

Privacy-Preserving AI Verification via Minimal Information Disclosure

Sleem Abdelghafar
Rice University

Gabriel Kulp
Intelligence Security Laboratories

Abstract

AI verification crosses a trust boundary: a verifier must learn enough to establish an authorized claim, yet the same evidence can reveal sensitive details about the model, workload, or hardware. We introduce minimal information disclosure (MID), which designs and quantifies the information content of verifier-facing evidence itself. MID measures collateral leakage with conditional mutual information: what the release reveals about the protected property after the authorized result is known. MID is general by design: it can accommodate different verification goals, protected properties, evidence sources, and deployment constraints. To demonstrate MID’s practicality, we evaluate it on four physical measurements and six verification tasks spanning execution type, hardware identity, compute scale, and model identity. These experiments use three mechanism-design variables—the evidence channel, collection policy, and release transformation—but MID is not limited to these choices and can accommodate other deployable mechanisms. Across these tasks, MID produces three releases with perfect held-out verification and zero measured collateral leakage, while the remaining tasks yield explicit privacy–utility frontiers. MID also supports ZKP-certified releases: we demonstrate our proposed linear-projection mechanism using a Groth16 zk-SNARK.

1 Introduction

Suppose a verifier must determine whether a data center trained a model, stayed within a compute limit, used an approved accelerator, or passed an evaluation [5, 6, 2, 3, 9]. The power traces, telemetry records, or electromagnetic measurements that support those claims may also reveal private workload details, model architecture, or hardware configuration [48, 49, 27, 46, 31]. Minimal information disclosure (MID) addresses this tension by designing the information content of verification evidence itself: enough to establish the authorized claim while revealing as little as possible about a declared protected property beyond it. Authentication can establish where the evidence came from, and ZKPs can establish how a report was computed; MID determines what the report should reveal. In July 2026, a statement signed by more than 1,200 employees of frontier AI companies called for an international effort to develop the technical and governance tools needed to deliberately pace frontier AI development [1]. Credible evidence of compliance will be central to such coordination, making the design problem addressed by MID increasingly consequential.

Why MID? Sastry et al. suggest that a regulator might receive only a single compliance bit; Baker et al. describe a yes/no determination as the minimal output of a compliance test; and Harack et al. propose one-bit outputs for human assessments conducted in a controlled facility [6, 3, 9]. Other proposals and systems use narrowly specified, minimal, or fixed-alphabet reports [2, 4, 8, 10]. These works establish the importance of controlling what reaches the verifier. MID addresses an unresolved design question: among candidate reports, which one reveals the least beyond the authorized answer while preserving verification? Evidence size alone cannot answer this question.

Auditor-in-a-Box provides a concrete example: it presents a deterministic output filter intended to impose a one-bit worst-case disclosure bound, and its published Monitor Query Validation evaluation reports a VALID/INVALID verdict for each researcher specification [12, 11]. This restricted output limits

Correspondence to: Sleem Abdelghafar (msm15@rice.edu).

