculations. As illustrated in Figure 1, the same tabular neural backbone was used for both analytical tasks, with only the output dimension changing between multiclass and binary prediction. Each 52-dimensional preprocessed memory-forensic vector entered a deterministic encoder comprising a 128-unit linear layer, LayerNorm, ReLU activation, and dropout of 0.15, followed by an analogous 64-unit block. A final 64-to-24 linear transformation with LayerNorm and ReLU produced the compact latent representation. During deterministic representation learning, this 24-dimensional embedding was connected to a linear task head producing four logits for the category-level task or a single logit for the binary task. The encoder was

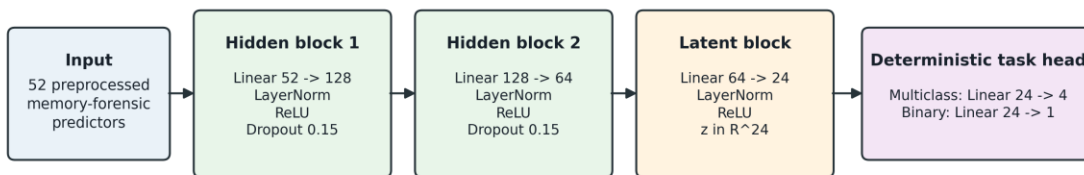
trained separately for each task; epoch selection used a stratified internal 15% subset of the training data, followed by refitting on the complete training partition for the selected number of epochs.

For MAL-BAYES-III posterior inference, the fitted encoder parameters were then frozen and the 24-dimensional embeddings were extracted for the training, validation, and locked-test partitions. A StandardScaler fitted only to the training embeddings defined the latent scale used by HMC. The Bayesian stage did not resample encoder parameters. Instead, it replaced the deterministic decision layer with a compact probabilistic head. For the binary task, HMC sampled a 24-element weight vector and scalar intercept. For the four-category task, a reference-class parameterisation sampled a 24-by-3 weight matrix and three intercepts, with the fourth class represented by a fixed zero reference logit. All free head parameters were assigned zero-centred Normal priors with unit scale. This partial-Bayesian construction preserves the learned nonlinear representation while restricting exact HMC to a tractable decision layer. Figure 1 illustrates the complete pathway from deterministic representation learning to the compact HMC Bayesian decision head.

### MAL-BAYES-III PUBLICATION model architecture

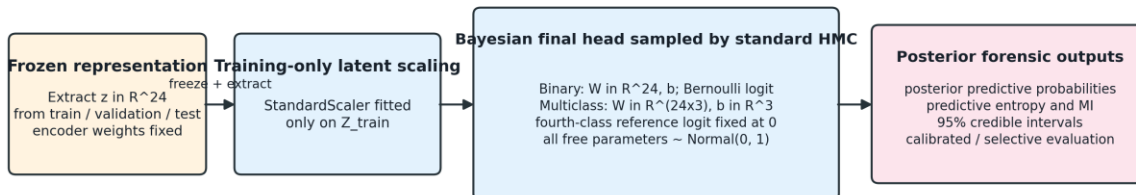
Exact HMC is restricted to the final Bayesian decision head; the nonlinear neural encoder remains deterministic.

#### A. Deterministic task-specific representation learning



Each task trains its own encoder. Epoch selection uses a stratified 15% subset of the training partition, followed by full-training refit.

#### B. Frozen representation and HMC Bayesian final head



The deterministic task head is retained for the temperature-scaled DNN, MC-dropout and deep-ensemble comparisons; it is not part of the HMC parameter state.

**Figure 1. End-to-end MAL-BAYES-III model architecture showing deterministic 52-128-64-24 representation learning, the task-specific deterministic output head, training-only latent scaling, and the compact HMC Bayesian final head.**

*Note. The deterministic head supports the neural baseline, MC-dropout, and deep-ensemble comparators. MAL-BAYES-III HMC conditions on the frozen 24-dimensional representation and samples only the final Bayesian head; encoder parameters are not resampled.*

The mathematical symbols used throughout the experimental formulation are summarized in Table 1 before their detailed use in the subsequent equations and algorithms.

**Table 1. Mathematical Notations and Descriptions Used in MAL-BAYES-III**

| Notation                                                 | Description                                                       | First use |
|----------------------------------------------------------|-------------------------------------------------------------------|-----------|
| <b>Data, task, preprocessing, and partition notation</b> |                                                                   |           |
| $\mathcal{D}$                                            | Complete cleaned analytical dataset.                              | Eq. (1)   |
| $N$                                                      | Number of observations in the relevant dataset or evaluation set. | Eq. (1)   |

| Notation                                                                        | Description                                                                                          | First use        |
|---------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|------------------|
| $\mathbf{x}_i, x_{ij}$                                                          | Predictor vector for observation i and its j-th feature value.                                       | Section 3.1      |
| $y_i, y_i^{(4)}, y_i^{(2)}$                                                     | Observed label; four-category target; binary Benign-versus-Malware target.                           | Eqs. (2)–(3)     |
| $\mathcal{C}$                                                                   | Ordered four-category class set.                                                                     | Section 3.1      |
| $\mathbb{I}(\cdot)$                                                             | Indicator function, equal to 1 when its condition is true and 0 otherwise.                           | Eq. (3)          |
| $\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{test}}$  | Mutually exclusive training, validation, and locked-test partitions.                                 | Eq. (4)          |
| $\tilde{x}_{ij}, m_j^{(\text{tr})}$                                             | Median-imputed feature value and training-derived median for feature j.                              | Eq. (5)          |
| $x_{ij}^{(\text{std})}$                                                         | Original predictor standardized using training-derived location and scale.                           | Eq. (6)          |
| $\mu_j^{(\text{tr})}, \sigma_j^{(\text{tr})}$                                   | Training-derived mean and standard deviation for feature j.                                          | Eq. (6)          |
| $\mathbf{H}_a, \mathbf{H}_b$                                                    | Exact predictor-vector hash sets for two data partitions.                                            | Eq. (7)          |
| $\mathcal{T}_a, d_{\text{NN}}$                                                  | Sampled training reference set and nearest-neighbour distance used for similarity auditing.          | Eq. (8)          |
| $g_i, G_{\text{tr}}$                                                            | Group label of observation i and the set of groups represented in training.                          | Eq. (9)          |
| <b>Neural representation and Bayesian classification-head notation</b>          |                                                                                                      |                  |
| $f_\phi(\cdot), \phi$                                                           | Task-specific deterministic encoder and its fitted parameters.                                       | Eq. (10)         |
| $\mathbf{z}_i, \mathbf{z}'_i$                                                   | 24-dimensional latent embedding and its training-standardized form.                                  | Eqs. (10), (12)  |
| $\eta_i, \eta_{ik}$                                                             | Binary linear predictor and multiclass category-specific logit.                                      | Eqs. (13)–(15)   |
| $\mathbf{w}, b, \mathbf{w}_k, b_k$                                              | Binary and multiclass Bayesian-head weights and intercepts.                                          | Eqs. (13)–(16)   |
| $\sigma(\cdot), K$                                                              | Logistic sigmoid function and number of analytical classes.                                          | Eqs. (13), (15)  |
| $\theta, D_H = (Z_H, y_H)$                                                      | Free Bayesian-head parameter vector and training-only posterior-fitting subset.                      | Eq. (17)         |
| <b>Hamiltonian Monte Carlo notation</b>                                         |                                                                                                      |                  |
| $U(\theta)$                                                                     | Potential energy, defined as the negative log posterior.                                             | Eq. (18)         |
| $\mathbf{r}, \mathbf{M}$                                                        | Auxiliary HMC momentum vector and diagonal mass matrix.                                              | Eq. (19)         |
| $K_{\text{HMC}}(\mathbf{r})$                                                    | HMC kinetic-energy function.                                                                         | Eq. (19)         |
| $H(\theta, \mathbf{r})$                                                         | Hamiltonian: sum of potential and kinetic energy.                                                    | Eq. (20)         |
| $\epsilon, \ell$                                                                | Leapfrog step size and leapfrog-step index.                                                          | Eqs. (22a)–(22c) |
| $\Delta H, \alpha$                                                              | Hamiltonian change and Metropolis acceptance probability.                                            | Eq. (23)         |
| <b>Posterior predictive and forensic-uncertainty notation</b>                   |                                                                                                      |                  |
| $S, p_{ik}^{(s)}, \bar{p}_{ik}$                                                 | Number of predictive posterior draws; draw-wise class probability; posterior mean class probability. | Eq. (25)         |
| $\hat{y}_i$                                                                     | Final class prediction obtained from the largest mean class probability.                             | Eq. (26)         |
| $H_i^{\text{pred}}, H_i^{\text{exp}}$                                           | Predictive entropy and expected entropy.                                                             | Eqs. (27)–(28)   |
| $\text{MI}_i$                                                                   | Mutual information, measuring posterior parameter-sensitive predictive disagreement.                 | Eq. (29)         |
| $L_i, U_i, W_i$                                                                 | Lower and upper 95% predicted-class probability limits and interval width.                           | Eqs. (30a)–(30b) |
| $Q_q(\cdot)$                                                                    | Empirical q-th quantile operator.                                                                    | Eq. (30a)        |
| <b>Calibration, performance, diagnostics, and selective-prediction notation</b> |                                                                                                      |                  |
| $T, T^*, \ell_i$                                                                | Temperature parameter, validation-selected temperature, and deterministic logit vector.              | Eqs. (31)–(32)   |
| $TP_k, FP_k, FN_k, n_k$                                                         | Class-specific confusion counts and class support.                                                   | Eqs. (34)–(36b)  |
| $B_{\text{bin}}, S_b, a_b, c_b$                                                 | Number of calibration bins (15), observations in bin b, bin accuracy, and bin confidence.            | Eqs. (39)–(40)   |

| Notation                                              | Description                                                                                             | First use        |
|-------------------------------------------------------|---------------------------------------------------------------------------------------------------------|------------------|
| $\hat{R}_j, ESS_{\text{bulk},j}, ESS_{\text{tail},j}$ | Rank-normalized split R-hat and bulk/tail effective sample sizes for parameter $j$ .                    | Eqs. (41)–(42b)  |
| $MCSE(\theta_j)$                                      | Monte Carlo standard error for sampled parameter $j$ .                                                  | Eq. (43)         |
| $C_H, n_{\text{ret}}, a_{cs}, d_{cs}$                 | Number of HMC chains, retained iterations per chain, acceptance probability, and divergence indicator.  | Eqs. (44a)–(44b) |
| $e_i, u_i, \rho_s(u, e), A_{\text{err}}$              | Prediction-error indicator, generic uncertainty score, Spearman association, and error-detection AUROC. | Eqs. (46)–(48)   |
| $\pi, c, m_c, R_c$                                    | Uncertainty-ordering permutation, requested coverage, retained sample size, and retained set.           | Eqs. (49a)–(50)  |
| $\text{Risk}(c)$                                      | Selective misclassification risk at coverage $c$ .                                                      | Eq. (51)         |
| <b>Bootstrap and controlled-shift notation</b>        |                                                                                                         |                  |
| $B_{\text{boot}}, I^{(b)}$                            | Number of paired-bootstrap repetitions (1,000) and resampled locked-test index vector.                  | Eqs. (52)–(55)   |
| $\Delta^{(b)}, \bar{\Delta}$                          | Paired macro-F1 difference (HMC minus DNN) in bootstrap repetition $b$ and its mean.                    | Eqs. (53)–(54a)  |
| $\hat{P}(\Delta > 0)$                                 | Directional bootstrap proportion favouring HMC; not a p-value.                                          | Eq. (55)         |
| $\mathcal{S}, s$                                      | Controlled-shift severity set and an individual severity level.                                         | Section 3.7      |
| $\varepsilon_{ij}, X_{s,ij}^{(G)}$                    | Gaussian perturbation and resulting perturbed standardized feature value.                               | Eq. (56)         |
| $\zeta_{ij}, X_{s,ij}^{(\text{mask})}$                | Bernoulli masking indicator and resulting masked standardized feature value.                            | Eq. (57)         |

Following the architectural specification, the MAL-BAYES-III experiment was designed as a leakage-controlled, two-task evaluation of posterior neural-head inference on memory-forensic data. CIC-MalMem-2022 was used as the sole modelling dataset for the experiment. The benchmark was introduced by Carrier et al. (2022) for obfuscated-malware detection from volatile-memory artefacts and contains 58,596 observations described by 55 extracted memory features before study-specific data controls. The implemented data controls reconstructed the analytical target from label-like fields, retained numerical memory-forensic predictors, resolved exact duplicate feature vectors before partitioning, and produced a final modelling corpus of 58,018 label-consistent observations with 52 predictors. The cleaned corpus comprised 29,231 Benign, 9,516 Ransomware, 9,795 Spyware, and 9,476 Trojan Horse observations.

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N, N = 58,018 \quad (1)$$

With the model architecture established, each observation was represented by a predictor vector  $x_i$  and an associated analytical label  $y_i$ . The same cleaned predictor matrix supported two related but distinct classification tasks. The primary four-category task preserved the ordered class set Benign, Ransomware, Spyware, and Trojan Horse, thereby evaluating whether posterior neural-head inference could discriminate category-level malware behaviour rather than only maliciousness. A second binary task collapsed the three malware categories into a single Malware state while retaining Benign as the negative class. Maintaining both tasks allowed the experiment to distinguish the comparatively easier benign-versus-malware decision from the more demanding category-level decision without changing the underlying memory-forensic observations.

$$y_i^{(4)} \in \mathcal{C} = \{0, 1, 2, 3\} \quad (2)$$

$$y_i^{(2)} = \mathbb{1}[y_i^{(4)} \neq 0] \quad (3)$$

For reproducibility, the experiment fixed the category order as Benign, Ransomware, Spyware, and Trojan Horse and used a global random seed of 2026. Category codes were therefore deterministic across preprocessing, neural representation learning, Bayesian-head fitting, posterior prediction, and evaluation. The four-category and binary targets are formally defined in Equations (2) and (3), respectively. Because the

binary target was deterministically derived from the four-category label after cleaning, the two tasks shared identical observation identities and partition boundaries; only the response representation differed.

The cleaned corpus was divided once by category-stratified sampling into mutually exclusive training, validation, and locked-test partitions. A stratified 20% test partition was first separated using seed 2026, after which 20% of the remaining observations was assigned to validation using seed 2027. This produced overall allocations of 64% training, 16% validation, and 20% locked testing: 37,131 training observations, 9,283 validation observations, and 11,604 locked-test observations. All four categories were represented in every subset. The training partition contained 18,707 Benign, 6,090 Ransomware, 6,269 Spyware, and 6,065 Trojan Horse records; validation contained 4,677, 1,523, 1,567, and 1,516, respectively; and the locked test contained 5,847, 1,903, 1,959, and 1,895. Table 2 reports the resulting training, validation, and locked-test distributions, including the representation of Benign, Ransomware, Spyware, and Trojan Horse observations in each partition. These counts are outputs of the implemented data-control procedure rather than assumed design targets.

$$\mathcal{D} = \mathcal{D}_{tr} \sqcup \mathcal{D}_{val} \sqcup \mathcal{D}_{test} \quad (4)$$

The three partitions had deliberately different roles. Training data alone were eligible to estimate preprocessing parameters, fit deterministic representations and comparators, and supply observations for posterior-head estimation. Validation data were isolated from parameter fitting except where explicitly required for configuration decisions, including temperature scaling and uncertainty-related selection rules. The locked test was withheld until these choices were frozen and was used only for final predictive, calibration, uncertainty, bootstrap, and controlled-shift evaluation. This separation follows the reproducibility principle that information from evaluation data must not influence model construction; data leakage can otherwise create systematically optimistic estimates of generalisation performance (Kapoor & Narayanan, 2023).

The experiment additionally treated partition identity as an auditable object rather than relying solely on stratification. Exact predictor-vector hashes were checked after splitting to ensure zero overlap between training, validation, and locked-test sets; near-duplicate distances and optional variant-group overlap were also treated as explicit integrity-audit quantities. The locked-test design therefore served two purposes: it provided an untouched basis for final model comparison and enforced a clear temporal order in which data cleaning and partitioning preceded preprocessing, calibration, posterior inference, uncertainty evaluation, and stress analysis. Algorithm 1 summarizes this experiment-wide construction and establishes the data contract used by all subsequent MAL-BAYES-III components.

---

#### Algorithm 1. Data Preparation, Task Construction, Partitioning, and Leakage Audit

**Input:** raw CIC-MalMem-2022 table  $\mathcal{D}_0$ ; fixed class order  $C$ ; random seed  $s = 2026$ ; split ratios  $(0.64, 0.16, 0.20)$ .

**Output:** mutually exclusive training, validation, and locked-test partitions; four-category and binary targets; partition-integrity audit.

1. Identify label-like fields and reconstruct the canonical four-category target  $C = \{\text{Benign, Ransomware, Spyware, Trojan Horse}\}$ ; exclude observations that cannot be mapped unambiguously.
  2. Retain usable numerical memory-forensic predictors after excluding label-like fields and non-informative variables.
  3. Hash complete predictor vectors before partitioning; remove every exact duplicate group containing conflicting category labels and retain one representative from same-category duplicate groups.
  4. Require all four analytical categories after cleaning and encode  $y^{(4)}$  using the fixed class order Benign = 0, Ransomware = 1, Spyware = 2, and Trojan Horse = 3.
  5. Construct the binary target as  $y^{(2)} = \mathbb{1}[y^{(4)} \neq 0]$ , so Benign = 0 and all malware categories = 1.
  6. Using seed  $s$ , draw a stratified 20% locked-test partition; from the remaining 80%, use seed  $s + 1$  to draw a stratified
-

---

20% validation partition, yielding overall proportions of 64% training, 16% validation, and 20% locked test.

7. Verify that exact predictor-vector hashes do not overlap across the three partitions; terminate the experiment if any overlap is detected.
8. Freeze the locked-test partition. Permit training data to fit model parameters and preprocessing objects; reserve validation data for configuration and calibration decisions; release the locked test only after all choices are fixed.

**Return** Dtr, Dval, Dtest,  $y^{(4)}$ ,  $y^{(2)}$ , and the integrity audit.

---

Here,  $\sqcup$  denotes a disjoint partition. Algorithm 1 therefore defines the experiment-wide data contract, task construction, split logic, and integrity conditions used throughout MAL-BAYES-III.

### 3.2 Data Preprocessing and Partition-Integrity Controls

Preprocessing was implemented as a sequence of feature-admissibility, missing-value, scale, and partition-integrity controls rather than as a single transformation of the full dataset. Before partition-specific transformations were estimated, label-like fields were excluded from the predictor space and each remaining field was coercively tested for numeric usability. A candidate predictor was retained when at least 80% of its mapped values could be represented numerically; positive and negative infinity were converted to missing values, after which all-missing and constant columns were removed. These target-independent admissibility checks produced the 52 predictors used by both analytical tasks. Distribution-estimating transformations were then restricted to the training partition. This separation is important because preprocessing information drawn from evaluation records can leak test-set structure into model construction and yield optimistic estimates of generalisation (Kapoor & Narayanan, 2023; Bouke & Abdullah, 2023).

Missing values were treated with feature-wise median imputation. For predictor  $j$ , the imputation constant was estimated only from the training observations and then reused unchanged for validation and locked-test records. Let  $m_j^{(tr)}$  denote the training median of feature  $j$ . The imputed value  $\tilde{x}_{ij}$  was defined as follows:

$$\tilde{x}_{ij} = \text{Imp}(x_{ij}; m_j^{(tr)}) \quad (5)$$

After imputation, standardization was fitted to the imputed training matrix only. If  $\mu_j^{(tr)}$  and  $\sigma_j^{(tr)}$  denote the corresponding training mean and standard deviation, every training, validation, and locked-test observation was mapped to the standardized form defined in Equation (6). The fitted SimpleImputer–StandardScaler pipeline was serialized with the publication run, ensuring that the same transformation parameters were applied whenever the frozen data partitions were reproduced.

$$x_{ij}^{(std)} = \frac{\tilde{x}_{ij} - \mu_j^{(tr)}}{\sigma_j^{(tr)}} \quad (6)$$

Duplicate control was enforced at two levels. Before splitting, complete numerical predictor vectors were hashed so that conflicting-label duplicate groups could be removed in their entirety and repeated vectors with the same category could be reduced to one representative. This choice avoids allowing repeated feature patterns to dominate different partitions; sample duplication has been shown to distort malware-evaluation conclusions and should therefore be audited explicitly (Zhao et al., 2021). After the split, hashes of the retained untransformed predictor vectors were recomputed for each partition. The publication pipeline required the pairwise intersections in Equation (7) to be empty and raised a runtime error if any exact cross-partition duplicate remained.

$$\mathbf{H}_a \cap \mathbf{H}_b = \emptyset, \quad a \neq b \quad (7)$$

Exact equality does not reveal whether highly similar records remain across partitions, so a separate near-duplicate diagnostic was performed in the standardized feature space. The audit used a seed-controlled sample

of at most 2,000 training observations and independently sampled up to 2,000 validation and 2,000 locked-test observations. A one-nearest-neighbour model with Euclidean distance was fitted to the sampled training vectors. For a query vector  $q$  and sampled training reference set  $\mathcal{T}_a$ , the diagnostic distance was defined by Equation (8). For each evaluation partition, the minimum, median, and first-percentile nearest-neighbour distances were retained. Table 3 reports these cross-partition distance summaries. The procedure was diagnostic rather than corrective: near neighbours were not deleted solely because their Euclidean distance was small.

$$d_{\text{NN}}(q, \mathcal{T}_a) = \min_{t \in \mathcal{T}_a} d_E(q, t) \quad (8)$$

The implementation also performed an optional group-overlap audit to determine whether validation or locked-test records belonged to groups absent from training. Candidate grouping variables were drawn from label-like fields having between 5 and 200 observed levels; the eligible field with the greatest number of distinct values was selected. In the publication run this rule selected the Category field, which represented 16 groups. If  $G_{\text{tr}}$  denotes the set of group values represented in training, the unseen-group indicator for observation  $i$  was defined by Equation (9). Counts and fractions of unseen groups, together with the number of unique groups in each evaluation partition, were recorded; Table 4 reports the resulting variant-group audit. This check should be interpreted as an overlap diagnostic, not as evidence of group-held-out or genuinely unseen-malware evaluation, because the primary split itself remained category-stratified rather than group-stratified.

$$u_i^{(g)} = \mathbb{1}[g_i \notin G_{\text{tr}}] \quad (9)$$

Taken together, these controls maintained a single fitted preprocessing state for both binary and four-category experiments while preserving the functional separation of training, validation, and locked-test data. The same preprocessed observation identities were therefore carried forward to deterministic representation learning and Bayesian-head inference, and every integrity check was exported as an auditable publication artefact. Algorithm 2 formalizes the complete preprocessing and partition-integrity procedure.

---

### Algorithm 2. Leakage-Controlled Preprocessing and Partition-Integrity Verification

**Input:** cleaned predictor matrix  $X$ ; training, validation, and locked-test indices; label-like columns  $L$ ; optional group labels  $g$ ; random seed  $s = 2026$ .

**Output:** standardized training, validation, and locked-test matrices; fitted preprocessing object; exact-overlap, near-duplicate, and variant-group audit records.

1. Exclude label-like fields. Convert every remaining candidate column to numeric form; retain a column only when at least 80% of its mapped values are numerically convertible.
  2. Replace positive and negative infinity with missing values, remove all-missing columns, and remove predictors with no observed variation; retain the resulting admissible predictor set.
  3. Using training observations only, estimate the median  $m_j^{(\text{tr})}$  for each retained predictor and replace missing values in training, validation, and locked-test records with the corresponding training median.
  4. Using the imputed training partition only, estimate  $\mu_j^{(\text{tr})}$  and  $\sigma_j^{(\text{tr})}$ ; transform all three partitions with the same training-derived standardization parameters.
  5. Persist the fitted median-imputation and standardization pipeline so the transformation applied during evaluation is exactly reproducible.
  6. Hash each complete cleaned predictor vector using the pre-split representation and form partition-specific hash sets. Require all pairwise cross-partition hash intersections to be empty; terminate if any exact cross-partition overlap exists.
  7. Let  $n_a$  be the smaller of 2,000 and the available partition sizes. Draw a seed-controlled training audit sample of size  $n_a$
-

---

and, separately, up to  $n_a$  validation and locked-test query records.

8. For each validation/test query vector  $q$ , compute its Euclidean distance to the nearest sampled training vector; record the minimum, median, and first-percentile nearest-neighbour distances for each comparison.
9. When an eligible group field is available, define the training group set  $G_{tr}$  and mark every validation/test record whose group label is absent from  $G_{tr}$ ; record unseen-row counts, fractions, and the number of represented groups.
10. Export the preprocessing object and all integrity-audit summaries without altering the frozen validation or locked-test partitions.

Return the standardized partition matrices, fitted preprocessing pipeline, and the three audit records.

---

Algorithm 2 defines the preprocessing contract used throughout the publication experiment: evaluation partitions can receive transformations estimated from training data, but they cannot contribute distributional statistics to those transformations; exact cross-partition duplication is prohibited; and similarity/group structure is retained as an explicit audit rather than silently ignored.

### 3.3 Deterministic Neural Representation Learning and Bayesian Classification Head

Figure 1 provides the architectural context for the representation-learning and Bayesian-head stages; this subsection formalizes how those components were fitted without repeating the architectural description. Separate deterministic encoders were trained for the binary and four-category tasks. For an input vector  $x_i \in \mathbb{R}^{52}$ , the frozen encoder was treated as a nonlinear mapping  $f_\phi$  that produced a 24-dimensional latent vector  $z_i$ . During deterministic training, the two hidden blocks applied LayerNorm, ReLU activation, and dropout with probability 0.15, whereas the latent block applied LayerNorm and ReLU without dropout. The resulting representation can therefore be written compactly as Equation (10).

$$z_i = f_\phi(x_i) \in \mathbb{R}^{24} \quad (10)$$

Network fitting was performed only with the outer training partition. A stratified 15% internal subset was separated from that partition for epoch selection, leaving the outer validation and locked-test data untouched. The final configuration allowed at most 80 epochs, used mini-batches of 512 observations, an AdamW learning rate of  $6 \times 10^{-4}$ , weight decay of  $10^{-4}$ , and gradient-norm clipping at 5.0. Class imbalance was handled inside the loss function rather than by resampling. For the binary task, the positive-class weight was the ratio of benign to malware observations in the optimization subset. For the four-category task, inverse-frequency class weights were normalized to have unit mean. These weights were used with binary cross-entropy with logits and multiclass cross-entropy, respectively.

Epoch selection was based on negative log-likelihood computed on the internal training-only selection subset. A new best state was retained only when the monitored NLL decreased by more than  $10^{-5}$ ; training stopped after 10 consecutive epochs without such improvement or when the 80-epoch limit was reached. If  $\mathcal{L}_{NLL}(e)$  denotes the selection-set NLL after epoch  $e$ , the retained training duration is represented by Equation (11).

$$e_{\text{best}} = \arg \min_{1 \leq e \leq 80} \mathcal{L}_{NLL}(e), \quad P = 10 \quad (11)$$

The network used for epoch selection was not reused as the final encoder. Instead, a fresh instance of the same architecture was initialized and refitted on the complete outer training partition for the selected number of epochs. This refit step ensured that the training-only observations temporarily reserved for epoch selection contributed to the final deterministic representation without exposing the outer validation or locked test to parameter estimation. After refitting, the encoder parameters  $\phi$  were frozen, the deterministic task head was no longer part of the MAL-BAYES-III posterior state, and embeddings were extracted for all three outer partitions with gradients disabled.

Because HMC operates most reliably when continuous predictors are on comparable scales, the latent representation underwent a second training-only standardization after extraction. Let  $\mu_j^{(z)}$  and  $\sigma_j^{(z)}$  denote the mean and standard deviation of latent dimension  $j$  estimated from the training embeddings alone. The standardized representation  $z_i'$  used by the Bayesian head was defined by Equation (12), and the same fitted transformation was applied unchanged to validation and locked-test embeddings.

$$z'_{ij} = \frac{z_{ij} - \mu_j^{(z)}}{\sigma_j^{(z)}} \quad (12)$$

The Bayesian head was then constructed directly on  $z_i'$  rather than on the original 52 predictors. This restriction is consistent with partially Bayesian neural designs in which a deterministic feature extractor is retained while posterior inference is concentrated on the decision layer, reducing posterior dimensionality relative to full-network HMC (Andrade & Sato, 2025; Thiagarajan et al., 2025). For binary detection, the latent vector entered a Bayesian logistic head with weight vector  $\mathbf{w} \in \mathbb{R}^{24}$  and scalar intercept  $b$ . Its linear predictor and malware probability are given in Equation (13).

$$\eta_i = \mathbf{w}^\top \mathbf{z}'_i + b, \quad p_i = \sigma(\eta_i) \quad (13)$$

For the four-category task, the implementation used a reference-class parameterization to remove the common softmax translation non-identifiability. Three category-specific coefficient vectors and intercepts were sampled, while the fourth category was assigned a fixed zero logit. With  $K = 4$ , the free logits were defined by Equation (14), followed by the softmax probabilities in Equation (15).

$$\eta_{ik} = \mathbf{w}_k^\top \mathbf{z}'_i + b_k \quad (k = 1, 2, 3), \quad \eta_{i4} = 0 \quad (14)$$

$$p(y_i = k \mid \mathbf{z}'_i, \mathbf{W}, \mathbf{b}) = \frac{\exp(\eta_{ik})}{\sum_{c=1}^4 \exp(\eta_{ic})} \quad (15)$$

All free Bayesian-head parameters were assigned independent zero-centred Normal priors with unit scale, as summarized in Equation (16). The prior therefore regularized the compact decision layer while leaving the frozen encoder outside the posterior parameter space. Consequently, the resulting credible intervals and epistemic-uncertainty measures describe uncertainty in the Bayesian decision head conditional on the learned neural representation; they should not be interpreted as full-network Bayesian uncertainty (Izmailov et al., 2021; Andrade & Sato, 2025).

$$\theta_j \stackrel{\text{iid}}{\sim} \mathcal{N}(0, 1), \quad \theta_j \in \boldsymbol{\theta}_{\text{head}} \quad (16)$$

Posterior fitting was deliberately restricted to a stratified subset of the outer training embeddings so that standard HMC remained computationally feasible without using validation or locked-test observations. The final configuration capped each analytical class at 3,000 training cases. Because every category exceeded that cap, the four-category Bayesian head used 12,000 posterior-fitting records; after the three malware categories were collapsed for the binary task, the binary head used 6,000 records. The selection was performed independently within each task and preserved the frozen outer data contract. Algorithm 3 summarizes the complete representation-learning and Bayesian-head construction procedure.

---

### Algorithm 3. Deterministic Neural Representation Learning and Bayesian-Head Construction

Input: standardized training matrix; binary or four-category target  $y$ ; fixed 52-128-64-24 encoder architecture; maximum epochs  $E_{\text{max}} = 80$ ; patience  $P = 10$ ; batch size  $B = 512$ ; learning rate  $\eta = 6 \times 10^{-4}$ .

---

---

Output: task-specific frozen encoder  $\phi$ ; selected training duration; standardized 24-dimensional embeddings; binary or reference-class Bayesian head; HMC training subset.

1. Stratify the training partition into an 85% optimization subset and a 15% epoch-selection subset using only training observations.
2. Initialize the 52-128-64-24 encoder and task-specific deterministic output head; compute class weights from the optimization labels.
3. For each epoch, update the network with AdamW (weight decay  $10^{-4}$ ), the task-appropriate weighted loss, mini-batches of 512, and gradient-norm clipping at 5.0.
4. Evaluate negative log-likelihood on the internal selection subset; retain a new best state only when the NLL improves by more than  $10^{-5}$ , and stop after 10 consecutive non-improving epochs or after 80 epochs.
5. Set the training duration to the best observed epoch and initialize a fresh network with the same architecture; refit it on the complete outer training partition for that number of epochs.
6. Freeze the refitted encoder and extract 24-dimensional embeddings for the training, validation, and locked-test partitions without gradient updates.
7. Fit a StandardScaler to the training embeddings only and apply the same latent transformation to all three partitions.
8. For the binary task, define a Bayesian logistic head with 24 free weights and one intercept; for the four-category task, define  $24 \times 3$  free weights and three intercepts with the fourth category fixed as the zero-logit reference.
9. Assign independent Normal(0,1) priors to every free Bayesian-head parameter.
10. Draw the posterior-fitting subset from standardized training embeddings only, capping each analytical class at 3,000 observations; retain 12,000 rows for the four-category head and 6,000 rows for the binary head.

Return  $\phi$ , the selected training duration, standardized embeddings, Bayesian-head specification, and posterior-fitting indices.

---

Algorithm 3 therefore separates two parameter states that are intentionally different in MAL-BAYES-III:  $\phi$  is learned deterministically and then frozen, whereas the final-head parameters are declared probabilistic and subsequently sampled. This separation preserves a nonlinear learned representation while confining exact Bayesian inference to a compact, well-defined parameter space.

### 3.4 Hamiltonian Monte Carlo Posterior and Predictive Inference with Forensic Uncertainty Quantification

The Bayesian-head parameters defined in Section 3.3 were inferred with standard Hamiltonian Monte Carlo rather than NUTS or a stochastic-gradient approximation. This choice preserved a fixed-trajectory HMC mechanism while restricting the sampled state to the compact final head, a design consistent with recent work showing that exact HMC is most tractable when posterior dimensionality is controlled (Izmailov et al., 2021; Thiagarajan et al., 2025). Let  $\mathcal{D}_H = (\mathbf{Z}_H, \mathbf{y}_H)$  denote the training-only subset used for posterior fitting and let  $\theta$  collect the free binary or four-category head parameters. The target posterior is given in Equation (17).

$$p(\theta \mid \mathcal{D}_H) \propto p(\mathbf{y}_H \mid \mathbf{Z}_H, \theta) p(\theta) \quad (17)$$

HMC represents the negative log posterior as potential energy. The potential-energy function used to guide the gradient-based proposals is defined by Equation (18). An auxiliary momentum vector  $\mathbf{r}$  was introduced with Gaussian density and diagonal mass matrix  $\mathbf{M}$ ; its corresponding kinetic energy is shown in Equation (19). During warm-up, the implementation adapted the step size and diagonal mass matrix, while the trajectory length remained fixed at 1.5. The target acceptance probability was 0.90.

$$U(\theta) = -\log p(\theta \mid \mathcal{D}_H) \quad (18)$$

$$\mathbf{r} \sim \mathcal{N}(\mathbf{0}, \mathbf{M}), \quad K_{\text{HMC}}(\mathbf{r}) = \frac{1}{2} \mathbf{r}^\top \mathbf{M}^{-1} \mathbf{r} \quad (19)$$

The joint Hamiltonian is the sum of potential and kinetic energy, as expressed in Equation (20). Hamiltonian dynamics evolve the parameter and momentum states according to Equation (21). In practice, these continuous dynamics were approximated by leapfrog integration. For step size  $\epsilon$ , one leapfrog update consists of a half momentum step, a full parameter step, and a second half momentum step, as formalized in Equations (22a)-(22c). Gradient-informed trajectories allow the sampler to propose distant states while controlling energy error, which is the principal advantage of HMC over uninformed random-walk proposals in continuous Bayesian neural models (Izmailov et al., 2021; Thiagarajan et al., 2025).

$$H(\theta, \mathbf{r}) = U(\theta) + K_{\text{HMC}}(\mathbf{r}) \quad (20)$$

$$\frac{d\theta}{dt} = \mathbf{M}^{-1} \mathbf{r}, \quad \frac{d\mathbf{r}}{dt} = -\nabla_{\theta} U(\theta) \quad (21)$$

$$\mathbf{r}_{\ell+1/2} = \mathbf{r}_{\ell} - \frac{\epsilon}{2} \nabla U(\theta_{\ell}) \quad (22a)$$

$$\theta_{\ell+1} = \theta_{\ell} + \epsilon \mathbf{M}^{-1} \mathbf{r}_{\ell+1/2} \quad (22b)$$

$$\mathbf{r}_{\ell+1} = \mathbf{r}_{\ell+1/2} - \frac{\epsilon}{2} \nabla U(\theta_{\ell+1}) \quad (22c)$$

Because numerical integration is approximate, the proposed terminal state was subjected to a Metropolis correction. Let  $\Delta H$  denote the proposed-minus-current Hamiltonian difference. The acceptance probability in Equation (23) preserves the required posterior target despite leapfrog discretization error. The HMC state was initialized from an L2-regularized logistic fit to the same posterior-fitting subset; for the four-category task, the fitted coefficients were transformed to the reference-class parameterization used by the Bayesian head. This initialization served only as a starting state and did not alter the posterior model.

$$\alpha = \min\{1, e^{-\Delta H}\} \quad (23)$$

Posterior sampling used four chains. Each chain underwent 1,200 warm-up iterations for adaptation and then retained 1,200 draws, yielding 4,800 post-warm-up parameter states across the four chains. Step-size adaptation and diagonal mass-matrix adaptation were enabled during warm-up, and sampling retained the fixed HMC trajectory length of 1.5. These settings were chosen for the final experiment so that posterior inference could be examined with multiple-chain convergence and effective-sampling diagnostics rather than relying on a single trajectory. The diagnostic criteria themselves are evaluated separately from the predictive construction so that convergence quality is not conflated with predictive accuracy.