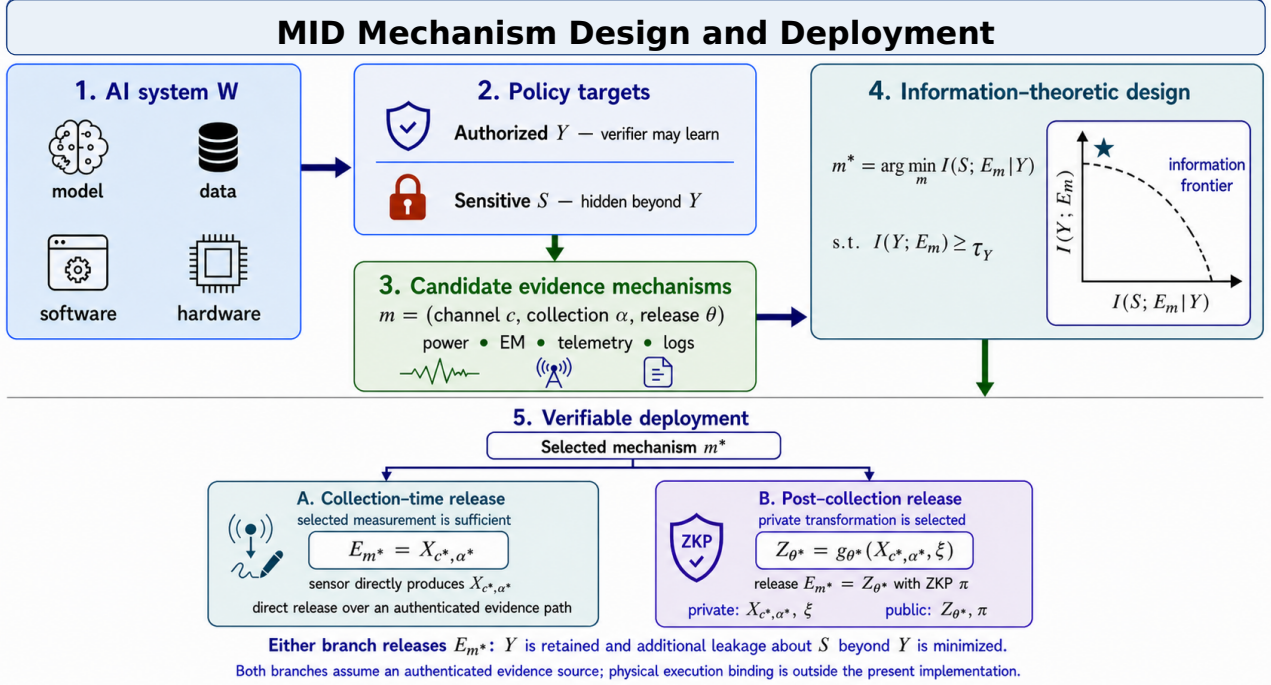


Figure 1: MID mechanism design and deployment. For an AI execution, a policy specifies an authorized target Y , a sensitive target S , and candidate evidence mechanisms $m = (c, \alpha, \theta)$. Information-theoretic optimization selects m^* to retain authorized information while minimizing additional information about S . Deployment then either releases the selected collection-time measurement directly or keeps the measurement private and releases the selected transformed value with a ZKP of correct computation.

how much can leave, but does not make the verdict privacy-preserving with respect to a protected property; on the authors’ evaluation distribution, the verdict measurably reveals request group. We focus on requests that human reviewers accepted. The monitor’s one-bit decision still distinguishes two sets in the authors’ evaluation: one contains many adjacent, proxy, or borderline requests, while the other consists largely of ordinary research requests. A simple predictor learns from the other test cases and then receives only the VALID/INVALID decision for a new request—not the request itself. It identifies which set the request came from with 72.4% balanced accuracy, compared with 50% without the decision. Appendix A gives the full experiment. Our RL experiment, Figure 5, shows the same problem for AI-execution evidence: several one-bit reports reveal exact-workload information, whereas MID selects one that achieves perfect held-out RL-versus-non-RL verification with zero measured collateral leakage. MID is not limited to one-bit reports: the same design question applies to richer evidence, motivating MID’s general formulation below.

For an AI execution W , the policy specifies an authorized target Y that the verifier may learn and a protected property S that should remain hidden beyond what Y necessarily reveals. A candidate mechanism $m = (c, \alpha, \theta)$ produces verifier-facing evidence E_m : c selects the evidence channel, α controls how it is collected, and θ optionally transforms it before release. MID evaluates each mechanism using two quantities. Mutual information $I(Y; E_m)$ measures how informative the evidence is about the authorized target—the information available for verification. Conditional mutual information $I(S; E_m | Y)$ measures what the evidence additionally reveals about the protected property after the authorized result is known—the collateral leakage. We formulate the Privacy-Preserving AI Verification Problem as

$$\min_{m \in \mathcal{M}} I(S; E_m | Y) \quad \text{subject to} \quad I(Y; E_m) \geq \tau_Y. \quad (1)$$

Equation (1) asks which deployable evidence mechanism reveals the least about S among those sufficiently informative to verify Y . The threshold τ_Y sets the required verification utility, and conditioning on Y prevents MID from penalizing information inherent in the authorized answer. The information content of E_m is therefore the design object; c , α , and θ are controls through which MID shapes it. Our experiments use these three controls, but MID is not limited to these controls and can accommodate other deployable mechanisms simply by extending $\mathcal{M}$. By Fano’s inequality, the resulting evidence limits the success of any adversary attempting to infer S from (E_m, Y) .

The mechanism variables map directly onto the two deployment paths in Figure 1. When channel and collection choices are sufficient, a collection-time mechanism allows the evidence source itself to produce a low-information measurement that meets the verification requirement; no richer digital trace or private post-processing step is needed. When a post-collection transformation achieves lower leakage at the required utility, the richer measurement remains within a protected boundary and only the selected transformation is released, accompanied by a ZKP of correct computation. Both paths require an authenticated evidence source. Together, they allow MID to accommodate different deployment constraints rather than assume a single architecture. We instantiate these paths using physical evidence channels because they can observe execution independently of the system being checked and because prior side-channel studies provide concrete tests of what such evidence can reveal [48, 49, 46, 31].

To the best of our knowledge, MID is the first AI-verification framework to design the information content of verifier-facing evidence itself by selecting among feasible mechanisms to minimize disclosure about a declared protected property beyond the authorized result while satisfying a verification-utility requirement. Section 6 distinguishes this contribution from systems that authenticate measurements, protect private inputs, prove prescribed computations, or constrain verifier-facing reports [2, 4, 9, 8, 41]; Section 5 explains how MID can be integrated with these methods.

Our specific contributions are as follows:

- We formulate privacy-preserving AI verification as an information-theoretic optimization problem. MID selects among deployable evidence mechanisms to preserve the information needed for authorized verification while minimizing collateral leakage about a protected property. This makes the information content of verifier-facing evidence itself the design object.
- We develop collection-time and post-collection mechanisms that let MID shape evidence through channel selection, collection policy, and release transformation. We show how a selected mechanism can be deployed as an authenticated direct release or as a release computed from private evidence and certified by a ZKP, and demonstrate our linear-projection mechanism with a Groth16 zk-SNARK [23].
- We demonstrate MID across six physical-measurement tasks spanning execution type, hardware identity, compute scale, and model identity. Three selected mechanisms achieve perfect held-out verification with zero measured collateral leakage, while the remaining tasks yield explicit privacy-utility frontiers. We also evaluate the selected releases against the ModelSpy and DeepTheft attack methods [48, 49].
- We show, using Auditor-in-a-Box’s released evaluation, that even a one-bit verdict can disclose information beyond its intended answer. A predictor that receives only the verdict identifies which of the authors’ two request sets a private request came from with 72.4% balanced accuracy, showing why limiting output size alone is not sufficient.