For an evaluation embedding  $\mathbf{z}'^*$ , posterior uncertainty was propagated through the Bayesian head rather than summarized only at the parameter level. For each evaluation record, the posterior predictive probability vector is the posterior expectation of the conditional class-probability vector, as defined in Equation (24). The implementation approximated this integral by Monte Carlo averaging. From the 4,800 retained head draws, at most 800 draws were used for posterior prediction; when the retained set exceeded that limit, 800 draws were selected without replacement using a fixed task-specific random seed. Evaluation records were processed in chunks of 512 to control memory use without changing the underlying probability calculation.

$$\mathbf{p}_i^{\text{pred}} = \mathbb{E}_{\theta|\mathcal{D}_H}[\mathbf{p}(y_i | \mathbf{z}'_i, \theta)] \quad (24)$$

For observation  $i$  and category  $k$ , the posterior predictive mean was computed by averaging the class probability over the  $S$  selected posterior draws, as shown in Equation (25). The final prediction was the

category with the largest posterior mean probability, given by Equation (26). For the binary task  $K = 2$  and the draw-wise probabilities were obtained from the Bernoulli-logit head; for the four-category task  $K = 4$  and probabilities were produced from the three free logits together with the fixed reference logit introduced in Section 3.3.

$$\hat{p}_{ik} = \frac{1}{S} \sum_{s=1}^S p(y_i = k \mid \mathbf{z}'_i, \boldsymbol{\theta}^{(s)}) \quad (25)$$

$$\hat{y}_i = \arg \max_{k \in \{1, \dots, K\}} \hat{p}_{ik} \quad (26)$$

Three complementary uncertainty summaries were retained from the same posterior predictive draws. Predictive entropy measured total ambiguity in the posterior mean class distribution and is defined in Equation (27). Larger entropy indicates that posterior predictive mass is distributed more evenly across competing outcomes, whereas smaller values indicate concentration on a particular prediction. Entropy-based uncertainty has been used in recent uncertainty-aware cybersecurity research to distinguish confident from ambiguous decisions (Yang et al., 2024; Yumlembam et al., 2025).

$$H_i^{\text{pred}} = - \sum_{k=1}^K \hat{p}_{ik} \log \hat{p}_{ik} \quad (27)$$

Expected entropy measured the average conditional ambiguity remaining within individual posterior draws, as expressed in Equation (28). Mutual information was then obtained as the difference between predictive entropy and expected entropy in Equation (29). Under this decomposition, predictive entropy reflects total predictive uncertainty, whereas the mutual-information term captures the component associated with variation across sampled head parameters conditional on the frozen neural representation (Yang et al., 2024; Yumlembam et al., 2025).

$$H_i^{\text{exp}} = - \frac{1}{S} \sum_{s=1}^S \sum_{k=1}^K p_{ik}^{(s)} \log p_{ik}^{(s)} \quad (28)$$

$$MI_i = H_i^{\text{pred}} - H_i^{\text{exp}} \quad (29)$$

Posterior probability intervals provided a complementary dispersion-based measure. For each observation, the implementation first identified the class selected by the posterior predictive mean and then collected that class probability across the retained predictive draws. The lower and upper limits were the 2.5th and 97.5th percentiles of this draw-wise distribution, producing the predicted-class 95% credible interval and width in Equations (30a) and (30b). These intervals quantify uncertainty in the Bayesian final head conditional on the fixed neural representation and the declared training-only posterior subset; they do not constitute full-network Bayesian credible intervals (Izmailov et al., 2021; Andrade & Sato, 2025).

$$L_i = Q_{0.025} \left( p_{i\hat{y}_i}^{(1:S)} \right), \quad U_i = Q_{0.975} \left( p_{i\hat{y}_i}^{(1:S)} \right) \quad (30a)$$

$$W_i = U_i - L_i \quad (30b)$$

The posterior predictive mean, predictive entropy, expected entropy, mutual information, and predicted-class credible-interval width were produced from one common set of HMC draws, ensuring that the uncertainty quantities remained internally coherent with the final posterior prediction. Algorithm 4 summarizes the merged posterior-sampling and uncertainty-quantification procedure.

---

**Algorithm 4. HMC Posterior Sampling, Prediction, and Forensic Uncertainty Quantification**

**Input:** standardized posterior-fitting data (ZH, yH); Bayesian-head model; four chains; 1,200 warm-up iterations; 1,200 retained draws per chain; trajectory length 1.5; target acceptance 0.90; maximum 800 predictive draws; prediction chunk size 512.

**Output:** retained posterior head samples; posterior mean class probabilities; final predictions; predictive entropy; expected entropy; mutual information; 95% predicted-class credible intervals and widths.

1. Fit an L2-regularized logistic model to the HMC training subset and convert its coefficients to the binary or reference-class parameterization; use these values only to initialize the HMC state.
2. Define the log posterior from the Bernoulli likelihood for the binary head or the categorical-softmax likelihood for the four-category head together with the independent Normal(0,1) priors specified in Section 3.3.
3. For each chain, augment the current parameter vector with Gaussian momentum and evaluate the Hamiltonian using the negative log posterior as potential energy.
4. During 1,200 warm-up iterations, adapt the HMC step size and a diagonal mass matrix toward a target acceptance probability of 0.90 while retaining the fixed trajectory length of 1.5.
5. Generate each HMC proposal by leapfrog integration along the fixed-length trajectory and accept or reject the terminal state using the Metropolis correction.
6. After warm-up, retain 1,200 states from each of four chains, producing 4,800 posterior draws of the Bayesian-head parameters.
7. For posterior prediction, use all retained draws when their number does not exceed 800; otherwise select 800 draws without replacement using a fixed task-specific random seed.
8. Process evaluation embeddings in chunks of 512. For every selected posterior draw, compute Bernoulli probabilities for the binary task or reference-class softmax probabilities for the four-category task.
9. Average the draw-wise class probabilities to obtain the posterior predictive mean and assign the final category by maximum posterior mean probability.
10. Compute predictive entropy from the posterior mean, expected entropy from the draw-wise distributions, and mutual information as their difference.
11. For the final predicted class, obtain the 2.5th and 97.5th percentiles of the draw-wise class probability and define the 95% credible-interval width as their difference.
12. Return the retained posterior and all prediction- and uncertainty-ready quantities without refitting the encoder or Bayesian head on validation or locked-test data.

**Return** posterior samples, mean probabilities, predictions, entropy measures, mutual information, and predicted-class credible intervals.

---

Algorithm 4 therefore converts the compact Bayesian head from a parameter distribution into case-level forensic evidence: each evaluation record receives a posterior mean class distribution together with complementary measures of predictive ambiguity, parameter-sensitive disagreement, and probability dispersion. No validation or locked-test record contributes to posterior fitting, and no encoder parameter is resampled during this stage.

### 3.5 Calibration, Comparator Models, and Evaluation Measures

MAL-BAYES-III was evaluated against probabilistic classical, deterministic neural, and approximate neural-uncertainty comparators so that posterior performance was not interpreted in isolation. All comparator models used the same training, validation, and locked-test observation identities established earlier. The classical set comprised L2-regularized Logistic Regression with balanced class weights; a 600-tree Random Forest using balanced subsampling and square-root feature sampling; Histogram Gradient Boosting with 300 boosting iterations, learning rate 0.08, and L2 regularization 1.0; an MLP with hidden dimensions 128 and 64, L2 penalty  $10^{-4}$ , early stopping, and a maximum of 400 iterations; and a Linear SVM with  $C = 1$  and balanced class weights whose decision scores were converted to probabilities by three-fold sigmoid calibration. These models provided conventional reference points for discrimination and probability quality without sharing the HMC posterior construction.

Three neural comparators were derived from the representation architecture defined in Figure 1. The deterministic DNN used the task-specific network obtained after training-only epoch selection and full-training refit. Its logits were temperature-scaled using the outer validation partition only. The selected scalar temperature minimized validation negative log-likelihood over the bounded interval  $T$  in  $[e^{-3}, e^3]$ , as shown in Equation (31). The corresponding multiclass probability mapping is given in Equation (32); for the binary task, the same division by  $T$  preceded the sigmoid transformation. The fitted temperature was then frozen and applied unchanged to the locked-test logits.

$$T^* = \arg \min_{T \in [e^{-3}, e^3]} \text{NLL}_{\text{val}}(T) \quad (31)$$

$$p_T(y_i = k \mid \ell_i) = \frac{\exp(\ell_{ik}/T)}{\sum_{c=1}^K \exp(\ell_{ic}/T)} \quad (32)$$

MC Dropout retained the fitted deterministic network but reactivated only its dropout layers at prediction time while leaving the remaining modules in evaluation mode. Two hundred stochastic forward passes were generated for each evaluation partition, and their mean probability vector was used for predictive scoring; the draw distribution also supplied uncertainty quantities. The Deep Ensemble contained five independently trained members with the same 52-128-64-24 architecture. The refitted deterministic network served as the first member, and four additional networks were trained on the complete training partition for the same selected number of epochs using different fixed seeds. Their predictive probabilities were averaged. These comparators separate post-hoc calibration, stochastic dropout variation, and between-network ensemble variation from the HMC posterior mechanism.

Predictive discrimination was summarized with accuracy, balanced accuracy, class-specific precision, recall and F1-score, macro-F1, weighted-F1, AUROC, and AUPRC. Accuracy measured the overall proportion of correct decisions, whereas balanced accuracy averaged class-specific recall and therefore prevented the larger Benign category from dominating the category-level summary. Their definitions are given in Equations (33) and (34).

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\hat{y}_i = y_i) \quad (33)$$

$$BA = \frac{1}{K} \sum_{k=1}^K \frac{TP_k}{TP_k + FN_k} \quad (34)$$

For category  $k$ , precision, recall, and F1-score were calculated from the corresponding true-positive, false-positive, and false-negative counts, as defined in Equations (35a)-(35c). Macro-F1 gave every analytical category equal weight, while weighted-F1 weighted each category by its locked-test support, as defined in Equations (36a) and (36b). For the binary task, malware-class precision, recall, F1-score, and support were also retained explicitly because overall near-saturated accuracy can obscure the direction of the remaining errors. The complete locked-test comparison is reported in Tables 5 and 6, while Table 7 reports the malware-class binary measures.

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k} \quad (35a)$$

$$\text{Recall}_k = \frac{TP_k}{TP_k + FN_k} \quad (35b)$$

$$F1_k = \frac{2 \text{Precision}_k \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \quad (35c)$$

$$\text{Macro} - \text{F1} = \frac{1}{K} \sum_{k=1}^K \text{F1}_k \quad (36a)$$

$$\text{Weighted} - \text{F1} = \sum_{k=1}^K \frac{n_k}{N} \text{F1}_k \quad (36b)$$

Probability quality was evaluated separately from classification correctness. Negative log-likelihood penalized probability mass withheld from the observed category, and the Brier score measured the squared error of the full probability vector against the one-hot outcome. Their definitions are given in Equations (37) and (38). Lower values indicate better probabilistic fit. Using proper-scoring and calibration measures alongside discrimination is important in uncertainty-aware cybersecurity because high classification performance does not by itself establish well-calibrated probabilities (Yang et al., 2024; Yumlembam et al., 2025).

$$\text{NLL} = -\frac{1}{N} \sum_{i=1}^N \log \hat{p}_{i,y_i} \quad (37)$$

$$\text{Brier} = \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K (\hat{p}_{ik} - \mathbb{I}(y_i = k))^2 \quad (38)$$

Calibration was quantified with top-label Expected Calibration Error (ECE) and Maximum Calibration Error (MCE). For every observation, confidence was the largest predicted class probability and correctness indicated whether the associated class matched the observed label. The interval  $[0,1]$  was partitioned into 15 fixed equal-width bins. For non-empty bin  $S_b$ ,  $a_b$  denotes observed accuracy and  $c_b$  denotes mean confidence. ECE was the support-weighted mean absolute gap between these quantities, whereas MCE retained the largest non-empty-bin gap, as defined in Equations (39) and (40). The same calibration definitions were applied to every method so that comparisons were not affected by changing bin rules. Reliability diagrams were generated as complementary visual diagnostics.

$$\text{ECE} = \sum_{b=1}^{B_{\text{bin}}} \frac{|S_b|}{N} |a_b - c_b| \quad (39)$$

$$\text{MCE} = \max_{b: |S_b| > 0} |a_b - c_b| \quad (40)$$

AUROC and AUPRC were computed from the malware probability for the binary task and as macro one-versus-rest summaries for the four-category task. Before metric calculation, each probability matrix was checked for finite values, clipped only at a numerical epsilon where logarithmic scoring required it, and renormalized row-wise to the probability simplex. This common evaluation contract ensured that classical, deterministic, approximate-uncertainty, ensemble, and HMC predictions were scored against identical targets and definitions. The evaluation therefore separates three questions that are interpreted independently later: whether the method predicts the correct category, whether its probability assignments are calibrated, and whether its uncertainty representation is informative beyond ordinary confidence.

### 3.6 HMC Convergence, Numerical Diagnostics, Uncertainty–Error Association, and Selective Prediction

For every free weight and intercept parameter, the diagnostic summary included rank-normalized split R-hat, bulk and tail effective sample sizes (ESS), and Monte Carlo standard error (MCSE). These quantities assess complementary aspects of simulation quality: between-chain agreement, the effective amount of information available for central and tail posterior functionals, and the simulation error attached to posterior means. Multiple-chain convergence diagnostics are particularly important for HMC because apparently plausible predictive performance does not establish that the sampled parameter distribution has been explored adequately (Vehtari et al., 2021). The aggregate quantities used for reporting were the maximum R-hat, minimum bulk ESS, minimum tail ESS, and maximum MCSE across the sampled head parameters, as defined in Equations (41)–(43).

$$\hat{R}_{\max} = \max_j \hat{R}_j \quad (41)$$

$$\text{ESS}_{\text{bulk},\min} = \min_j \text{ESS}_{\text{bulk},j} \quad (42a)$$

$$\text{ESS}_{\text{tail},\min} = \min_j \text{ESS}_{\text{tail},j} \quad (42b)$$

$$\text{MCSE}_{\max} = \max_j \text{MCSE}(\bar{\theta}_j) \quad (43)$$

Sampler behaviour was audited separately from parameter-wise convergence. Let  $C_H$  denote the number of HMC chains,  $n_{\text{ret}}$  the number of retained iterations per chain,  $a_{cs}$  the recorded acceptance probability, and  $d_{cs}$  the divergence indicator for retained state  $s$  of chain  $c$ . Mean acceptance probability and the retained divergence fraction were calculated as in Equations (44a) and (44b); mean leapfrog-step count was also recorded as an empirical indicator of trajectory effort. The final HMC configuration used four chains, 1,200 warm-up iterations and 1,200 retained draws per chain, so these summaries were calculated from the 4,800 post-warm-up states. The predeclared final-readiness conditions required maximum R-hat no greater than 1.01, minimum bulk ESS of at least 400, and a divergence fraction no greater than 0.001, as summarized in Equation (45). All aggregate diagnostics were additionally required to be finite, while acceptance probability was required to remain numerically valid; the target acceptance probability used during adaptation remained 0.90 as specified in Section 3.4. Table 10 reports the resulting diagnostic summary, while Figures 5 and 9 provide the corresponding multiclass and binary intercept trace diagnostics.

$$\bar{a} = \frac{1}{C_H n_{\text{ret}}} \sum_{c=1}^{C_H} \sum_{s=1}^{n_{\text{ret}}} a_{cs} \quad (44a)$$

$$f_{\text{div}} = \frac{1}{C_H n_{\text{ret}}} \sum_{c=1}^{C_H} \sum_{s=1}^{n_{\text{ret}}} \mathbb{I}(d_{cs} = 1) \quad (44b)$$

$$\begin{aligned} \hat{R}_{\max} &\leq 1.01, \\ \text{ESS}_{\min} &\geq 400, \\ f_{\text{div}} &\leq 0.001. \end{aligned} \quad (45)$$

The second component of the merged audit examined whether uncertainty was associated with actual prediction errors after every model and posterior state had been frozen. This analysis was restricted to MC Dropout, the Deep Ensemble, and MAL-BAYES-III HMC because these methods produced repeated predictive draws from which the three common uncertainty summaries could be constructed. For each method, the evaluated scores were predictive entropy, mutual information, and predicted-class interval width. For

HMC, the interval width is the 95% posterior credible-interval width defined in Section 3.4; for MC Dropout and the Deep Ensemble, the analogous quantity is an empirical interval over stochastic or member-wise predictive probabilities and is not interpreted as a Bayesian credible interval. The binary error indicator for observation  $i$  is defined in Equation (46).

$$e_i = \mathbb{I}(\hat{y}_i \neq y_i) \quad (46)$$

Two complementary statistics quantified error association. First, Spearman correlation measured the monotonic association between an uncertainty score  $u_i$  and the binary error indicator, using the rank correlation in Equation (47). Second, error-detection AUROC treated erroneous predictions as the positive class and the uncertainty score as the detection statistic. Its probabilistic interpretation is given in Equation (48): a value above 0.5 indicates that an erroneous prediction tends to receive greater uncertainty than a correct prediction, with ties contributing one half. Mean uncertainty among correct and erroneous cases was retained alongside these ranking statistics. When the error indicator contained only one observed state, rank-based detection was undefined; more generally, very small error counts were treated as a limitation on the stability of the resulting AUROC rather than as strong evidence of uncertainty quality. Tables 8 and 9 report the multiclass and binary uncertainty-error analyses, respectively. This evaluation follows the broader uncertainty-aware cybersecurity principle that a useful uncertainty score should help identify predictions that warrant additional scrutiny rather than merely produce non-zero dispersion (Yang et al., 2024; Yumlembam et al., 2025).

$$\rho_s(u, e) = \text{Corr}(\text{rank}(u_i), \text{rank}(e_i)) \quad (47)$$

$$A_{\text{err}} = P(U_e > U_c) + \frac{1}{2}P(U_e = U_c) \quad (48)$$

Selective prediction translated this uncertainty ranking into a retained-risk diagnostic. For each of MC Dropout, the Deep Ensemble, and MAL-BAYES-III HMC, locked-test observations were ordered from lowest to highest predictive entropy. The coverage grid comprised 19 equally spaced levels from 1.00 to 0.10 in decrements of 0.05. If  $\pi_i$  denotes the entropy-order permutation and  $c$  is a requested coverage, the entropy ordering and retained sample size are defined in Equations (49a) and (49b), and the retained set together with its realized coverage is defined in Equation (50). Only the lowest-entropy observations were retained at each level; the remainder represented deferred cases. This construction was used to characterize risk-coverage behaviour after model fitting and was not used to refit the encoder, recalibrate probabilities, or alter the locked-test class predictions.

$$u_{\pi_1} \leq \dots \leq u_{\pi_N} \quad (49a)$$

$$m(c) = \max\{1, \text{round}(cN)\} \quad (49b)$$

$$\mathcal{R}(c) = \{\pi_1, \dots, \pi_{m(c)}\}, \quad \tilde{c} = \frac{m(c)}{N} \quad (50)$$

Selective risk was the empirical misclassification rate among the retained observations, as formalized in Equation (51). Macro-F1 was recomputed on the same retained subset using the definition in Equation (36a), and the retained sample count was recorded at every coverage level. A useful uncertainty ranking should preferentially defer difficult observations so that selective risk decreases as coverage is reduced, although the magnitude and stability of this effect depend on the number and distribution of errors. Figures 4 and 8 present the multiclass and binary risk-coverage curves generated from predictive-entropy ranking. Because these curves rank the frozen locked-test observations directly, they are interpreted as descriptive selective-prediction

diagnostics rather than as prospectively validated operational abstention thresholds. A deployment threshold would require selection on validation data and transfer unchanged to future cases (Yumlembam et al., 2025).

$$\text{Risk}(c) = \frac{1}{m(c)} \sum_{i \in \mathcal{R}(c)} \mathbb{I}(\hat{y}_i \neq y_i) \quad (51)$$

Algorithm 5 consolidates the convergence audit, uncertainty-error analysis, and entropy-guided selective-prediction evaluation. Keeping these operations in one procedure enforces a clear ordering: posterior numerical adequacy is assessed first, uncertainty is then tested against observed errors, and selective risk is finally summarized without using any of these locked-test diagnostics to modify the fitted model.