2 MID Mechanism Design

MID turns the policy’s disclosure boundary into an information-theoretic mechanism-selection problem. We first define the authorized and protected properties, candidate evidence mechanisms, and trust assumptions, then formalize the optimization that selects what the verifier should receive.

2.1 Problem Setup

Privacy-preserving AI verification begins with a policy decision: what may the verifier learn, what must remain protected beyond that answer, and what verification utility is required? Let W denote an execution episode, including any relevant model, data, software, and hardware state. The policy or audit rule defines

$$Y = f_{\text{ver}}(W), \quad S = f_{\text{sens}}(W).$$

Y is the authorized target. S is a property that should not be revealed beyond what follows from Y , such as model identity, architecture, training data, prompt family, workload purpose, or fine-grained resource use. A policy may specify several sensitive targets $S_1, \dots, S_J$.

Four roles are conceptually distinct. A policy setter specifies Y , the protected properties S , and the required verification utility—how informative the evidence must be. An operator executes W . An evidence source observes some consequence of that execution. A verifier receives the released evidence and decides whether to accept the authorized claim. One organization may occupy several roles, but separating them distinguishes two security goals: the underlying verification system must establish that the evidence is trustworthy, while MID limits what the verifier can learn from that evidence.

Evidence mechanisms. Once the policy fixes Y and S , MID designs the evidence passed from the execution to the verifier. During development, a rich baseline observation $R \sim P_R(\cdot | W)$ is available to compare and calibrate candidate mechanisms. It can be a high-rate trace, a multichannel telemetry record, a log, a stored model or related record, or a collection of these. The deployed system need not create or retain R . A candidate mechanism is

$$m = (c, \alpha, \theta) \in \mathcal{M},$$

where c selects an evidence channel or channel family, α controls collection, and θ controls an optional post-collection release transformation. At deployment, the selected sensor and collection rule directly produce

$$X_{c,\alpha} \sim P_{c,\alpha}(\cdot | W),$$

without first creating R . An optional release transformation produces

$$Z_\theta = g_\theta(X_{c,\alpha}, \xi),$$

where ξ is the release mechanism’s randomness. The principal release is

$$E_m^{\text{rel}} = \begin{cases} X_{c,\alpha}, & \text{collection-time release,} \\ Z_\theta, & \text{post-collection release.} \end{cases} \quad (2)$$

The complete verifier-visible evidence E_m consists of this release and any other fields the protocol sends to the verifier. Our offline experiments derive $X_{c,\alpha}$ from R . In deployment, a collection-time mechanism implements (c, α) at the evidence source, so the richer digital record R need not be created, transmitted, or stored. If a post-collection transformation is selected, $X_{c,\alpha}$ remains inside a sensor, secure meter, or TEE and only Z_θ is disclosed.

Selecting memory capacity rather than dynamic power, lowering an electromagnetic (EM) sampling rate, and releasing one noisy number are therefore all mechanism choices. MID chooses among them.

Hard decisions enter MID as candidate evidence mechanisms: their source must be bound to the execution, and selection evaluates whether their errors depend on S at fixed Y .

Threat model. The operator is the integrity adversary: it may attempt to fabricate, omit, replay, or misattribute evidence. The underlying verification system must therefore authenticate the evidence source and bind its output to the relevant execution; MID assumes this integrity layer operates according to its stated security assumptions. It may be implemented by an auditor-controlled sensor, signed secure meter, trusted accelerator firmware, hardware attestation, monitored network boundary, or redundant observations [8, 4]. For post-collection mechanisms, a ZKP can bind the released report to the authenticated private measurement.

The verifier is the privacy adversary. It is authorized to learn Y but is not trusted with S . It receives E_m , knows the selected mechanism, and may use any inference method to recover S . MID limits this risk by minimizing $I(S; E_m | Y)$, where E_m includes the complete verifier-visible interface, including metadata, commitments, signatures, and ZKP transcripts. MID does not make a compromised evidence source honest or establish that its measurement is truthful; it determines what authenticated evidence should reveal.

MID is calibrated using grouped, labeled executions from the intended environment; the policy declares the protected targets before selection. The mechanism designers may access R during development, while the deployed interface releases only the selected evidence and required verification fields.