---

**Algorithm 5. HMC Diagnostic Auditing, Uncertainty–Error Association, and Selective Prediction**

**Input:** retained HMC chains and sampler diagnostics for the binary and four-category Bayesian heads; frozen locked-test probability draws and mean probabilities for MC Dropout, Deep Ensemble, and MAL-BAYES-III HMC; predictive entropy, mutual information, and predicted-class interval width; coverage grid  $\{1.00, 0.95, \dots, 0.10\}$ .

**Output:** aggregate HMC convergence and numerical-quality audit; uncertainty-error association statistics; predictive-entropy risk-coverage curves and retained-set performance summaries.

1. For every free Bayesian-head parameter, compute rank-normalized split R-hat, bulk ESS, tail ESS, and MCSE from the retained multiple-chain HMC draws.
2. Aggregate the parameter-wise diagnostics by retaining maximum R-hat, minimum bulk ESS, minimum tail ESS, and maximum MCSE; separately compute mean acceptance probability, divergence fraction, and mean leapfrog-step count from sampler diagnostics.
3. Require the final HMC audit to satisfy maximum R-hat  $\leq 1.01$ , minimum bulk ESS  $\geq 400$ , divergence fraction  $\leq 0.001$ , and finite aggregate diagnostics; retain trace plots as complementary visual evidence rather than as a substitute for numerical diagnostics.
4. For each uncertainty-producing method, obtain the frozen locked-test class prediction from its mean probability vector and mark each observation as correct or erroneous.
5. For predictive entropy, mutual information, and predicted-class interval width, calculate mean uncertainty for correct and erroneous observations, Spearman uncertainty-error association, and error-detection AUROC whenever both error states are represented.
6. Interpret HMC interval width as a posterior credible-interval measure; interpret MC-Dropout and ensemble interval widths only as empirical dispersion across repeated predictions or ensemble members.
7. For selective prediction, order locked-test observations from lowest to highest predictive entropy independently for each uncertainty-producing method.
8. At each requested coverage  $c$ , retain the first  $\max(1, \text{round}(cN))$  observations in the uncertainty ordering and defer all remaining observations.
9. On the retained subset, compute realized coverage, selective error, fixed-task macro-F1, and retained sample count; do not use these locked-test curves to refit, recalibrate, or select model parameters.
10. Export the diagnostic, uncertainty-error, and risk-coverage summaries for later Results reporting and forensic interpretation.

**Return** the HMC diagnostic audit, uncertainty-error statistics, and entropy-ranked selective-prediction summaries.

---

Algorithm 5 therefore prevents three distinct questions from being conflated. Convergence diagnostics establish whether the HMC posterior simulation is numerically credible; uncertainty-error statistics test whether the resulting uncertainty quantities correspond to difficult decisions; and risk-coverage analysis evaluates whether the same ranking can support selective forensic review. None of these diagnostics changes the frozen locked-test predictions, preserving their role as evaluation evidence rather than model-selection information.

### 3.7 Statistical Comparison and Controlled-Shift Evaluation

The final methodological stage examined two questions after every model, calibration parameter, neural encoder, latent scaler, and HMC posterior had been frozen. First, a paired nonparametric bootstrap quantified

the sampling variability of the macro-F1 difference between MAL-BAYES-III HMC and the temperature-scaled deterministic DNN on the same locked-test observations. Second, controlled feature shifts tested how the two frozen predictors responded when standardized memory-forensic inputs were perturbed or partially suppressed. Neither procedure contributed information back to model fitting or calibration. The shift experiments were therefore treated as synthetic sensitivity tests rather than evidence of genuine unseen-malware, open-set, temporal, or external out-of-distribution generalisation; those stronger claims require data whose distributional novelty is established independently of the benchmark (Maniriho et al., 2024; Madamidola et al., 2025).

The paired bootstrap was conducted separately for the binary and four-category tasks with 1,000 repetitions. At repetition  $b$ ,  $N$  locked-test row indices were sampled with replacement from the  $N$  available cases. The same resampled indices were applied to the HMC and temperature-scaled DNN predictions, preserving the paired observation structure. The implementation used an ordinary row-wise bootstrap rather than a category-stratified bootstrap. If  $I^{(b)}$  denotes the resampled index vector, its construction is represented in Equation (52).

$$i_r^{(b)} \stackrel{iid}{\sim} \mathcal{U}_d\{1, \dots, N\}, \quad r = 1, \dots, N \quad (52)$$

For each resample, macro-F1 was recomputed independently from the two frozen probability matrices and the paired difference was oriented as MAL-BAYES-III HMC minus temperature-scaled DNN, as defined in Equation (53). Positive values therefore favour the HMC predictor for macro-F1, whereas negative values favour the deterministic DNN. No other metric was used in this implemented bootstrap comparison.

$$\Delta^{(b)} = F_{1,\text{macro}}^{\text{HMC}}(I^{(b)}) - F_{1,\text{macro}}^{\text{DNN}}(I^{(b)}) \quad (53)$$

The bootstrap summary retained the mean paired difference, the percentile 95% interval, and the empirical proportion of resamples in which the difference exceeded zero. These quantities are defined in Equations (54a), (54b), and (55), respectively. The empirical proportion in Equation (55) is a bootstrap directional proportion rather than a frequentist p-value. A percentile interval excluding zero was treated as evidence that the observed direction was stable under locked-test resampling; an interval spanning zero was interpreted as insufficient evidence of a stable macro-F1 difference.

$$\bar{\Delta} = \frac{1}{B_{\text{boot}}} \sum_{b=1}^{B_{\text{boot}}} \Delta^{(b)}, \quad B_{\text{boot}} = 1000 \quad (54a)$$

$$\text{CI}_{0.95}(\Delta) = [Q_{0.025}(\Delta), Q_{0.975}(\Delta)] \quad (54b)$$

$$\hat{P}(\Delta > 0) = \frac{1}{B_{\text{boot}}} \sum_{b=1}^{B_{\text{boot}}} \mathbb{I}(\Delta^{(b)} > 0) \quad (55)$$

Controlled-shift evaluation operated on the already preprocessed locked-test feature matrix and used the fixed severity set  $s$  in  $\{0.05, 0.10, 0.20, 0.35\}$ . Two shift mechanisms were applied. Under Gaussian perturbation, independent zero-centred noise with standard deviation  $s$  was added element-wise to the standardized feature matrix, as shown in Equation (56). Because the clean predictors had already been standardized using training-derived parameters, the severity values are interpretable relative to that standardized feature scale rather than to the heterogeneous raw-feature units.

$$X_{s,ij}^{(G)} = X_{ij} + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathcal{N}(0, s^2) \quad (56)$$

Feature masking independently replaced an element with zero with probability  $s$ , as defined in Equation (57). Zero is the training-derived standardized reference point after the fitted preprocessing transformation, so masking represented progressive loss or neutralization of observed feature information rather than removal of entire records or malware categories. The same four severities were used for Gaussian perturbation and masking, giving eight controlled shift conditions per analytical task.

$$\zeta_{ij} \sim \text{Bern}(s), \quad X_{s,ij}^{(\text{mask})} = X_{ij}(1 - \zeta_{ij}) \quad (57)$$

Each stressed matrix was passed through the already fitted temperature-scaled DNN without retraining or re-estimating the temperature. For MAL-BAYES-III HMC, the same stressed matrix was passed through the frozen task-specific encoder, transformed by the latent StandardScaler fitted to clean training embeddings, and evaluated with the retained Bayesian-head posterior. To limit repeated stress-evaluation cost, 200 posterior draws were used for each HMC stress condition, with evaluation processed in chunks of 512 observations. The same prediction and probability-quality metrics defined in Section 3.5 were then recomputed for both methods at every severity. Consequently, any change under stress reflects the response of the already frozen predictive system rather than adaptation to the perturbed data.

These controlled shifts were deliberately narrower than adversarial or external-shift evaluation. Gaussian noise probes local sensitivity to diffuse feature perturbation, whereas random masking probes sensitivity to progressive feature absence. Neither mechanism reproduces a specific obfuscation strategy, establishes a new malware distribution, or demonstrates adversarial robustness. This distinction is important because Bayesian prediction does not automatically confer robustness to deliberately constructed perturbations (Feng et al., 2024). The resulting shift metrics are therefore interpreted only as controlled robustness diagnostics. The paired-bootstrap and controlled-shift outputs are retained for the final Results analysis when their exported tables are available; numerical values are not inferred from the methodology alone.

After both tasks completed, the experiment also consolidated the numerical-readiness conditions defined throughout Section 3. Probability matrices were required to contain finite values within the unit interval and to sum to one within numerical tolerance; the HMC convergence and divergence conditions from Section 3.6 had to pass; and the minimum predictive-signal criterion had to be satisfied before the final results were released. Table 14 reports these final numerical-readiness gate results. Algorithm 6 summarizes the paired bootstrap, controlled-shift evaluation, and final integrity checks that complete the MAL-BAYES-III methodology.

---

#### Algorithm 6. Paired Bootstrap Comparison and Controlled-Shift Evaluation

**Input:** locked-test labels  $y$ ; frozen MAL-BAYES-III HMC and temperature-scaled DNN predictions; clean standardized locked-test matrix  $X$ ; fitted neural encoder and latent scaler; retained Bayesian-head posterior; 1,000 bootstrap repetitions; severity set  $\{0.05, 0.10, 0.20, 0.35\}$ .

**Output:** paired macro-F1 bootstrap summary; Gaussian-perturbation and feature-masking performance records for both tasks; final numerical-readiness audit.

1. For each analytical task, freeze the clean locked-test labels and the HMC and temperature-scaled DNN probability matrices; do not refit, recalibrate, or reselect any model component.
  2. For  $b = 1, \dots, 1,000$ , sample  $N$  locked-test row indices with replacement and apply the same resampled indices to both methods.
  3. Compute macro-F1 for MAL-BAYES-III HMC and the temperature-scaled DNN on each paired resample and store the difference HMC minus DNN.
  4. Summarize the paired differences by their mean, 2.5th and 97.5th percentiles, and the proportion of resamples with a positive difference; do not interpret this proportion as a p-value.
-

- 
5. For each severity  $s$  in  $\{0.05, 0.10, 0.20, 0.35\}$ , generate a Gaussian-shift matrix by adding independent  $\text{Normal}(0, s^2)$  noise to every standardized locked-test feature.
  6. For the same severity set, generate a masking-shift matrix by independently replacing each standardized feature value with zero with probability  $s$ .
  7. Apply the frozen DNN and its validation-selected temperature directly to every stressed matrix without parameter updates.
  8. For HMC, pass every stressed matrix through the frozen encoder and training-fitted latent scaler, then compute posterior predictions with 200 retained head draws in chunks of 512 observations.
  9. Recompute the common discrimination and probability-quality metrics under every stress condition and retain task, stress type, and severity with each record.
  10. Treat Gaussian perturbation and feature masking strictly as controlled sensitivity analyses, not as evidence of genuine unseen-malware, adversarial, open-set, or external OOD performance.
  11. Verify final probability integrity and the previously declared HMC readiness criteria before releasing the experimental outputs for Results reporting.
- Return** the paired-bootstrap summary, controlled-shift metrics, and final readiness audit.
- 

Algorithm 6 completes the experimental protocol by preserving the paired identity of locked-test observations during statistical resampling and by applying controlled perturbations only after all predictive components are frozen. Together with the convergence, uncertainty, and selective-prediction audits in Section 3.6, this final stage separates sampling variability, numerical posterior adequacy, uncertainty usefulness, and sensitivity to synthetic feature shifts rather than treating them as a single notion of model robustness.

## 4. Results and Analysis

### 4.1 Data Integrity and Experimental Readiness

The empirical analysis first examined whether the cleaned corpus and frozen data partitions satisfied the data-level conditions required for credible locked-test evaluation. After duplicate and conflicting-label controls, the modelling corpus contained 58,018 label-consistent observations with 52 retained predictors. The category-stratified split assigned 37,131 observations to training, 9,283 to validation, and 11,604 to the locked test, corresponding to 64%, 16%, and 20% of the cleaned corpus, respectively. All four analytical categories remained represented in every partition. As shown in Table 2, the locked test contained 5,847 Benign, 1,903 Ransomware, 1,959 Spyware, and 1,895 Trojan Horse observations, providing category-level support for the subsequent multi-class analysis.

**Table 2. Train, Validation, and Locked-Test Distribution**

| Split       | N      | Benign | Ransomware | Spyware | Trojan Horse |
|-------------|--------|--------|------------|---------|--------------|
| Training    | 37,131 | 18,707 | 6,090      | 6,269   | 6,065        |
| Validation  | 9,283  | 4,677  | 1,523      | 1,567   | 1,516        |
| Locked test | 11,604 | 5,847  | 1,903      | 1,959   | 1,895        |

The exact-vector partition audit satisfied its required independence condition: no complete predictor-vector hash was shared between training, validation, and locked-test partitions. The additional nearest-neighbour audit nevertheless showed that highly similar, but non-identical, records remained across partitions. Table 3 reports a minimum standardized Euclidean distance of  $6.682 \times 10^{-6}$  from training to validation and  $5.960 \times 10^{-7}$  from training to the locked test. The corresponding first-percentile distances were 0.001371 and 0.001992, while the median distances were 0.180751 and 0.191776. Thus, the audit identified isolated observations lying very close to training records, but the positive distances and zero exact-hash overlap distinguish this local similarity from literal cross-partition duplication.

**Table 3. Cross-Partition Near-Duplicate Distance Audit**

| Comparison             | Minimum Distance       | Median Distance | P01 Distance |
|------------------------|------------------------|-----------------|--------------|
| Training → Validation  | $6.682 \times 10^{-6}$ | 0.180751        | 0.001371     |
| Training → Locked test | $5.960 \times 10^{-7}$ | 0.191776        | 0.001992     |

The optional variant-group audit provided a complementary view of partition composition. The selected grouping field contained 16 represented groups, and every group observed in validation and locked testing was already represented in training. Table 4 therefore records zero unseen-group rows and an unseen-group fraction of 0.000000 for both evaluation partitions. This result confirms complete group coverage under the implemented audit, but it also limits the interpretation of the experiment: the locked test is not a group-held-out or unseen-malware evaluation because none of the 16 audited groups was absent from training.

**Table 4. Variant-Group Audit**

| Split       | Rows   | Unseen Group Rows | Unseen Group Fraction | Unique Groups |
|-------------|--------|-------------------|-----------------------|---------------|
| Validation  | 9,283  | 0                 | 0.000000              | 16            |
| Locked test | 11,604 | 0                 | 0.000000              | 16            |

These checks support locked-test evaluation but bound the evidence to leakage-controlled within-benchmark generalisation: no exact cross-partition overlap was present, yet the near-neighbour similarity and complete 16-group overlap preclude claims of unseen-family, source-independent, or external out-of-distribution performance.

## 4.2 Multiclass Locked-Test Predictive Performance and Calibration

The four-category locked-test task exposed a clear separation between maximum discriminative performance and calibration quality. Table 5 compares the nine evaluated methods across overall discrimination, class-balanced performance, proper probability scores, and calibration error. Histogram Gradient Boosting (HGB) achieved the strongest multiclass discrimination, with accuracy of 0.874440, balanced accuracy of 0.809967, macro-F1 of 0.809754, and weighted-F1 of 0.874271. Random Forest was almost indistinguishable on these measures, reaching 0.874095 accuracy and 0.809461 macro-F1. These two classical ensemble methods therefore established the highest predictive benchmark for the four-category task.

**Table 5. Multiclass Locked-Test Predictive Performance and Calibration**

**Panel A. Discrimination metrics**

| Method                      | Accuracy        | Balanced Accuracy | Macro-F1        | Weighted-F1     |
|-----------------------------|-----------------|-------------------|-----------------|-----------------|
| Histogram Gradient Boosting | <b>0.874440</b> | <b>0.809967</b>   | <b>0.809754</b> | <b>0.874271</b> |
| Random Forest               | 0.874095        | 0.809522          | 0.809461        | 0.874101        |
| Deep Ensemble               | 0.824285        | 0.734265          | 0.734255        | 0.824356        |
| Temperature-scaled DNN      | 0.822130        | 0.730773          | 0.730462        | 0.821859        |
| MC Dropout                  | 0.822130        | 0.730791          | 0.730294        | 0.821763        |
| <b>MAL-BAYES-III HMC</b>    | 0.821527        | 0.730090          | 0.730271        | 0.821695        |
| MLP                         | 0.815926        | 0.721798          | 0.722343        | 0.816451        |
| Logistic Regression         | 0.748621        | 0.621597          | 0.618201        | 0.747112        |
| Calibrated Linear SVM       | 0.742244        | 0.611738          | 0.607578        | 0.740102        |

**Panel B. Probability-quality and calibration metrics**

| Method                      | NLL             | Brier           | ECE             | MCE             |
|-----------------------------|-----------------|-----------------|-----------------|-----------------|
| Histogram Gradient Boosting | <b>0.300988</b> | <b>0.172378</b> | 0.011759        | 0.058638        |
| Random Forest               | 0.311266        | 0.177828        | 0.016248        | 0.077023        |
| Deep Ensemble               | 0.379355        | 0.224909        | 0.013826        | 0.063720        |
| Temperature-scaled DNN      | 0.386655        | 0.229017        | 0.010145        | <b>0.036348</b> |
| MC Dropout                  | 0.388931        | 0.230035        | 0.016466        | 0.055235        |
| <b>MAL-BAYES-III HMC</b>    | 0.384903        | 0.228269        | <b>0.007615</b> | 0.041573        |
| MLP                         | 0.404844        | 0.238171        | 0.011880        | 0.049883        |
| Logistic Regression         | 0.502137        | 0.294612        | 0.016586        | 0.140547        |
| Calibrated Linear SVM       | 0.510406        | 0.298398        | 0.021105        | 0.328860        |

*Note. Higher values are better for the discrimination metrics in Panel A; lower values are better for NLL, Brier, ECE, and MCE in Panel B. Bold values identify the best result in each column.*

MAL-BAYES-III HMC did not match the discrimination of HGB or Random Forest. Its accuracy was 0.821527 and macro-F1 was 0.730271, corresponding to absolute deficits of 0.052913 in accuracy and 0.079483 in macro-F1 relative to HGB. The same pattern was visible in balanced accuracy, where HMC reached 0.730090 compared with 0.809967 for HGB. The Bayesian neural-head result should therefore not be interpreted as a multiclass accuracy improvement over the strongest ensemble baselines.