2.2 Information-Theoretic Optimization

Equation (1) is our information-theoretic formulation of the Privacy-Preserving AI Verification Problem: among deployable evidence mechanisms, select the one that minimizes disclosure about S beyond Y while meeting the required verification utility. The same problem can be written generally by letting $U_Y(E_m)$ denote verification utility and $L_S(E_m | Y)$ denote leakage beyond the authorized target:

$$m^* = \arg \min_{m \in \mathcal{M}} L_S(E_m | Y) \quad \text{s.t.} \quad U_Y(E_m) \geq \tau_Y. \quad (3)$$

For several sensitive targets, we minimize the largest leakage:

$$m^* = \arg \min_{m \in \mathcal{M}} \max_{1 \leq j \leq J} L_{S_j}(E_m | Y) \quad \text{s.t.} \quad U_Y(E_m) \geq \tau_Y. \quad (4)$$

This form controls each declared target separately; if their combination is also sensitive, the vector $(S_1, \dots, S_J)$ is included as another target. We use mutual information for authorized utility and conditional mutual information for sensitive leakage [34]:

$$\begin{aligned} U_Y(E_m) &\triangleq I(Y; E_m), \\ L_S(E_m | Y) &\triangleq I(S; E_m | Y). \end{aligned} \quad (5)$$

The first measures evidence about the authorized target; the second measures additional evidence about S after Y is known. With these definitions, Eq. (4) minimizes $\max_j I(S_j; E_m | Y)$. We use base-2 logarithms. Zero leakage is equivalent to $S \perp\!\!\!\perp E_m | Y$: the release may reveal Y while conveying nothing further about S . If S and Y are correlated, learning the authorized result can still reveal something about S ; MID treats that as authorized and penalizes only the additional disclosure carried by the evidence.

For discrete Y , the best possible point is

$$I(Y; E_m) = H(Y), \quad I(S; E_m | Y) = 0. \quad (6)$$

It is possible only if some allowed release determines Y and satisfies $S \perp\!\!\!\perp E_m | Y$. At that point, $P(S | E_m, Y) = P(S | Y)$ almost surely. Otherwise, the allowed mechanisms define the best achievable privacy–utility tradeoffs, which is fully characterized by the information frontier (e.g., Figure 2).

The objective does not depend on a particular attack. Let any later procedure, possibly randomized and computationally unbounded, output $A = a(E_m, Y, U)$, where U is independent randomness. Because A is computed only from E_m , Y , and U , these variables satisfy the conditional Markov relation $S \rightarrow E_m \rightarrow A$ given Y , so

$$I(S; A | Y) \leq I(S; E_m | Y). \quad (7)$$

This is the conditional data-processing inequality. Thus the population leakage objective upper-bounds the information available to any later rule for inferring the specified S . This includes, for example, membership, attribute, inversion, and architecture-inference attacks [51, 52, 53, 55, 48, 49].

Fano’s inequality then shows how the resulting evidence limits the success of any adversary attempting to infer S from (E_m, Y) . For a binary S that is balanced within each authorized class, it gives

$$P_e \geq h_2^{-1}(1 - I(S; E_m | Y)),$$

where P_e is the minimum error achievable by such an adversary and h_2^{-1} is the lower branch of the inverse binary-entropy function.

Only mechanisms that can be deployed are included:

$$\mathcal{M} \subseteq \{(c, \alpha, \theta) : c \in \mathcal{C}, \alpha \in \mathcal{A}_c, \theta \in \Theta_{c,\alpha}\}. \quad (8)$$

The sets $\mathcal{C}$, $\mathcal{A}_c$, and $\Theta_{c,\alpha}$ encode limits on cost and delay and the required security guarantees. MID can therefore select the sensor, change how data are collected, or compute a narrow release inside a protected device; it is not restricted to adding noise after a sensor has been fixed.

Finite-sample estimation. The population objective is defined with respect to the data-generating distribution, but in practice mechanisms must be compared using a finite development set. We therefore use either mutual-information estimators or finite-sample comparison scores appropriate to the candidate family. Mutual information measures how much each class-specific evidence distribution differs from the overall evidence distribution:

$$I(C; E_m) = \sum_c p(c) D_{\text{KL}}(P(E_m | C = c) \| P(E_m)). \quad (9)$$

Direct estimation is unreliable for high-dimensional traces and small datasets. Different types of evidence therefore require different estimators or finite-sample surrogates. For continuous traces, we use a regularized Gaussian separation score.

Suppose $E_m | C = c \approx \mathcal{N}(\mu_c, \Sigma_C)$ with shared covariance. We estimate class separation with the regularized Mahalanobis score

$$D_{a,b}(E_m) = (\mu_a - \mu_b)^\top (\Sigma_C + \gamma I)^{-1} (\mu_a - \mu_b). \quad (10)$$

Here $\gamma \geq 0$ stabilizes covariance inversion. When $\gamma = 0$, the quadratic form equals the sum of the two directed Kullback–Leibler (KL) divergences between shared-covariance Gaussians. For $\gamma > 0$, it is a regularized separation proxy: regularization stabilizes the estimate without changing the released evidence. For binary Y , we write

$$D_Y(E_m) = (\mu_1 - \mu_0)^\top (\Sigma_Y + \gamma I)^{-1} (\mu_1 - \mu_0), \quad (11)$$

and use the weighted pairwise average for multiclass Y . Let $\mu_{s,y}$ denote the evidence mean for sensitive class s at fixed authorized class y , and let Σ_y be the corresponding pooled within-sensitive-class covariance. Conditional sensitive separability is computed within each authorized class:

$$D_S^{\text{cond}}(E_m) = \sum_y p(y) \sum_{s < s'} p(s | y) p(s' | y) (\mu_{s,y} - \mu_{s',y})^\top (\Sigma_y + \gamma I)^{-1} (\mu_{s,y} - \mu_{s',y}). \quad (12)$$

Under the balanced binary shared-covariance model, in the population limit with $\gamma = 0$, the Bayes accuracy associated with authorized separation D_Y is

$$a_Y = \Phi\left(\frac{\sqrt{D_Y}}{2}\right), \quad D_Y \geq 4[\Phi^{-1}(a_Y)]^2. \quad (13)$$

where Φ is the standard-normal cumulative distribution function. Thus a target accuracy can be translated into a separation threshold under this model; for example, $a_Y = 0.80$ gives $D_Y \geq 2.833$. This model-specific translation guides candidate comparison, and we also report held-out balanced accuracy (BA), the mean recall across classes. Because the score measures mean separation under shared covariance, we pair it with held-out prediction and attack evaluations that probe additional structure.

For reports with a small set of possible values, we compute plug-in mutual information from held-out frequencies and describe a zero value as zero measured leakage. Unless noted otherwise, authorized classes and sensitive classes within each authorized class receive equal prior weight. For rich evidence, information values derived from held-out prediction models are lower bounds on the information in the underlying trace.

Nested mechanism selection. Estimating every candidate on the same data used to select it would make both utility and leakage look better than they are. Algorithm 1 therefore splits the data twice. Inner splits fit and compare candidates; an outer group, unseen during selection, evaluates the chosen mechanism. Grouping is by physical run or matched configuration, so windows from one run and paired workloads never appear on both sides of a split.

Algorithm 1 MID mechanism selection

- 1: **Input:** episodes $\mathcal{D} = \{(R_i, Y_i, S_i)\}$, specified candidates $\mathcal{M}$, threshold τ_Y , group labels
 - 2: **for** each outer held-out group G **do**
 - 3: **for** each $m = (c, \alpha, \theta) \in \mathcal{M}$ **do**
 - 4: Fit m on $\mathcal{D} \setminus G$ using inner folds and generate inner out-of-fold releases
 - 5: Estimate $\hat{U}_Y(m)$ and $\hat{L}_S(m | Y)$ on those inner releases
 - 6: **end for**
 - 7: **if** no candidate satisfies $\hat{U}_Y(m) \geq \tau_Y$ **then**
 - 8: **return** infeasible under the declared $\mathcal{M}$ and τ_Y
 - 9: **else**
 - 10: $\hat{m}_G \leftarrow \arg \min_m \hat{L}_S(m | Y)$ subject to $\hat{U}_Y(m) \geq \tau_Y$
 - 11: Fit $\hat{m}_G$ on $\mathcal{D} \setminus G$ and release evidence for G
 - 12: **end if**
 - 13: **end for**
 - 14: Combine the outer held-out results to evaluate the complete selection procedure
 - 15: Re-run the same selection rule on all calibration groups and fit the final mechanism m^*
 - 16: Compile the interface for m^* as either an authenticated direct release or an authenticated private input with a ZKP relation for its transformation
 - 17: **return** m^* , its deployment specification, and outer-held-out measurements of the selection procedure
-

The deployed monitor fixes m^* before observing a new execution and discloses only E_{m^*} . The outer groups evaluate the complete process, including the choice among candidates; the final fit applies the same rule, declared in advance, to all development groups. Mutual information supplies the common objective; Gaussian scores and prediction models support finite-sample selection, while attack evaluations provide separate empirical checks.

3 Concrete MID Mechanisms And Deployment

The information-theoretic objective determines what a selected evidence interface must achieve; this section turns that choice into concrete mechanisms and deployable releases. We first describe mechanism families that act during or after collection, then show how authenticated sources and ZKPs bind the selected release to the execution.

3.1 Concrete Mechanisms

We instantiate $\mathcal{M}$ with reusable mechanism families that act either during collection or after it. Section 4 specifies the channel c , collection policy α , and release rule θ selected in each task.

Collection-time mechanisms. Channel restriction selects which physical signal, telemetry field, counter, or static hardware attribute the evidence source acquires. Acquisition minimization controls the sampling rate, observation window, sensor placement, or summary computed inside the sensor. These choices directly construct $X_{c,\alpha}$; when the result meets the utility threshold, the disclosed evidence is $E_m = X_{c,\alpha}$ and no richer deployment trace is created.

Hard-decision and quantized reports. When $X_{c,\alpha}$ remains more informative than the authorized claim requires, a post-collection rule can disclose a coarser result. Given an authorized score $h(X_{c,\alpha})$, a hard-decision mechanism releases

$$Z_{\text{hard}} = \mathbb{I}\{h(X_{c,\alpha}) \geq t\}. \quad (14)$$

Here t is the hard-decision threshold. A quantized randomized mechanism instead releases

$$Z_q = q \text{round}\left(\frac{h(X_{c,\alpha}) + \eta}{q}\right), \quad q > 0, \quad \eta \sim \mathcal{N}(0, \sigma^2), \quad (15)$$

which includes deterministic quantization when $\sigma = 0$. These transformations can be combined with a selected collection-time mechanism and computed inside a protected device after collection.