The class-level error structure behind this aggregate performance is shown in Figure 2. Benign detection was almost perfect, whereas most multiclass errors occurred among Ransomware, Spyware, and Trojan Horse, indicating that the principal difficulty was malware-category separation rather than benign-versus-malware discrimination.

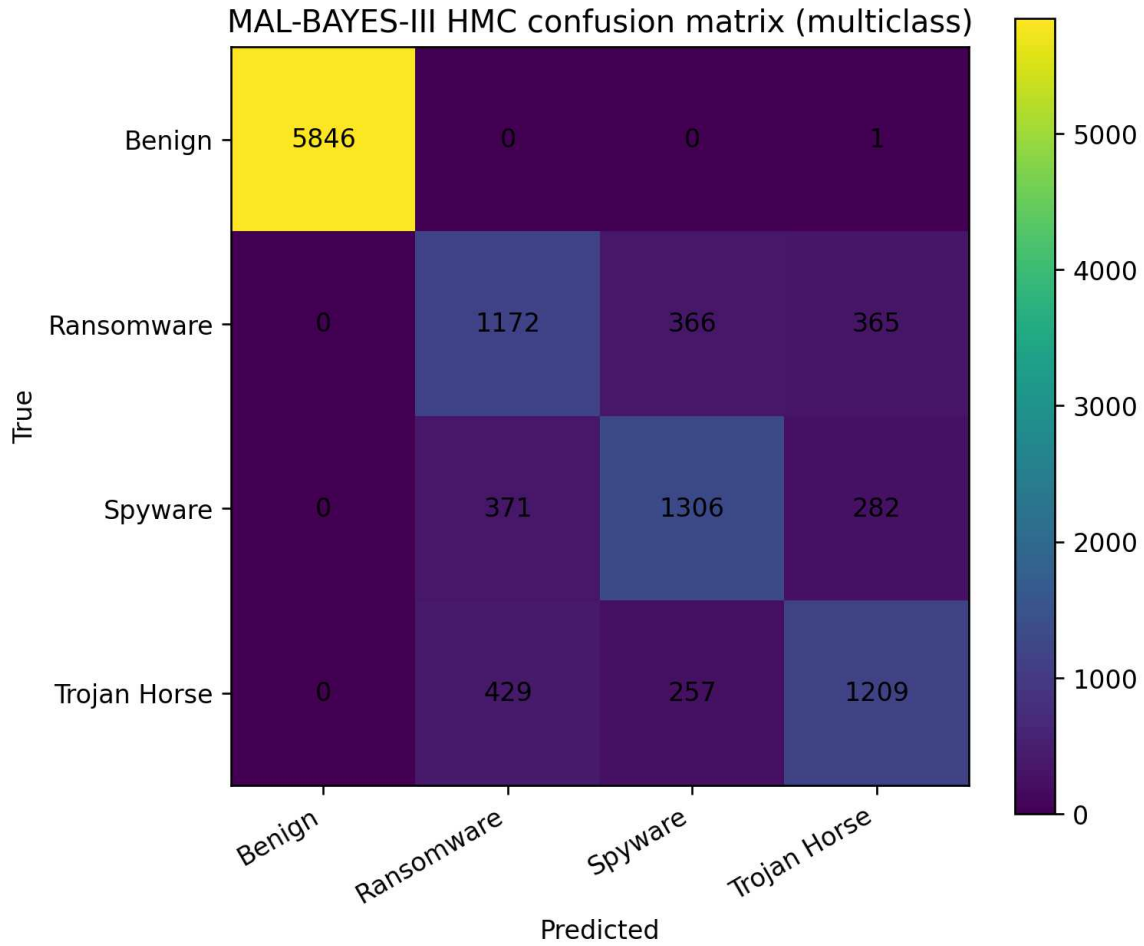


**Figure 2. MAL-BAYES-III HMC multiclass confusion matrix for the locked-test set.**

Against the neural comparators, however, the predictive differences were much smaller. The temperature-scaled DNN reached 0.822130 accuracy and 0.730462 macro-F1, only 0.000603 and 0.000191 above HMC, respectively. MC Dropout had the same accuracy as the temperature-scaled DNN and a macro-F1 of 0.730294, differing from HMC by only 0.000023. The Deep Ensemble was moderately stronger, with 0.824285 accuracy and 0.734255 macro-F1. HMC nevertheless outperformed the standalone MLP, Logistic Regression, and calibrated Linear SVM on all four discrimination measures reported in Panel A.

The calibration results altered the ranking. MAL-BAYES-III HMC recorded the lowest ECE of all displayed methods at 0.007615. The next-lowest ECE was 0.010145 for the temperature-scaled DNN, followed by 0.011759 for HGB. HMC also achieved an MCE of 0.041573, second only to the temperature-scaled DNN value of 0.036348. This advantage was not universal across probability-quality measures: HGB retained the best NLL (0.300988) and Brier score (0.172378), while HMC produced NLL of 0.384903 and Brier score of 0.228269. Thus, the HMC head was the most closely calibrated under the aggregate ECE criterion, but it did not provide the strongest overall probabilistic fit.

Figure 3 provides the corresponding reliability comparison for the neural uncertainty models. The curves remain close to the diagonal over most populated confidence regions, while the HMC curve shows some localized deviation at higher intermediate confidence despite its lowest aggregate ECE.

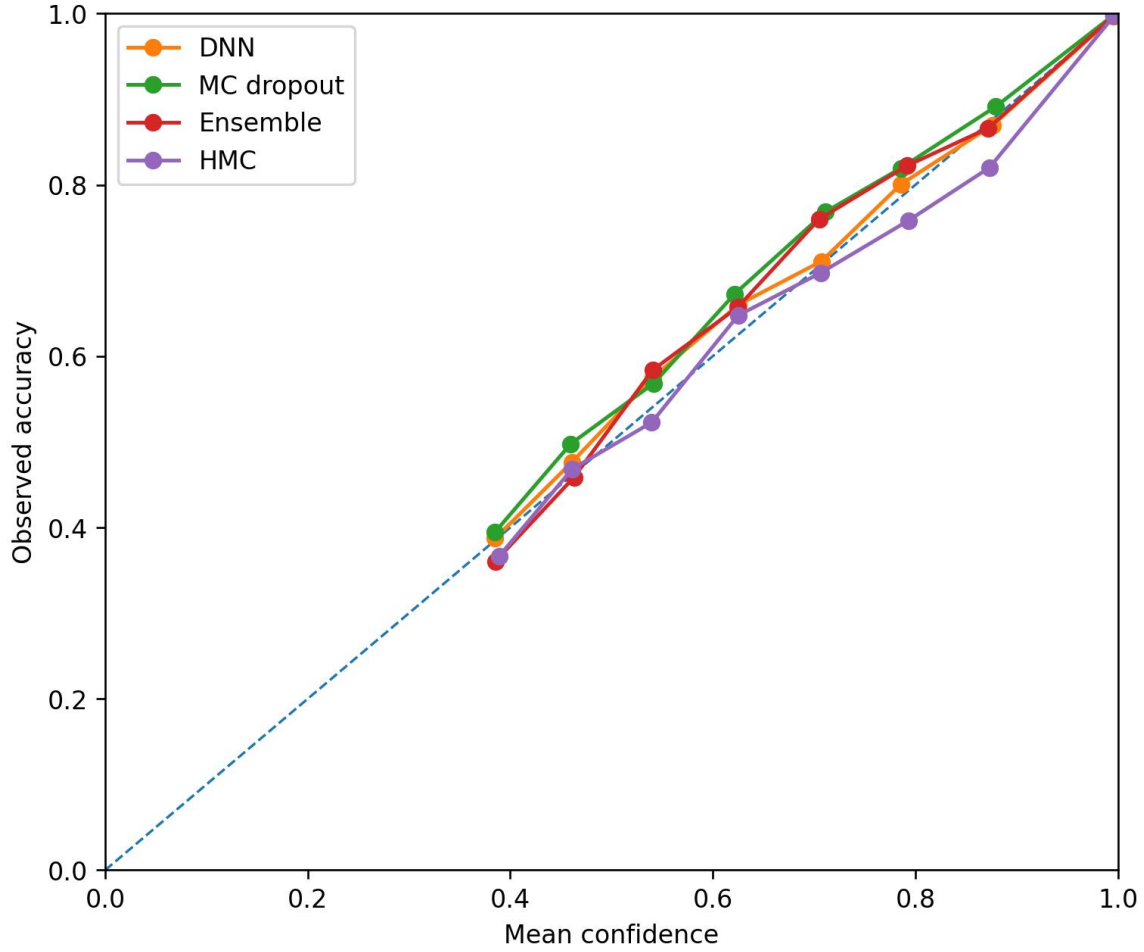


**Figure 3. Multiclass reliability comparison for the temperature-scaled DNN, MC Dropout, Deep Ensemble, and MAL-BAYES-III HMC.**

Within the neural family, HMC combined essentially unchanged multiclass macro-F1 with improved calibration relative to the temperature-scaled DNN and MC Dropout. Compared with the temperature-scaled DNN, HMC reduced ECE by 0.002530 and slightly improved NLL and Brier score by 0.001752 and 0.000748, respectively, although its MCE was 0.005225 higher. Relative to MC Dropout, HMC reduced ECE by 0.008851 while preserving nearly identical macro-F1. The multiclass result therefore supports the intended trade-off of MAL-BAYES-III: posterior neural-head inference did not maximize category-level accuracy, but it retained neural-level discrimination while producing the lowest aggregate calibration error among the evaluated methods.

### 4.3 Binary Locked-Test Performance and Calibration

The binary Benign-versus-Malware task was substantially easier than the four-category problem and produced near-ceiling performance for several methods. Table 6 shows that Random Forest, the temperature-scaled DNN, MC Dropout, Deep Ensemble, and MAL-BAYES-III HMC all achieved the same locked-test accuracy, balanced accuracy, macro-F1, and weighted-F1 of 0.999914. On 11,604 locked-test observations, this value corresponds to only one classification error. Histogram Gradient Boosting and the MLP were only marginally lower at approximately 0.99974, while the calibrated Linear SVM and Logistic Regression remained above 0.998. The binary task therefore provided little separation among the strongest models when judged solely by class-label accuracy.

**Table 6. Binary Locked-Test Predictive Performance and Calibration****Panel A. Discrimination metrics**

| Method                      | Accuracy        | Balanced Accuracy | Macro-F1        | Weighted-F1     |
|-----------------------------|-----------------|-------------------|-----------------|-----------------|
| Random Forest               | <b>0.999914</b> | <b>0.999914</b>   | <b>0.999914</b> | <b>0.999914</b> |
| Temperature-scaled DNN      | <b>0.999914</b> | <b>0.999914</b>   | <b>0.999914</b> | <b>0.999914</b> |
| MC Dropout                  | <b>0.999914</b> | <b>0.999914</b>   | <b>0.999914</b> | <b>0.999914</b> |
| Deep Ensemble               | <b>0.999914</b> | <b>0.999914</b>   | <b>0.999914</b> | <b>0.999914</b> |
| <b>MAL-BAYES-III HMC</b>    | <b>0.999914</b> | <b>0.999914</b>   | <b>0.999914</b> | <b>0.999914</b> |
| Histogram Gradient Boosting | 0.999741        | 0.999743          | 0.999741        | 0.999741        |
| MLP                         | 0.999741        | 0.999742          | 0.999741        | 0.999741        |
| Calibrated Linear SVM       | 0.999224        | 0.999226          | 0.999224        | 0.999224        |
| Logistic Regression         | 0.998621        | 0.998622          | 0.998621        | 0.998621        |

**Panel B. Probability-quality and calibration metrics**

| Method                      | NLL             | Brier           | ECE             | MCE             |
|-----------------------------|-----------------|-----------------|-----------------|-----------------|
| Random Forest               | 0.001138        | 0.000409        | 0.000580        | 0.420000        |
| Temperature-scaled DNN      | 0.002381        | <b>0.000172</b> | 0.000086        | <b>0.000086</b> |
| MC Dropout                  | 0.001069        | 0.000201        | 0.000290        | 0.355073        |
| Deep Ensemble               | <b>0.000960</b> | 0.000181        | 0.000199        | 0.204194        |
| <b>MAL-BAYES-III HMC</b>    | 0.000963        | 0.000193        | <b>0.000074</b> | 0.320981        |
| Histogram Gradient Boosting | 0.001192        | 0.000347        | 0.000146        | 0.614642        |
| MLP                         | 0.001086        | 0.000422        | 0.000439        | 0.619948        |
| Calibrated Linear SVM       | 0.004198        | 0.001128        | 0.002614        | 0.377395        |
| Logistic Regression         | 0.004508        | 0.002247        | 0.000454        | 0.630516        |

*Note. Higher values are better for the discrimination metrics in Panel A; lower values are better for NLL, Brier, ECE, and MCE in Panel B. Bold values identify the best result or a tied best result in each column.*

Because the leading methods were tied on the headline classification measures, their probability behaviour provides the more informative comparison. Deep Ensemble produced the lowest NLL at 0.000960, only 0.000003 below the HMC value of 0.000963. The temperature-scaled DNN achieved the lowest Brier score, 0.000172, followed by Deep Ensemble at 0.000181 and MAL-BAYES-III HMC at 0.000193. In contrast, HMC achieved the lowest ECE of the entire binary comparison at 0.000074, slightly below the temperature-scaled DNN value of 0.000086 and the HGB value of 0.000146.

The very low HMC ECE should not be read as uniform superiority across every calibration statistic. Its MCE was 0.320981, whereas the temperature-scaled DNN achieved 0.000086. The binary HMC result therefore combines exceptionally small average calibration error with a substantially larger worst-bin deviation under the fixed-bin MCE definition. This contrast is especially important in a near-perfect classification regime, where aggregate calibration measures can be dominated by the large mass of confidently correct observations while a small calibration bin still determines the maximum deviation.

The binary reliability curves in Figure 7 visualize this distinction. Near-ceiling prediction concentrates most observations at very high confidence, so the global reliability pattern appears strong even though a sparsely populated bin can still determine the reported MCE.

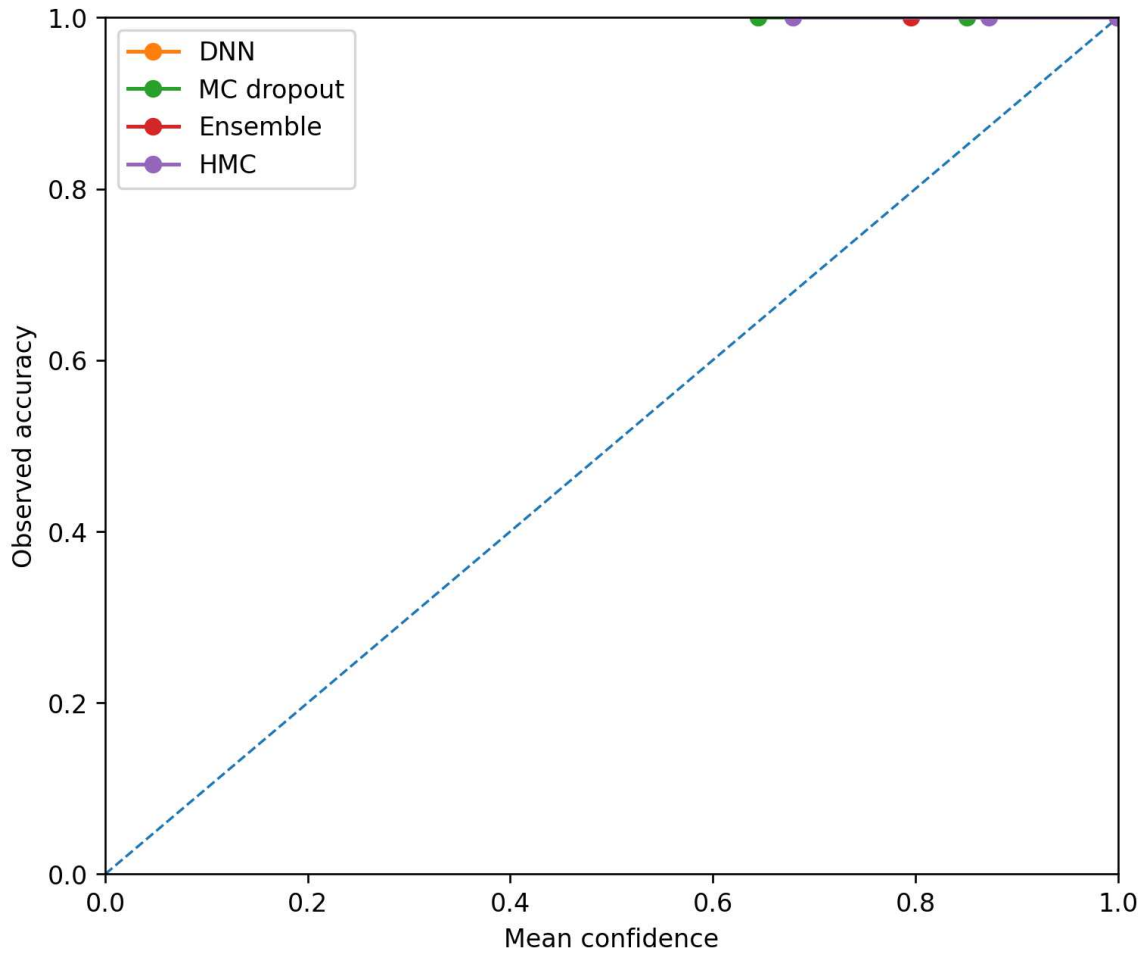


**Figure 7. Binary reliability comparison for the temperature-scaled DNN, MC Dropout, Deep Ensemble, and MAL-BAYES-III HMC.**

The malware-class results in Table 7 clarify the direction of the remaining errors. MAL-BAYES-III HMC achieved malware precision of 0.999826, recall of 1.000000, and F1 of 0.999913 across all 5,757 malware observations. Random Forest, the temperature-scaled DNN, MC Dropout, and Deep Ensemble produced the same malware-class values. Because malware recall was 1.000000, HMC did not miss any malware observation in the locked test; combined with its precision and one total classification error, the residual error corresponds to a single Benign observation classified as Malware.

Figure 6 confirms the error direction directly: all 5,757 Malware observations were correctly classified, 5,846 of 5,847 Benign observations were correctly retained as Benign, and the sole error was one Benign observation predicted as Malware.