Privacy-aware linear projections. Hard decisions and fixed quantizers may not yield a feasible point under Eq. (3). A more flexible post-collection family is the k -dimensional noisy linear release

$$Z = A_\beta X_{c,\alpha} + \sigma \xi, \quad \xi \sim \mathcal{N}(0, I_k). \quad (16)$$

For binary Y , let $\mu_y = \mathbb{E}[X_{c,\alpha} \mid Y = y]$, let $\Delta_Y = \mu_1 - \mu_0$, and let Σ_Y be the pooled within- Y covariance. With $\mu_{s,y} = \mathbb{E}[X_{c,\alpha} \mid S = s, Y = y]$, define conditional sensitive scatter

$$B_{S|Y} = \sum_y p(y) \sum_s p(s \mid y) (\mu_{s,y} - \mu_y)(\mu_{s,y} - \mu_y)^\top. \quad (17)$$

For a covariance-regularization parameter $\lambda > 0$, the rows of A_β are the leading generalized eigenvectors v satisfying

$$\Delta_Y \Delta_Y^\top v = \lambda_v (\Sigma_Y + \beta B_{S|Y} + \lambda I) v. \quad (18)$$

Here λ_v is the generalized eigenvalue associated with v . For $k = 1$, the release direction is proportional to

$$w_\beta = (\Sigma_Y + \beta B_{S|Y} + \lambda I)^{-1} \Delta_Y. \quad (19)$$

β penalizes directions that distinguish S within fixed Y ; σ adds release noise.

3.2 Verifiable Deployment With ZKPs

The mechanisms above determine what should cross the trust boundary; deployment must also establish that the evidence came from the claimed execution and that any post-collection transformation was applied correctly. Both paths require an authenticated evidence source, while the post-collection path additionally uses a ZKP to certify the selected transformation without revealing the private witness beyond what follows from the public instance. This computation-integrity guarantee holds for a computationally bounded verifier under the ZKP system’s cryptographic and setup assumptions. Groth16 [23] additionally requires a circuit-specific structured reference string; a deployment must generate it through an appropriate trusted or multi-party setup and bind the proving and verification keys to the audited circuit. Let $m^* = (c^*, \alpha^*, \theta^*)$. The evidence source commits to X_{c^*, α^*} and links that commitment to an execution identifier id_W and the selected collection configuration. The values given to the verifier are called the public instance:

$$x_{\text{pub}} = (\text{id}_W, C_X, Z_{\theta^*}, m^*, \sigma_{\text{src}}),$$

where σ_{src} is the source-authentication signature or tag. Here *public* means visible to the verifier outside the ZKP; it does not require unrestricted publication. The values kept hidden inside the ZKP, called the private witness, are $w = (X_{c^*, \alpha^*}, \xi, r)$, where r opens the commitment. A deployment accepts only if

$$\begin{aligned} \exists X_{c^*, \alpha^*}, \xi, r : \quad & C_X = \text{Com}(X_{c^*, \alpha^*}; r) \\ & \wedge \text{VerifySrc}(\sigma_{\text{src}}; \text{id}_W, C_X, c^*, \alpha^*) = 1 \\ & \wedge Z_{\theta^*} = g_{\theta^*}(X_{c^*, \alpha^*}, \xi). \end{aligned} \tag{20}$$

The source-authentication check may be evaluated inside the circuit or separately. In either case, acceptance also requires

$$\text{ZK.Verify}(vk, x_{\text{pub}}, \pi) = 1. \tag{21}$$

Here vk is the verification key and π is the ZKP. The monitor sends $(\text{id}_W, C_X, Z_{\theta^*}, m^*, \sigma_{\text{src}}, \pi)$ to the verifier. Computational zero knowledge protects X_{c^*, α^*} , ξ , and r beyond what follows from the public instance. The source signature establishes measurement origin and integrity, while the commitment and ZKP bind the released value to the private measurement and selected computation. All verifier-visible fields are included in E_m for disclosure accounting.

For a linear scalar release, a fixed-point circuit checks

$$\tilde{Z} = \sum_{j=1}^d \tilde{w}_j \tilde{X}_j + \tilde{\sigma} \tilde{\xi}. \tag{22}$$

Tildes denote fixed-point integer encodings, and d is the measurement-vector dimension. The mechanism parameters and $\tilde{Z}$ are public; $\tilde{X}$ and $\tilde{\xi}$ are private. A deployment circuit also enforces signed fixed-point encodings, scaling, rounding, and range bounds so that field equality matches the intended integer arithmetic. Randomized releases additionally require verifiable sampling; for example, the prover can commit to a hidden seed before a verifier challenge and derive finite-precision noise inside the circuit. The public commitment and challenge are included in E_m , while the seed and noise remain private so that the verifier cannot subtract the noise from the release. Section 4.4 demonstrates the arithmetic in Eq. (22) in Circom [24] and uses snarkjs [25] to generate and verify a Groth16 zk-SNARK [23] over the BN254 pairing-friendly curve, which snarkjs calls **bn128**.

4 Empirical Evaluation

Having defined MID’s objective, mechanisms, and deployment paths, we now ask whether it works in practice: Can MID select deployable evidence that preserves the information needed to verify a policy-relevant claim while suppressing collateral leakage?

Table 1: Verification questions evaluated in this paper.

Question	Why it matters	Authorized target Y	Measurement signal
Execution Type			
Is the allowed kind of computation running?	Inference-only proposals require evidence that distinguishes training from inference; policy and safety work also distinguishes consequential post-training methods [7, 27, 56].	Training vs. inference (Sec. 4.2); reinforcement learning (RL) vs. non-RL training (Sec. 4.2)	Facility electrical power; central processing unit (CPU) and graphics processing unit (GPU) utilization telemetry
Hardware Identity			
Is the approved accelerator being used?	Chip controls, operating licenses, and end-use verification require trustworthy claims about advanced-computing hardware [14, 2, 15, 60].	H100 vs. B200 (Sec. 4.3)	Accelerator memory, power, utilization, and temperature telemetry
Compute Scale			
Is resource use within the allowed class?	Compute-governance and international-verification proposals require evidence about the amount and organization of accelerator use [6, 7, 9, 13, 57].	Low vs. high execution tier (Sec. 4.4); number of compute nodes / number of GPUs used (Sec. 4.4)	Processor-package and memory power; facility electrical power
Model Identity			
Is the approved model type running?	Inference-verification and property-attestation systems require evidence that a claimed model or authorized variant actually executed [45, 18, 17, 26].	Convolutional neural network (CNN) vs. Transformer (Sec. 4.5)	GPU electromagnetic trace

4.1 Experimental Setup

To test MID across different policy questions and evidence sources, we evaluate six tasks on four public physical-measurement datasets, organized by the question answered by Y . Table 1 gives the policy motivation and measurement channel. Each case defines S , compares deployable mechanisms, and reports either a utility-constrained selection or the measured frontier from which a policy would select. The experiments treat the dataset records as authentic and focus on disclosure; Section 3.2 specifies deployment checks for origin, integrity, coverage, freshness, and execution binding. Throughout the evaluation, BA denotes balanced accuracy, the mean recall across classes. A score is *out-of-fold* when it is produced by a model that was not trained on the physical run being scored.

4.2 Execution Type

We begin with execution-type verification, which asks what computation is occurring without disclosing the finer workload. We study training versus inference and, within training, reinforcement learning versus non-RL procedures.

Training versus inference. Training detection is a direct prerequisite for policies that distinguish restricted training from permitted inference. Rahman and Tajdari evaluate this distinction using GPU telemetry under explicit evasion strategies [27]. We study the same authorized question through a different physical channel. Our processed corpus, constructed from the National Laboratory of the Rockies (NLR) dataset, contains Llama-2 70B low-rank adaptation (LoRA) and Stable Diffusion training runs together with Llama-3.1 70B inference runs from the InstructCoder and MLPerf prompt collections [61]. We authorize

$$Y \in \{\text{training, inference}\},$$

Table 2: Training-versus-inference results. MID retains 0.9966 authorized balanced accuracy while reducing mean sensitive-workload balanced accuracy from 0.9182 to 0.5010 and measured conditional leakage to 0.000239 bits.

Evidence	Y BA	$S \mid Y=\text{train}$ BA	$S \mid Y=\text{infer}$ BA	Mean $S \mid Y$ BA	$I(Y; E)$	$I(S; E \mid Y)$
Raw trace features	1.0000	0.9750	0.8613	0.9182	1.0000	0.6375
Scale-invariant MID report	0.9966	0.5000	0.5020	0.5010	0.9723	0.000239

and treat the finer workload identity as sensitive: Llama-2 versus Stable Diffusion within training and InstructCoder versus MLPerf within inference.

The raw baseline contains 27 facility-power summaries. Candidate reports vary the feature family, quantization width, and Gaussian noise. The selected feature family contains six dimensionless shape statistics—coefficient of variation, peak-to-mean ratio, autocorrelation at lags 1, 5, and 10, and spectral entropy—and excludes absolute power, duration, energy, sample count, and sampling interval. The report uses Eq. (15) with the out-of-fold binary classifier $h(X) = \hat{Y}_X$, $q = 0.05$, and $\sigma = 0.15$. We average the reported metrics over the mechanism’s randomness rather than report one noise draw.

In grouped out-of-fold evaluation, the raw features distinguish training from inference perfectly but also support sensitive-workload inference with mean balanced accuracy 0.9182. The selected MID report retains Y balanced accuracy 0.9966 and $I(Y; Z) = 0.9723$ bits, while reducing sensitive inference to 0.5010 and measured conditional leakage to 0.000239 bits. Table 2 therefore shows that the workload distinctions contained in raw facility-power evidence are not necessary for training-versus-inference verification. Figure 2 shows how MID places the selected report close to the ideal of high authorized information and zero collateral leakage. Figure 3 shows that the authorized classes remain separated after MID, while Figure 4 shows that the protected workload distributions become nearly indistinguishable within both training and inference, hence minimal leakage.

Reinforcement learning versus non-RL training. Reinforcement-learning post-training can materially change model behavior and capabilities, so whether RL was used is itself worth verifying [59, 58]. We ask whether telemetry can support that claim without disclosing the exact non-RL workload.