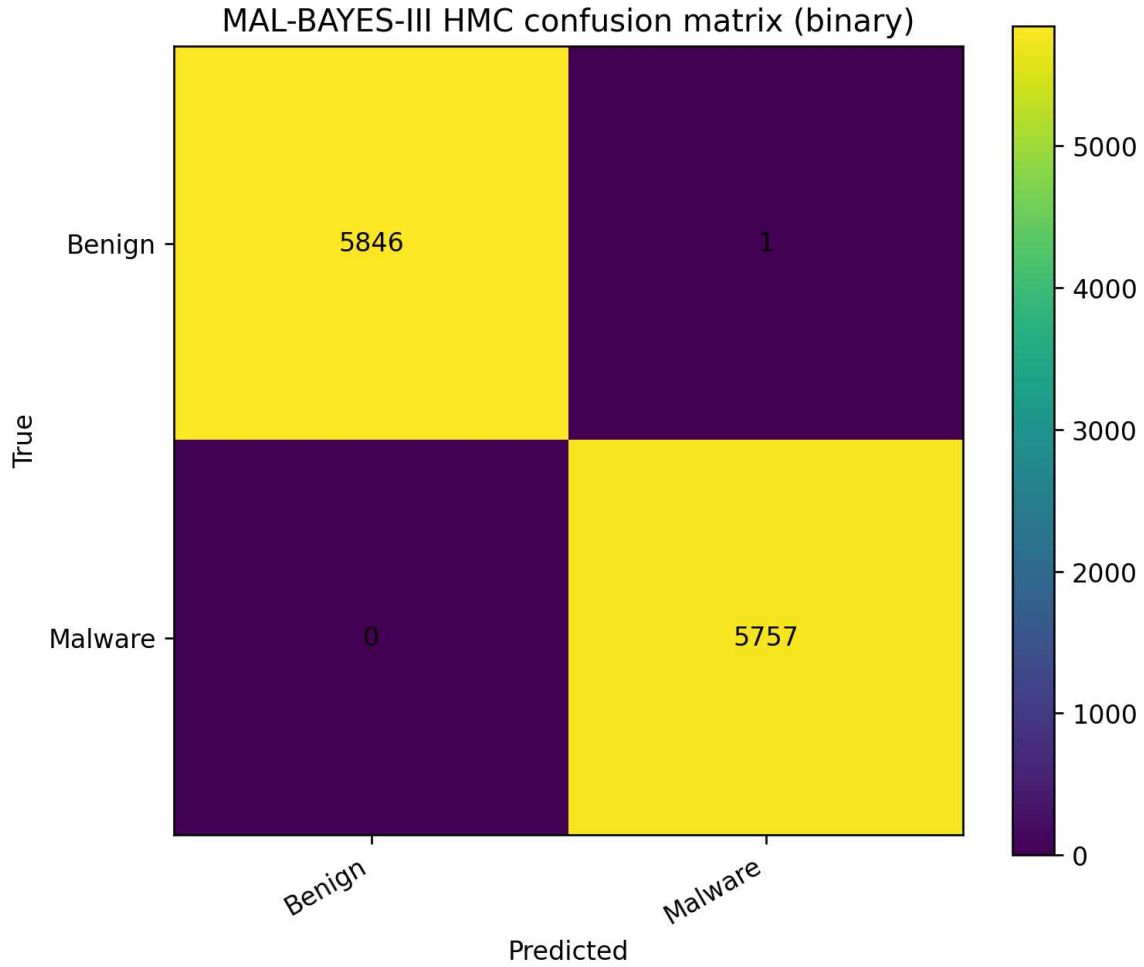


Figure 6. MAL-BAYES-III HMC binary confusion matrix for the locked-test set.

Table 7. Binary Malware-Class Precision, Recall, and F1

| Method                      | Precision (Malware) | Recall (Malware) | F1 (Malware)    | Support |
|-----------------------------|---------------------|------------------|-----------------|---------|
| Random Forest               | 0.999826            | 1.000000         | 0.999913        | 5,757   |
| Temperature-scaled DNN      | 0.999826            | 1.000000         | 0.999913        | 5,757   |
| MC Dropout                  | 0.999826            | 1.000000         | 0.999913        | 5,757   |
| Deep Ensemble               | 0.999826            | 1.000000         | 0.999913        | 5,757   |
| <b>MAL-BAYES-III HMC</b>    | <b>0.999826</b>     | <b>1.000000</b>  | <b>0.999913</b> | 5,757   |
| Histogram Gradient Boosting | 0.999479            | 1.000000         | 0.999740        | 5,757   |
| MLP                         | 0.999653            | 0.999826         | 0.999739        | 5,757   |
| Calibrated Linear SVM       | 0.998958            | 0.999479         | 0.999219        | 5,757   |
| Logistic Regression         | 0.998437            | 0.998784         | 0.998611        | 5,757   |

Thus, the binary task confirms near-ceiling discrimination and the lowest aggregate ECE for MAL-BAYES-III HMC, but the large HMC MCE and widespread ceiling effect preclude claims of uniformly superior calibration or binary classification accuracy.

#### 4.4 Posterior Uncertainty and Selective Prediction

The multiclass task provided the more informative test of whether uncertainty tracked prediction difficulty. Table 8 compares predictive entropy, mutual information, and predicted-class interval width for MC Dropout, the Deep Ensemble, and MAL-BAYES-III HMC. Across all three HMC measures, erroneous

predictions were more uncertain than correct predictions. Predictive entropy showed the largest separation: mean entropy increased from 0.260889 for correct HMC predictions to 0.895055 for errors, with a Spearman uncertainty-error association of 0.516926 and an error-detection AUROC of 0.889710.

**Table 8. Multiclass Uncertainty-Error Analysis**

| Method                   | Uncertainty measure  | Mean correct | Mean error | Spearman (error) | Error-detection AUROC |
|--------------------------|----------------------|--------------|------------|------------------|-----------------------|
| MC Dropout               | Predictive entropy   | 0.296642     | 0.943002   | 0.509836         | 0.884874              |
| MC Dropout               | Mutual information   | 0.011874     | 0.026011   | 0.382066         | 0.788421              |
| MC Dropout               | CI width (predicted) | 0.118543     | 0.310908   | 0.419716         | 0.816843              |
| Deep Ensemble            | Predictive entropy   | 0.280330     | 0.925595   | 0.514809         | 0.890492              |
| Deep Ensemble            | Mutual information   | 0.004674     | 0.013133   | 0.435192         | 0.830101              |
| Deep Ensemble            | CI width (predicted) | 0.048954     | 0.141689   | 0.439245         | 0.833174              |
| <b>MAL-BAYES-III HMC</b> | Predictive entropy   | 0.260889     | 0.895055   | 0.516926         | 0.889710              |
| <b>MAL-BAYES-III HMC</b> | Mutual information   | 0.001412     | 0.003077   | 0.403145         | 0.803930              |
| <b>MAL-BAYES-III HMC</b> | CI width (predicted) | 0.035601     | 0.105163   | 0.463442         | 0.849388              |

*Note. Higher positive Spearman association and higher error-detection AUROC indicate stronger ordering of erroneous predictions toward greater uncertainty. HMC interval width is a 95% posterior credible-interval width; MC Dropout and Deep Ensemble interval widths are empirical dispersion measures.*

Within MAL-BAYES-III HMC, predicted-class credible-interval width was the second-strongest error-ranking measure. Its mean increased from 0.035601 on correct cases to 0.105163 on errors, yielding Spearman correlation 0.463442 and AUROC 0.849388. Mutual information also increased on errors, from 0.001412 to 0.003077, but produced a lower AUROC of 0.803930. Across methods, the Deep Ensemble predictive-entropy AUROC of 0.890492 was marginally above the HMC entropy AUROC, whereas HMC produced the strongest displayed Spearman association for predictive entropy and the strongest interval-width AUROC. The results therefore identify predictive entropy as the most consistently informative HMC uncertainty score for multiclass error identification, while credible-interval width provides a complementary posterior-dispersion signal.

This ranking translated directly into selective prediction. Figure 4 orders locked-test observations from lowest to highest predictive entropy and recomputes error after progressively deferring uncertain cases. At full coverage, the three curves begin near their ordinary multiclass error rates. As coverage decreases, selective error falls almost monotonically for all three methods and approaches zero when roughly one-half of the locked-test observations are retained. The HMC curve closely tracks MC Dropout and the Deep Ensemble throughout the coverage range, indicating that entropy concentrates a substantial share of the multiclass mistakes among the records ranked as more uncertain.

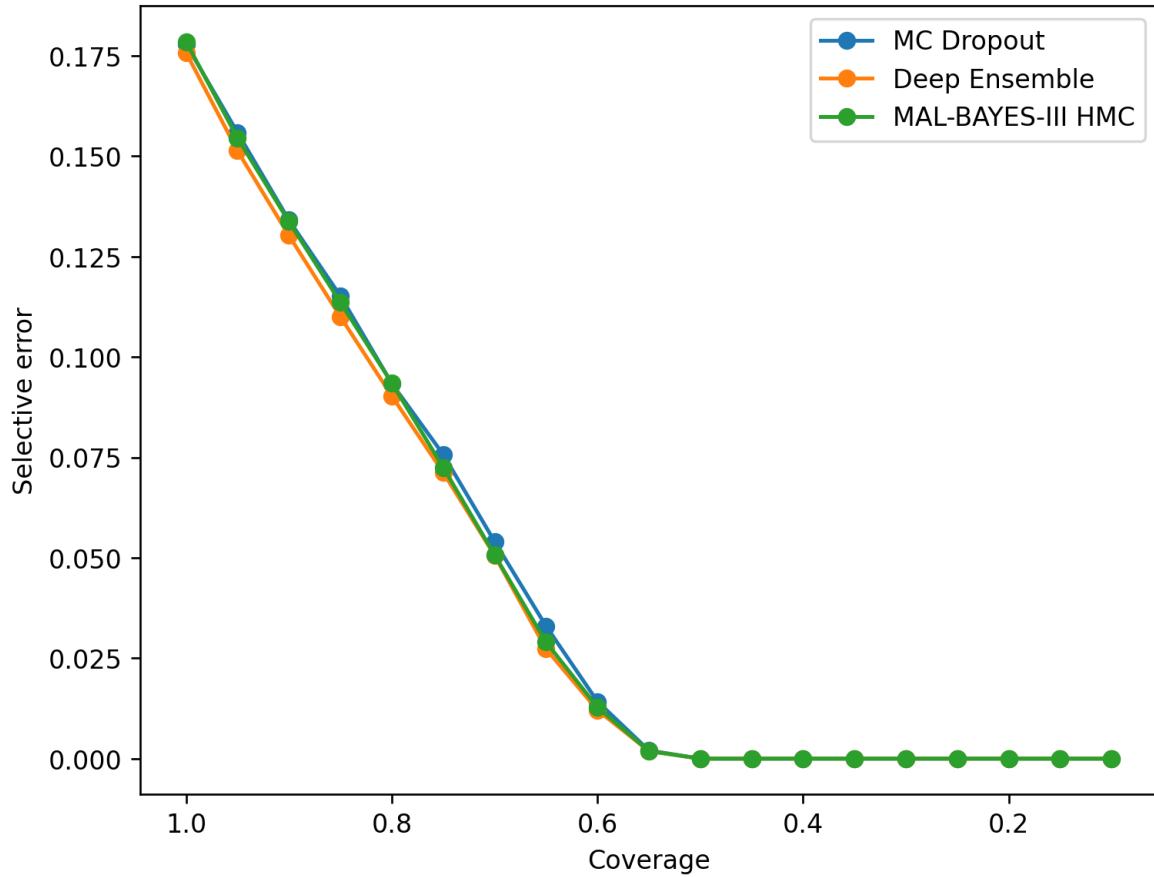


**Figure 4. Multiclass risk-coverage curves based on predictive-entropy ranking for MC Dropout, Deep Ensemble, and MAL-BAYES-III HMC.**

The binary uncertainty results require a different interpretation because the locked test contained only one HMC classification error. Table 9 shows that the HMC error received lower, rather than higher, uncertainty than the average correct prediction for all three measures. Predictive entropy was 0.000994 for the single error versus 0.001257 across correct predictions, giving a Spearman association of -0.011450 and error-detection AUROC of 0.143928. Mutual information and interval width exhibited the same direction, with AUROCs of 0.068344 and 0.152547, respectively. These values do not provide evidence that HMC uncertainty identified the lone binary error.

**Table 9. Binary Uncertainty-Error Analysis**

| Method                   | Uncertainty measure  | Mean correct | Mean error | Spearman (error) | Error-detection AUROC |
|--------------------------|----------------------|--------------|------------|------------------|-----------------------|
| MC Dropout               | Predictive entropy   | 0.002907     | 0.003191   | 0.015826         | 0.992157              |
| MC Dropout               | Mutual information   | 0.000179     | 1.100e-05  | 0.015854         | 0.993019              |
| MC Dropout               | CI width (predicted) | 0.000723     | 0.000300   | 0.015884         | 0.993967              |
| Deep Ensemble            | Predictive entropy   | 0.002468     | 0.003594   | 0.015976         | 0.996811              |
| Deep Ensemble            | Mutual information   | 4.900e-05    | 1.700e-05  | 0.015940         | 0.995691              |
| Deep Ensemble            | CI width (predicted) | 0.000276     | 0.000323   | 0.015955         | 0.996122              |
| <b>MAL-BAYES-III HMC</b> | Predictive entropy   | 0.001257     | 0.000994   | -0.011450        | 0.143928              |
| <b>MAL-BAYES-III HMC</b> | Mutual information   | 0.000145     | 7.200e-05  | -0.013881        | 0.068344              |
| <b>MAL-BAYES-III HMC</b> | CI width (predicted) | 0.000854     | 0.000513   | -0.011173        | 0.152547              |

*Note. Binary error-detection statistics are based on extremely rare errors and are therefore unstable. High or low AUROC values in this setting should not be interpreted as robust uncertainty-ranking evidence.*

The apparently high binary error-detection AUROCs for MC Dropout and the Deep Ensemble are subject to the same small-error limitation. With essentially one misclassified observation, the AUROC is driven by the rank assigned to that single case and can change sharply with a different residual error. Figure 8 makes this instability visible. MC Dropout and the Deep Ensemble remove their residual error immediately after a small reduction in coverage, whereas the HMC error remains in the retained low-entropy set over much of the curve. Its selective error therefore rises as coverage is reduced and the denominator becomes smaller, before falling to zero only at the lowest displayed coverage. This non-monotonic pattern is consistent with the negative HMC uncertainty-error associations in Table 9 rather than with effective binary error triage.

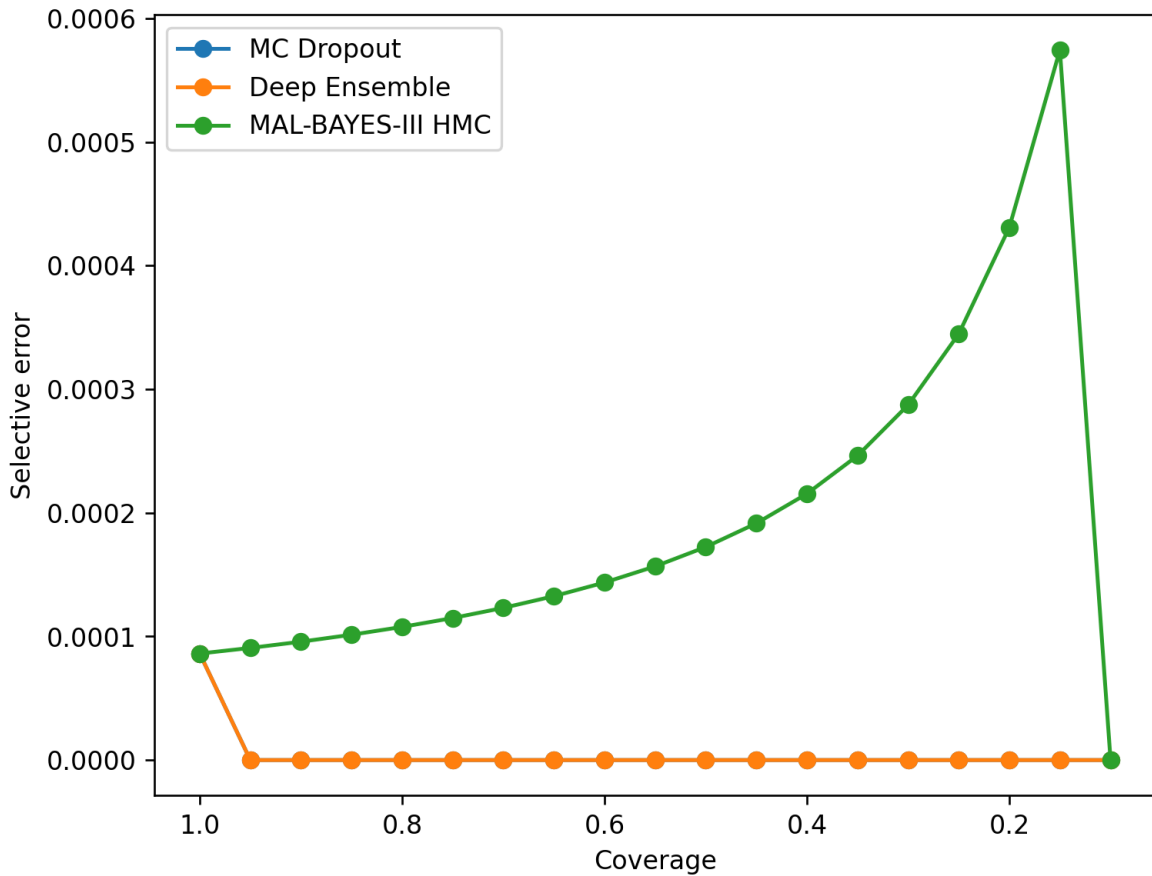


**Figure 8. Binary risk-coverage curves based on predictive-entropy ranking for MC Dropout, Deep Ensemble, and MAL-BAYES-III HMC.**

Accordingly, the uncertainty evidence is task-dependent. The four-category experiment provides substantive evidence that HMC predictive entropy, credible-interval width, and mutual information are elevated on difficult cases and can support descriptive selective review. The binary experiment is too close to perfect classification to provide a stable test of error ranking, and its HMC curve specifically shows that the single residual error was not assigned high entropy. The selective-prediction curves are therefore interpreted as locked-test diagnostics rather than operational abstention thresholds; prospective forensic deployment would require threshold selection on validation data and transfer to future cases without retuning (Yumlembam et al., 2025).

#### 4.5 HMC Convergence and Posterior Sampling Reliability

The posterior uncertainty results were interpreted only after confirming that the retained HMC states satisfied the predeclared convergence and numerical-quality criteria. Table 10 summarizes the final diagnostic

results for the four-category and binary Bayesian heads. Both tasks met the required maximum R-hat threshold of 1.01, exceeded the minimum bulk effective sample size of 400, and recorded zero divergences. The retained posterior therefore passed the numerical gates used to determine whether the HMC output was suitable for subsequent predictive and uncertainty analysis.

**Table 10. HMC Posterior Diagnostic Summary**

**Panel A. Convergence and effective-sampling diagnostics**

| Task       | Max R-hat | Min Bulk ESS | Min Tail ESS | Max MCSE Mean | Mean Acceptance |
|------------|-----------|--------------|--------------|---------------|-----------------|
| Multiclass | 1.010000  | 674          | 1,075        | 0.018000      | 0.930484        |
| Binary     | 1.000000  | 3,215        | 3,271        | 0.017000      | 0.968234        |

**Panel B. Sampler behaviour and retained-chain configuration**

| Task       | Divergence Fraction | Mean Leapfrog Steps | Chains | Warm-up / Chain | Retained / Chain |
|------------|---------------------|---------------------|--------|-----------------|------------------|
| Multiclass | 0.000000            | 123.750000          | 4      | 1,200           | 1,200            |
| Binary     | 0.000000            | 9.750000            | 4      | 1,200           | 1,200            |

*Note. Each task retained 4,800 post-warm-up HMC states in total. The predeclared readiness gates required maximum R-hat  $\leq 1.01$ , minimum bulk ESS  $\geq 400$ , and divergence fraction  $\leq 0.001$ .*

The multiclass sampler satisfied the convergence criteria but operated closer to the declared limits than the binary sampler. Its maximum R-hat was exactly 1.010000, while the minimum bulk ESS and tail ESS were 674 and 1,075, respectively. These values remain above the effective-sampling threshold but show that the four-category posterior was more difficult to explore. By comparison, the binary head achieved maximum R-hat of 1.000000, minimum bulk ESS of 3,215, and minimum tail ESS of 3,271. The maximum MCSE of the posterior means was similar across tasks (0.018000 multiclass and 0.017000 binary), indicating that the principal distinction lay in sampling efficiency and trajectory effort rather than in a large difference in the reported Monte Carlo error summary.

Sampler behaviour reinforced this contrast. Mean acceptance probability was 0.930484 for the multiclass posterior and 0.968234 for the binary posterior, both above the 0.90 adaptation target and within the declared numerical-quality range. No post-warm-up divergence was recorded for either task. However, the multiclass sampler required an average of 123.75 leapfrog steps per retained iteration compared with only 9.75 for the binary head, a difference of roughly 12.7-fold. The larger trajectory effort, together with the lower multiclass ESS, indicates substantially more demanding posterior geometry for category-level inference even though the sampler remained within the predefined readiness gates.

Figure 5 provides the visual trace diagnostic for the three free multiclass intercepts. The chain-specific density curves show substantial overlap, and the retained traces fluctuate around stable regions without an obvious sustained drift. This visual pattern is consistent with the numerical R-hat and ESS results, although the trace display is limited to the intercepts and therefore complements rather than replaces the parameter-wide diagnostics summarized in Table 10.

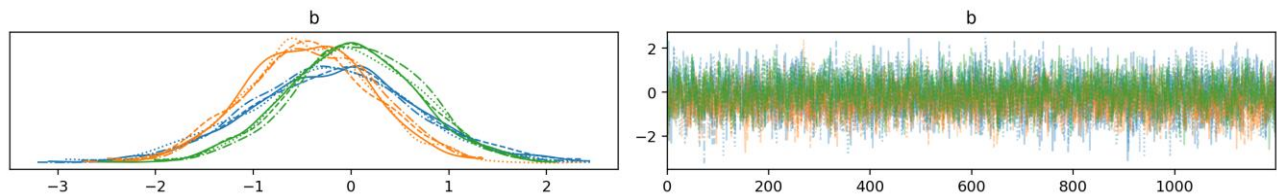


**Figure 5. Multiclass HMC intercept posterior-density and retained-chain trace diagnostics.**

The corresponding binary intercept diagnostic is shown in Figure 9. The four chain densities are closely aligned and the retained trace exhibits stable fluctuation across the sampled iterations, consistent with the

binary maximum R-hat of 1.000000 and substantially larger bulk and tail ESS. The binary trace is consequently visually compatible with the stronger numerical sampling efficiency reported in Table 10.

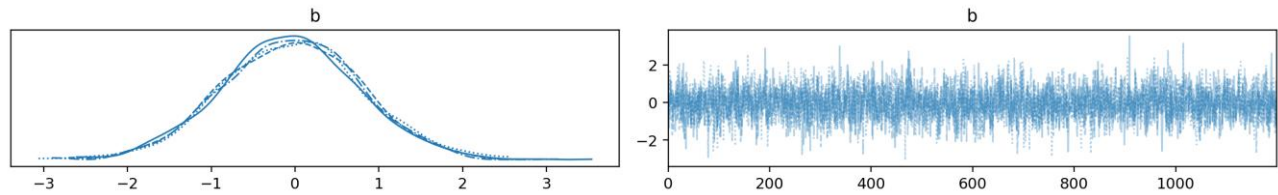


**Figure 9. Binary HMC intercept posterior-density and retained-chain trace diagnostics.**

Overall, the HMC diagnostics support the numerical interpretability of both posterior heads. The binary posterior was sampled more efficiently, whereas the multiclass posterior required much greater leapfrog effort and reached the R-hat boundary of the acceptance criterion. The absence of divergences, adequate effective sample sizes, finite MCSE summaries, and stable intercept traces nevertheless provide convergent evidence that the retained HMC draws were sufficiently reliable for the posterior predictive and uncertainty analyses reported in the preceding results.

#### 4.6 Cross-Study Comparison with CIC-MalMem-2022 State-of-the-Art Methods

To position MAL-BAYES-III against the published CIC-MalMem-2022 literature, Table 11 supplements the same-protocol comparisons in Tables 5 and 6 with a cross-study state-of-the-art comparison. This distinction is important. Tables 5 and 6 compare models rerun under one common MAL-BAYES-III data contract, whereas the studies in Table 11 used their own feature sets, preprocessing, balancing procedures, split rules, and evaluation designs. The reported accuracies are therefore literature-level reference points rather than controlled head-to-head estimates.

**Table 11. Cross-Study State-of-the-Art Comparison on CIC-MalMem-2022**

**Panel A. Reported predictive performance**

| Study                 | Method                           | Binary Accuracy (%) | 4-Class Accuracy (%) | 16-Class Accuracy (%) | Protocol / scope note                                          |
|-----------------------|----------------------------------|---------------------|----------------------|-----------------------|----------------------------------------------------------------|
| Carrier et al. (2022) | Stack ensemble                   | 99.00               | NR                   | NR                    | Original CIC-MalMem-2022 benchmark study                       |
| Dener et al. (2022)   | Logistic Regression              | 99.97               | NR                   | NR                    | Published CIC-MalMem-2022 protocol                             |
| Shafin et al. (2023)  | CompactCBL                       | 99.92               | 84.22                | 71.42                 | Resource-constrained neural detector                           |
| Shafin et al. (2023)  | RobustCBL                        | 99.96               | 84.56                | 72.60                 | Larger compact neural detector                                 |
| Roy et al. (2023)     | MalHyStack                       | 99.98               | 85.04                | 70.29                 | Hybrid stacking; 16 selected features                          |
| <b>MAL-BAYES-III</b>  | <b>HMC posterior neural head</b> | 99.9914             | 82.1527              | NR                    | 52 predictors; duplicate controls; 64/16/20 locked-test design |

**Panel B. Methodological capability comparison**

| Study                    | Posterior Inference   | Calibration        | MCMC Diagnostics              | Case-Level Uncertainty    | Selective Prediction  | Shift / Novelty Evaluation           |
|--------------------------|-----------------------|--------------------|-------------------------------|---------------------------|-----------------------|--------------------------------------|
| Carrier et al. (2022)    | No                    | NR                 | No                            | No                        | No                    | No                                   |
| Shafin et al. (2023)     | No                    | NR                 | No                            | No                        | No                    | No                                   |
| Roy et al. (2023)        | No                    | NR                 | No                            | No                        | No                    | No                                   |
| Hossain & Islam (2024)   | No                    | NR                 | No                            | No                        | No                    | No                                   |
| Maniriho et al. (2024)   | No                    | NR                 | No                            | No                        | No                    | Temporal evaluation                  |
| Madamidola et al. (2025) | No                    | NR                 | No                            | No                        | No                    | Unseen-subtype evaluation            |
| <b>MAL-BAYES-III</b>     | <b>HMC final head</b> | ECE/MCE; NLL/Brier | R-hat, ESS, MCSE, divergences | Entropy, MI, 95% CI width | Entropy risk-coverage | Controlled Gaussian + masking shifts |

*Note. NR = not reported in the cited study as a primary evaluation quantity. Cross-study accuracy values are reported as published and should not be interpreted as if they were produced under identical preprocessing, feature selection, balancing, partitioning, or leakage-control conditions. MAL-BAYES-III controlled shifts are synthetic sensitivity tests and are not equivalent to temporal or genuinely unseen-subtype evaluation.*

The literature comparison shows that MAL-BAYES-III is positioned differently across the two analytical tasks. Its binary locked-test accuracy of 99.9914% lies at the near-ceiling level already reported for CIC-MalMem-2022, including MalHyStack and the compact models of Shafin et al. (2023). The four-category result is less competitive in raw accuracy: MAL-BAYES-III achieved 82.1527%, compared with 84.22%-84.56% for CompactCBL/RobustCBL and 85.04% for MalHyStack (Roy et al., 2023; Shafin et al., 2023). Hossain and Islam (2024) reported still higher multiclass results under a substantially different SMOTE-assisted protocol. Consequently, the cross-study evidence does not support a claim that MAL-BAYES-III establishes a new four-category accuracy state of the art.

Panel B instead identifies the methodological dimension on which MAL-BAYES-III extends the conventional CIC-MalMem-2022 literature. The earlier benchmark, compact-neural, stacking, and ensemble studies principally report point-prediction performance, while MAL-BAYES-III couples a sampled HMC final-head posterior with calibration metrics, multiple-chain convergence diagnostics, posterior credible intervals, predictive entropy, mutual information, uncertainty-error analysis, and entropy-ranked selective prediction. This distinction does not make the framework uniformly superior: Maniriho et al. (2024) explicitly evaluated temporal change, and Madamidola et al. (2025) evaluated malware subtypes absent from training, whereas the present study's stress experiments are controlled Gaussian and masking perturbations rather than genuine unseen-malware or temporal validation.

The resulting state-of-the-art position is therefore two-part. MAL-BAYES-III retains near-ceiling binary detection and competitive, but not leading, category-level discrimination while adding a diagnostically audited Bayesian uncertainty layer that is largely absent from the directly comparable CIC-MalMem-2022 studies. For this reason, its contribution is better interpreted as extending the evidential content of memory-forensic classification—from point predictions toward calibrated posterior probabilities and uncertainty-aware review—rather than as replacing the strongest deterministic ensemble methods on multiclass accuracy alone.

#### 4.7 Statistical Comparison, Controlled-Shift Sensitivity, and Final Experimental Readiness

The analysis examined whether the small predictive difference between MAL-BAYES-III HMC and the temperature-scaled DNN was stable under paired resampling and how both frozen predictors responded to controlled feature perturbation. Table 12 reports the 1,000-repetition paired bootstrap for macro-F1. In the multiclass task, the mean HMC-minus-DNN difference was -0.000220, with a percentile 95% interval from -0.004360 to 0.004143 and a directional proportion  $P(\Delta > 0)$  of 0.455. Because the interval spans zero and the directional proportion is close to one half, the locked-test resampling evidence does not support a stable macro-F1 advantage for either neural predictor.

**Table 12. Paired Bootstrap Comparison of MAL-BAYES-III HMC and Temperature-Scaled DNN**

| Task       | Metric   | Mean $\Delta$ (HMC-DNN) | 95% Bootstrap Interval | $P(\Delta > 0)$ | Repetitions | Interpretation                   |
|------------|----------|-------------------------|------------------------|-----------------|-------------|----------------------------------|
| Multiclass | Macro-F1 | -0.000220               | [-0.004360, 0.004143]  | 0.455           | 1,000       | Interval spans zero              |
| Binary     | Macro-F1 | 0.000000                | [0.000000, 0.000000]   | 0.000           | 1,000       | All paired differences were zero |

*Note.*  $\Delta$  is defined as MAL-BAYES-III HMC macro-F1 minus temperature-scaled DNN macro-F1.  $P(\Delta > 0)$  is a directional bootstrap proportion, not a frequentist  $p$ -value. In the binary task,  $P(\Delta > 0) = 0$  because every paired bootstrap difference was exactly zero.

The binary comparison was even less separative. The two methods produced identical macro-F1 under every bootstrap resample, giving a mean difference of 0.000000 and a degenerate percentile interval [0.000000, 0.000000]. The directional proportion of zero therefore reflects complete equality of the resampled macro-F1 values rather than evidence that the DNN is superior. Together with the near-ceiling binary metrics reported in Section 4.3, this result confirms that binary classification performance provides no meaningful statistical basis for ranking these two neural predictors.

Controlled-shift results are summarized in Table 13. For the four-category task, both predictors degraded as Gaussian perturbation increased. The temperature-scaled DNN fell from its clean macro-F1 of 0.730462 to 0.546748 at severity 0.35, while HMC fell from 0.730271 to 0.543845. ECE simultaneously increased to 0.142381 for the DNN and 0.151810 for HMC. Across all four Gaussian severities, the DNN retained a small discrimination and calibration advantage, so these perturbations do not support a claim that the HMC posterior head is intrinsically more robust.

**Table 13. Controlled-Shift Sensitivity of the Temperature-Scaled DNN and MAL-BAYES-III HMC**

**Panel A. Multiclass Gaussian perturbation**

| Severity | DNN Accuracy | HMC Accuracy | DNN Macro-F1 | HMC Macro-F1 | DNN ECE  | HMC ECE  |
|----------|--------------|--------------|--------------|--------------|----------|----------|
| 0.05     | 0.796277     | 0.792744     | 0.691348     | 0.686784     | 0.015406 | 0.030947 |
| 0.10     | 0.775595     | 0.773009     | 0.660287     | 0.657068     | 0.038766 | 0.053925 |
| 0.20     | 0.738883     | 0.734402     | 0.605326     | 0.599599     | 0.088844 | 0.103088 |
| 0.35     | 0.699931     | 0.696398     | 0.546748     | 0.543845     | 0.142381 | 0.151810 |

**Panel B. Multiclass feature masking**

| Severity | DNN Accuracy | HMC Accuracy | DNN Macro-F1 | HMC Macro-F1 | DNN ECE  | HMC ECE  |
|----------|--------------|--------------|--------------|--------------|----------|----------|
| 0.05     | 0.770941     | 0.772492     | 0.654223     | 0.657070     | 0.047569 | 0.057040 |
| 0.10     | 0.741899     | 0.739142     | 0.611737     | 0.609086     | 0.086938 | 0.098181 |
| 0.20     | 0.705963     | 0.707428     | 0.557217     | 0.561750     | 0.140595 | 0.142390 |
| 0.35     | 0.668907     | 0.667098     | 0.502202     | 0.505824     | 0.198625 | 0.191845 |

Feature masking produced a different, but still mixed, multiclass pattern. At severities 0.05 and 0.20, HMC achieved slightly higher accuracy and macro-F1 than the DNN; at 0.35 it retained a slightly higher macro-F1 (0.505824 versus 0.502202) despite marginally lower accuracy. At this highest masking severity, HMC also had a lower ECE (0.191845 versus 0.198625) and lower NLL (1.091913 versus 1.163071). These local advantages are not consistent across all masking levels and therefore indicate condition-specific sensitivity differences rather than a general robustness dominance.

**Panel C. Binary Gaussian perturbation**

| Severity | DNN Accuracy | HMC Accuracy | DNN Macro-F1 | HMC Macro-F1 | DNN ECE  | HMC ECE  |
|----------|--------------|--------------|--------------|--------------|----------|----------|
| 0.05     | 0.999741     | 0.999741     | 0.999741     | 0.999741     | 0.000259 | 0.000212 |
| 0.10     | 0.999741     | 0.999655     | 0.999741     | 0.999655     | 0.000259 | 0.000235 |
| 0.20     | 0.999397     | 0.999397     | 0.999397     | 0.999397     | 0.000603 | 0.000502 |
| 0.35     | 0.997070     | 0.996725     | 0.997070     | 0.996725     | 0.002920 | 0.002403 |

**Panel D. Binary feature masking**

| Severity | DNN Accuracy | HMC Accuracy | DNN Macro-F1 | HMC Macro-F1 | DNN ECE  | HMC ECE  |
|----------|--------------|--------------|--------------|--------------|----------|----------|
| 0.05     | 0.999483     | 0.999397     | 0.999483     | 0.999397     | 0.000517 | 0.000380 |
| 0.10     | 0.998707     | 0.998707     | 0.998707     | 0.998707     | 0.001302 | 0.001242 |
| 0.20     | 0.996294     | 0.995777     | 0.996294     | 0.995777     | 0.003698 | 0.003026 |
| 0.35     | 0.985350     | 0.985091     | 0.985348     | 0.985090     | 0.014718 | 0.011091 |

*Note. The stress conditions are synthetic sensitivity tests applied to the frozen predictors. They are not unseen-malware, open-set, temporal, or adversarial evaluations. The table reports accuracy, macro-F1, and ECE; the exported stress results also contain NLL, Brier score, AUROC/AUPRC, and class-specific measures.*

Binary discrimination remained extremely high even at the strongest perturbation. Under Gaussian severity 0.35, accuracy remained 0.997070 for the DNN and 0.996725 for HMC; under 35% feature masking, the corresponding values were 0.985350 and 0.985091. HMC was therefore not consistently superior in binary accuracy. Its probability quality was more resistant, however. Across every displayed binary Gaussian and masking condition, HMC had lower ECE than the temperature-scaled DNN. The difference was also

pronounced in NLL at the strongest shifts: Gaussian severity 0.35 gave NLL 0.016370 for HMC versus 0.078786 for the DNN, while 35% masking gave 0.062343 versus 0.369943. The controlled binary results therefore suggest greater preservation of probability quality under perturbation, without a corresponding advantage in classification accuracy.

The final integrity audit confirmed that the reported posterior results satisfied every hard release criterion. Table 14 condenses the 20 task-specific hard checks into their ten common gate definitions. Both tasks contained finite probabilities within the unit interval, probability rows summed to one within the  $10^{-5}$  tolerance, all aggregate HMC diagnostics were finite, no divergences occurred, R-hat and bulk ESS satisfied their thresholds, and both predictive-signal checks exceeded the required macro-F1 floor.

**Table 14. Final Experimental Readiness and Numerical-Integrity Gates**

| Check                            | Multiclass Observed    | Binary Observed        | Criterion               | Result      |
|----------------------------------|------------------------|------------------------|-------------------------|-------------|
| Probabilities finite             | True                   | True                   | Must be True            | PASS / PASS |
| Max probability-row sum error    | $3.366 \times 10^{-6}$ | $2.846 \times 10^{-6}$ | $\leq 1 \times 10^{-5}$ | PASS / PASS |
| Probability range                | [6.021e-07, 0.9999814] | [7.735e-05, 0.9999238] | Within [0,1] tolerance  | PASS / PASS |
| Aggregate HMC diagnostics finite | True                   | True                   | All finite              | PASS / PASS |
| Divergence fraction              | 0.000000               | 0.000000               | $\leq 0.001$            | PASS / PASS |
| Acceptance probability valid     | 0.930484               | 0.968234               | $> 0$ and $\leq 1$      | PASS / PASS |
| Maximum R-hat                    | 1.010000               | 1.000000               | $\leq 1.01$             | PASS / PASS |
| Minimum bulk ESS                 | 674                    | 3,215                  | $\geq 400$              | PASS / PASS |
| Acceptance target quality        | 0.930484               | 0.968234               | 0.50-1.00               | PASS / PASS |
| Minimum predictive signal        | 0.730271               | 0.999914               | Macro-F1 $\geq 0.50$    | PASS / PASS |

*Note. The original run gate contains ten hard checks for each task (20 checks total). All 20 passed, yielding FINAL STATUS: SUCCESS.*

Taken together, the final evidence does not indicate a statistically stable multiclass macro-F1 difference between HMC and its deterministic neural counterpart, and it does not support a universal robustness advantage for HMC under the controlled shifts. Instead, the stress analysis reveals a more specific pattern: multiclass performance degrades similarly for both neural predictors, whereas the binary HMC posterior tends to preserve NLL and ECE more effectively as synthetic perturbation increases. Combined with the convergence, calibration, and uncertainty findings reported earlier in Section 4, these results support MAL-BAYES-III primarily as an uncertainty-aware and diagnostically audited posterior extension to memory-forensic classification rather than as a uniformly more accurate or universally more robust classifier.

## 5. Discussion

### 5.1 Predictive–Calibration Trade-off and the Role of the Bayesian Neural Head

The principal implication of MAL-BAYES-III is that posterior inference should not be judged only by whether it raises classification accuracy. In the four-category task, deterministic tree ensembles remained more discriminative, whereas the HMC neural head stayed close to the deterministic and approximate neural comparators while producing the lowest aggregate calibration error. The Bayesian component is therefore understood as an evidential extension to a learned forensic representation than as a universal accuracy booster. This interpretation is consistent with Bayesian-neural research showing that posterior modelling changes the uncertainty content of prediction without guaranteeing superior classification performance (Izmailov et al., 2021; Andrade & Sato, 2025).

For digital-forensic use, this distinction matters because a high-performing classifier may still provide confidence values that are poorly aligned with empirical correctness. MAL-BAYES-III instead provides posterior probabilities, credible intervals, entropy, and mutual information that can be examined alongside the predicted label. The multiclass results showed that this additional uncertainty structure can identify difficult cases without materially sacrificing neural-level discrimination, which aligns with recent uncertainty-aware cybersecurity work that treats uncertainty as decision-support evidence rather than a substitute for predictive accuracy (Yang et al., 2024; Yumlembam et al., 2025). The binary task, by contrast, was so close to perfect classification that its single residual error could not support a stable test of uncertainty ranking; very low average calibration error should therefore not be mistaken for strong evidence of operational triage value in that saturated setting.

The partial-Bayesian architecture also limits what these uncertainty quantities mean. HMC samples only the compact final decision head, so the posterior describes uncertainty in the mapping from the frozen 24-dimensional embedding to class probabilities; uncertainty in the encoder weights is not propagated. This is a deliberate computational compromise rather than full-network Bayesian inference. Recent work on partially deterministic Bayesian networks supports such restricted designs as a tractable way to obtain Bayesian decision-layer uncertainty while avoiding the cost and difficult geometry of sampling an entire deep network (Andrade & Sato, 2025; Thiagarajan et al., 2025). Accordingly, the MAL-BAYES-III credible intervals and mutual information should be interpreted as conditional on the learned neural representation.

The cross-study comparison reinforces this positioning. Several CIC-MalMem-2022 systems report higher four-category accuracy, but they principally evaluate point prediction, whereas MAL-BAYES-III adds multiple-chain posterior diagnostics, calibration analysis, credible intervals, uncertainty-error association, and selective prediction within one leakage-controlled protocol. Its contribution is therefore methodological rather than an accuracy-record claim: it demonstrates how a tractable HMC neural head can extend memory-forensic classification from label prediction toward auditable posterior evidence and uncertainty-aware review. This is consistent with the wider direction of uncertainty-aware cybersecurity, where reliability and evidential strength are increasingly evaluated alongside discrimination (Doan et al., 2023, 2024; Yang et al., 2024).

### 5.2 Forensic Value of Posterior Uncertainty and Selective Prediction

The uncertainty analysis is most informative in the four-category task because category-level decisions contained enough errors to test whether uncertainty increased on difficult observations. Predictive entropy, credible-interval width, and mutual information all rose on erroneous MAL-BAYES-III predictions, indicating that the posterior did not merely attach dispersion to every case indiscriminately. Predictive entropy was the strongest overall ranking signal, while credible-interval width provided a complementary measure of posterior probability dispersion. This pattern is important for forensic interpretation because uncertainty becomes

operationally useful only when it is systematically related to cases in which the classification decision is less dependable (Yang et al., 2024; Yumlembam et al., 2025).

The three uncertainty quantities also describe different aspects of evidential ambiguity. Predictive entropy measures how diffuse the final posterior predictive distribution is across competing classes, whereas mutual information isolates the component associated with disagreement across sampled Bayesian-head parameters. Predicted-class credible-interval width, in turn, describes how much the posterior probability assigned to the selected class varies across retained HMC draws. Their unequal error-ranking performance therefore should not be interpreted as redundancy or failure. Rather, the results suggest that total predictive ambiguity was more strongly aligned with multiclass mistakes than parameter-sensitive disagreement alone, while interval width supplied additional evidence about posterior dispersion conditional on the frozen representation.

The risk-coverage behaviour translates this statistical association into a practical review concept. In the multiclass experiment, progressively deferring high-entropy observations reduced the error rate among retained cases, showing that posterior uncertainty can support prioritization when analyst attention is limited. The appropriate interpretation is not that MAL-BAYES-III autonomously resolves uncertain cases, but that it provides a defensible mechanism for distinguishing records that may be suitable for automated acceptance from those that warrant additional forensic scrutiny. This use of uncertainty as a triage signal is consistent with recent cybersecurity studies that combine uncertainty estimates with rejection, escalation, or conformal decision mechanisms to reduce risky automated decisions (Doan et al., 2023, 2024; Yumlembam et al., 2025).

The binary task demonstrates why selective-prediction claims must remain task-dependent. Because only one HMC binary prediction was wrong, the uncertainty ranking was determined almost entirely by the position of that single case; in fact, the residual error received relatively low HMC uncertainty and therefore remained in the retained set through much of the risk-coverage curve. This does not negate the strong binary classification result, but it prevents the study from claiming that HMC uncertainty provides reliable binary error triage on this benchmark. More broadly, near-perfect classification can make uncertainty-ranking metrics unstable even when posterior probabilities are well behaved, because there are too few failures to estimate how uncertainty behaves across heterogeneous error types.

For deployment, the present risk-coverage curves should therefore be viewed as descriptive evidence rather than as ready-to-use abstention rules. The curves were calculated directly on the locked test after all models were frozen, so choosing a coverage or entropy threshold from them would reuse evaluation data for operational tuning. A forensic system would instead require thresholds selected on validation data and then transferred unchanged to future investigations, preferably under temporal, source, and malware-family shifts. The current findings nevertheless establish the key prerequisite for such a design in the multiclass setting: uncertainty produced by the compact HMC head is meaningfully associated with decision difficulty and can order cases in a way that reduces retained risk.

### **5.3 HMC Reliability, Statistical Stability, and Controlled-Shift Behaviour**

The HMC diagnostics support the numerical credibility of the reported posterior results. Both tasks satisfied the predefined R-hat, effective-sample-size, and divergence criteria, although the multiclass posterior required substantially greater leapfrog effort and operated closer to the convergence boundary. The binary sampler was more efficient. These findings justify interpretation of the retained posterior draws while also showing that category-level posterior geometry was materially more demanding (Vehtari et al., 2021).

The paired bootstrap further showed that HMC did not produce a stable macro-F1 advantage over the temperature-scaled DNN. The multiclass interval spanned zero, while the binary differences were identically zero across resamples. Accordingly, MAL-BAYES-III should not be presented as statistically superior in

predictive accuracy; its added value lies in calibrated posterior inference and uncertainty information rather than a consistent gain in macro-F1.

Controlled Gaussian perturbation and feature masking produced mixed sensitivity patterns. Multiclass degradation was similar for HMC and the DNN, whereas the binary HMC probabilities generally retained lower ECE and NLL under stronger perturbation despite no consistent accuracy advantage. These experiments therefore indicate condition-specific probability resilience, not universal robustness. Because the shifts were synthetic rather than adversarial, temporal, or genuinely unseen-malware evaluations, stronger robustness claims are not supported (Feng et al., 2024; Maniriho et al., 2024; Madamidola et al., 2025).

## 5.4 Limitations and Generalisability

The main limitation is external validity. MAL-BAYES-III was evaluated only on CIC-MalMem-2022 using a category-stratified locked-test split in which all 16 audited groups were represented in training. Although exact cross-partition duplicates were removed, very close near-neighbours remained. These conditions do not negate the reported within-benchmark results, but they restrict inference about unseen malware families, new acquisition sources, or temporal drift. Such claims require explicit group-held-out, temporal, or unseen-subtype evaluation (Maniriho et al., 2024; Madamidola et al., 2025).

A second limitation follows from the partial-Bayesian design. HMC sampled only the compact final head, using training-only posterior subsets of 12,000 multiclass and 6,000 binary observations, while the encoder remained fixed. The resulting credible intervals and mutual information therefore quantify decision-head uncertainty conditional on the learned representation rather than full-network uncertainty. This restriction makes standard HMC tractable, but it may omit uncertainty arising from representation learning (Izmailov et al., 2021; Andrade & Sato, 2025; Thiagarajan et al., 2025).

Finally, the binary task contained only one HMC error, making uncertainty-error ranking unstable, and the reported risk-coverage curves are descriptive locked-test diagnostics rather than prospectively validated abstention rules. Similarly, Gaussian perturbation and random feature masking measure controlled sensitivity, not adversarial or external out-of-distribution robustness. Future validation should therefore combine independent temporal or unseen-family data with validation-selected uncertainty thresholds before operational forensic deployment.

## 5.5 Practical and Methodological Implications

For memory-forensic practice, MAL-BAYES-III is most useful as a decision-support extension rather than as a replacement for the strongest discriminative classifier. Its posterior outputs can accompany the predicted malware category with calibration and uncertainty information, allowing ambiguous multiclass cases to be prioritized for analyst review. This role is consistent with uncertainty-aware cybersecurity approaches that use uncertainty to support escalation or selective acceptance rather than treating confidence as an automatic decision rule (Yang et al., 2024; Yumlembam et al., 2025).

Methodologically, the study shows that standard HMC can be made auditable and computationally manageable by restricting posterior sampling to a compact neural decision head while preserving a frozen nonlinear representation. The resulting framework separates discrimination, calibration, convergence, uncertainty usefulness, and sensitivity instead of treating them as one performance claim. This provides a reproducible basis for future memory-forensic studies to test richer Bayesian representations and prospectively validated uncertainty thresholds without overstating accuracy or robustness.

## 6. Conclusion

MAL-BAYES-III developed a partial-Bayesian memory-forensic classifier in which a deterministic neural encoder was combined with an HMC-sampled final decision head. The framework preserved near-ceiling binary detection and competitive neural-level four-category discrimination while providing calibrated posterior probabilities, credible intervals, predictive entropy, mutual information, convergence diagnostics, and selective-prediction evidence. The multiclass task provided the strongest uncertainty result, with posterior uncertainty rising on difficult cases and supporting lower retained risk when high-entropy observations were deferred.

The study does not establish superior multiclass accuracy or universal robustness; its contribution is the integration of diagnostically validated HMC posterior inference with leakage-controlled malware classification and case-level uncertainty assessment. Future work should evaluate the framework on temporally separated or genuinely unseen malware data and derive operational uncertainty thresholds from validation data before forensic deployment.

## References

- Andrade, D., & Sato, K. (2025). On the effectiveness of partially deterministic Bayesian neural networks. *Computational Statistics*, 40, 2491-2518. <https://doi.org/10.1007/s00180-024-01561-7>
- Bouke, M. A., & Abdullah, A. (2023). An empirical study of pattern leakage impact during data preprocessing on machine learning-based intrusion detection models reliability. *Expert Systems with Applications*, 230, 120715. <https://doi.org/10.1016/j.eswa.2023.120715>
- Carrier, T., Victor, P., Tekeoglu, A., & Habibi Lashkari, A. (2022). Detecting obfuscated malware using memory feature engineering. In *Proceedings of the 8th International Conference on Information Systems Security and Privacy (ICISSP 2022)* (pp. 177-188). SCITEPRESS. <https://doi.org/10.5220/0010908200003120>
- Dehfouli, Y., & Habibi Lashkari, A. (2025). Memory analysis for malware detection: A comprehensive survey using the OSCAR methodology. *ACM Computing Surveys*, 58(4), Article 86, 1-58. <https://doi.org/10.1145/3764580>
- Dener, M., Ok, G., & Orman, A. (2022). Malware detection using memory analysis data in big data environment. *Applied Sciences*, 12(17), 8604. <https://doi.org/10.3390/app12178604>
- Doan, B. G., Yang, S., Montague, P., De Vel, O., Abraham, T., Camtepe, S., Kanhere, S. S., Abbasnejad, E., & Ranasinghe, D. C. (2023). Feature-space Bayesian adversarial learning improved malware detector robustness. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(12), 14783-14791. <https://doi.org/10.1609/aaai.v37i12.26727>
- Doan, B. G., Nguyen, D. Q., Montague, P., Abraham, T., De Vel, O., Camtepe, S., Kanhere, S. S., Abbasnejad, E., & Ranasinghe, D. C. (2024). Bayesian learned models can detect adversarial malware for free. In J. Garcia-Alfaro, R. Kozik, M. Choraś, & S. Katsikas (Eds.), *Computer security - ESORICS 2024 (Lecture Notes in Computer Science, Vol. 14982, pp. 45-65)*. Springer. [https://doi.org/10.1007/978-3-031-70879-4\\_3](https://doi.org/10.1007/978-3-031-70879-4_3)
- Feng, Y., Rudner, T. G. J., Tsilivis, N., & Kempe, J. (2024). Attacking Bayes: On the adversarial robustness of Bayesian neural networks. *Transactions on Machine Learning Research*, 2024.
- Hossain, M. A., & Islam, M. S. (2024). Enhanced detection of obfuscated malware in memory dumps: A machine learning approach for advanced cybersecurity. *Cybersecurity*, 7, Article 16. <https://doi.org/10.1186/s42400-024-00205-z>
- Izmailov, P., Vikram, S., Hoffman, M. D., & Wilson, A. G. G. (2021). What are Bayesian neural network posteriors really like? In *Proceedings of the 38th International Conference on Machine Learning (Vol. 139, pp. 4629-4640)*. *Proceedings of Machine Learning Research*.
- Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9), 100804. <https://doi.org/10.1016/j.patter.2023.100804>
- Madamidola, O. A., Ngobigha, F., & Ez-zizi, A. (2025). Detecting new obfuscated malware variants: A lightweight and interpretable machine learning approach. *Intelligent Systems with Applications*, 25, 200472. <https://doi.org/10.1016/j.iswa.2024.200472>
- Maniriho, P., Mahmood, A. N., & Chowdhury, M. J. M. (2024). MeMalDet: A memory analysis-based malware detection framework using deep autoencoders and stacked ensemble under temporal evaluations. *Computers & Security*, 142, 103864. <https://doi.org/10.1016/j.cose.2024.103864>
- Naeem, H., Dong, S., Falana, O. J., & Ullah, F. (2023). Development of a deep stacked ensemble with process based volatile memory forensics for platform independent malware detection and classification. *Expert Systems with Applications*, 223, 119952. <https://doi.org/10.1016/j.eswa.2023.119952>
- Osei-Wusu, F., Appiah, O., Mensah, P. K., Nimbe, P., & Donkoh, E. K. (2026a). A graph-based Bayesian intrusion detection framework with Gibbs sampling for uncertainty-aware IoT security. *Journal of Reliable Intelligent Environments*, 12, Article 9. <https://doi.org/10.1007/s40860-026-00272-8>
- Osei-Wusu, F., Appiah, O., Mensah, P. K., Nimbe, P., & Donkoh, E. K. (2026b). A graph-based Bayesian intrusion detection framework with No-U-Turn sampling for uncertainty-aware IoT security. *Concurrency and Computation: Practice and Experience*, 38(12), e70817. <https://doi.org/10.1002/cpe.70817>

- Roy, K. S., Ahmed, T., Udas, P. B., Karim, M. E., & Majumdar, S. (2023). MalHyStack: A hybrid stacked ensemble learning framework with feature engineering schemes for obfuscated malware analysis. *Intelligent Systems with Applications*, 20, 200283. <https://doi.org/10.1016/j.iswa.2023.200283>
- Shafin, S. S., Karmakar, G., & Mareels, I. (2023). Obfuscated memory malware detection in resource-constrained IoT devices for smart city applications. *Sensors*, 23(11), 5348. <https://doi.org/10.3390/s23115348>
- Thiagarajan, P., Zaki, T. A., & Shields, M. D. (2025). Accelerating Hamiltonian Monte Carlo for Bayesian inference in neural networks and neural operators. *Computer Methods in Applied Mechanics and Engineering*, 447, 118401. <https://doi.org/10.1016/j.cma.2025.118401>
- Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Bürkner, P.-C. (2021). Rank-normalization, folding, and localization: An improved R-hat for assessing convergence of MCMC. *Bayesian Analysis*, 16(2), 667-718. <https://doi.org/10.1214/20-BA1221>
- Yang, T., Qiao, Y., & Lee, B. (2024). Towards trustworthy cybersecurity operations using Bayesian deep learning to improve uncertainty quantification of anomaly detection. *Computers & Security*, 144, 103909. <https://doi.org/10.1016/j.cose.2024.103909>
- Yumlembam, R., Issac, B., & Jacob, S. M. (2025). Enhancing decision-making in Windows PE malware classification during dataset shifts with uncertainty estimation. *Knowledge-Based Systems*, 331, 114723. <https://doi.org/10.1016/j.knosys.2025.114723>
- Zhao, Y., Li, L., Wang, H., Cai, H., Bissyandé, T. F., Klein, J., & Grundy, J. (2021). On the impact of sample duplication in machine-learning-based Android malware detection. *ACM Transactions on Software Engineering and Methodology*, 30(3), Article 40. <https://doi.org/10.1145/3446905>