After removing byte-identical copies, the single-machine Elsayed dataset contains physical sessions spanning RL, forecasting, image captioning, image classification, and text generation [62]. We authorize $Y \in \{\text{RL}, \text{non-RL}\}$ and, at fixed $Y = \text{non-RL}$, protect the four-way exact workload. After excluding startup and shutdown edges, each session is divided into non-overlapping 300-sample windows. All windows from a session remain in the same one of four held-out folds. Models are fitted on windows, but held-out probabilities are averaged within each session before computing any metric or information value.

The rich baseline contains 144 summaries of GPU and CPU power, utilization, memory, temperature, and voltage. Candidate channels include individual and combined power or utilization, GPU and system memory, temperature, voltage, and scale-independent temporal shape. The selected channel contains three fields: GPU core load, activity through NVIDIA’s CUDA computing interface, and mean Direct3D (D3D) utilization. If $\hat{p}_i$ is the held-out session-level RL probability, the release is

$$Z_i = \mathbb{I}\{\hat{p}_i \geq 1/2\}, \quad 0 \equiv \text{non-RL}, \quad 1 \equiv \text{RL}. \quad (23)$$

A classifier using all 144 telemetry summaries recovers both Y and the four-way non-RL workload perfectly, giving one bit of authorized information and two bits of sensitive information under equal priors. The one-bit release is correct for every held-out session and constant across non-RL workloads: $I(Y; Z) = 1$ bit, sensitive balanced accuracy is 0.25, and the held-out plug-in estimate is $I(S; Z \mid Y = \text{non-RL}) = 0$.

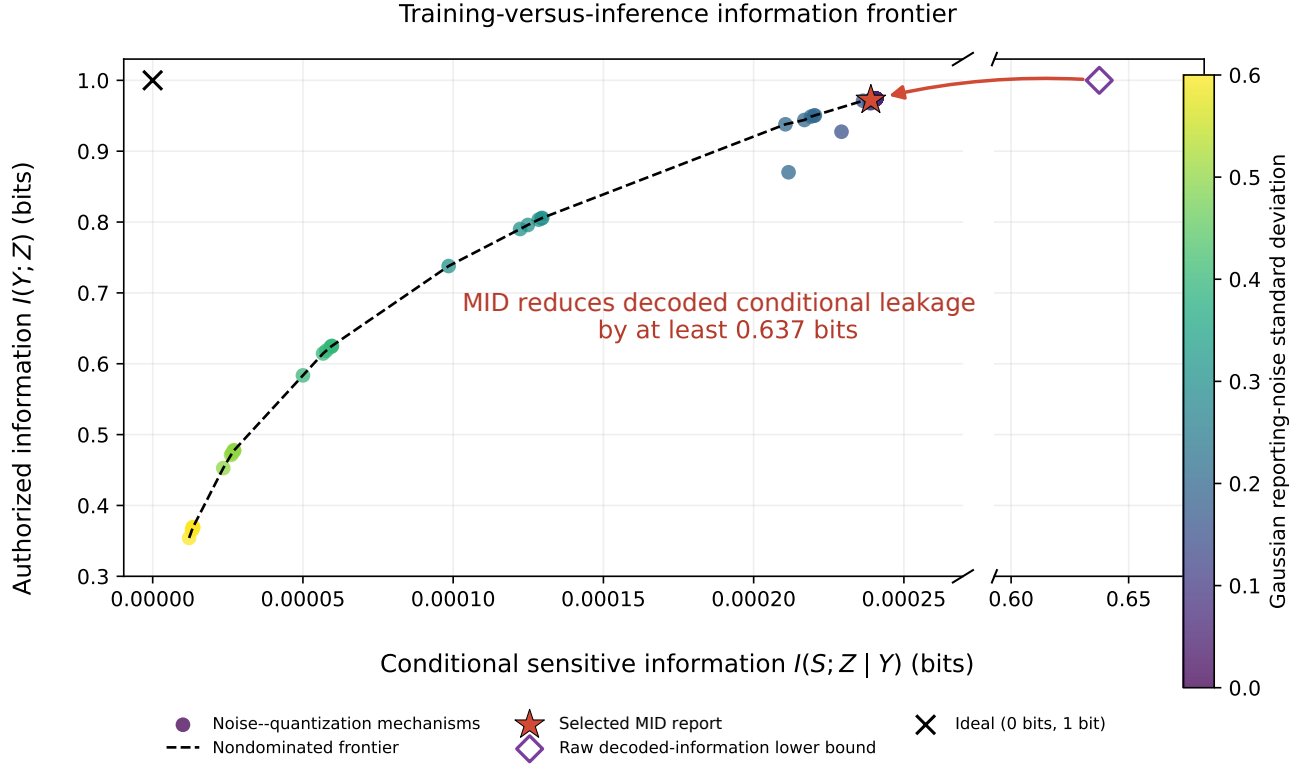


Figure 2: Training-versus-inference information frontier. The selected report retains 0.9723 bits of authorized information while reducing measured conditional leakage from a raw-evidence lower bound of 0.6375 bits to 0.000239 bits. Its position near the upper-left ideal shows that MID removes workload information without making the evidence uninformative for verification.

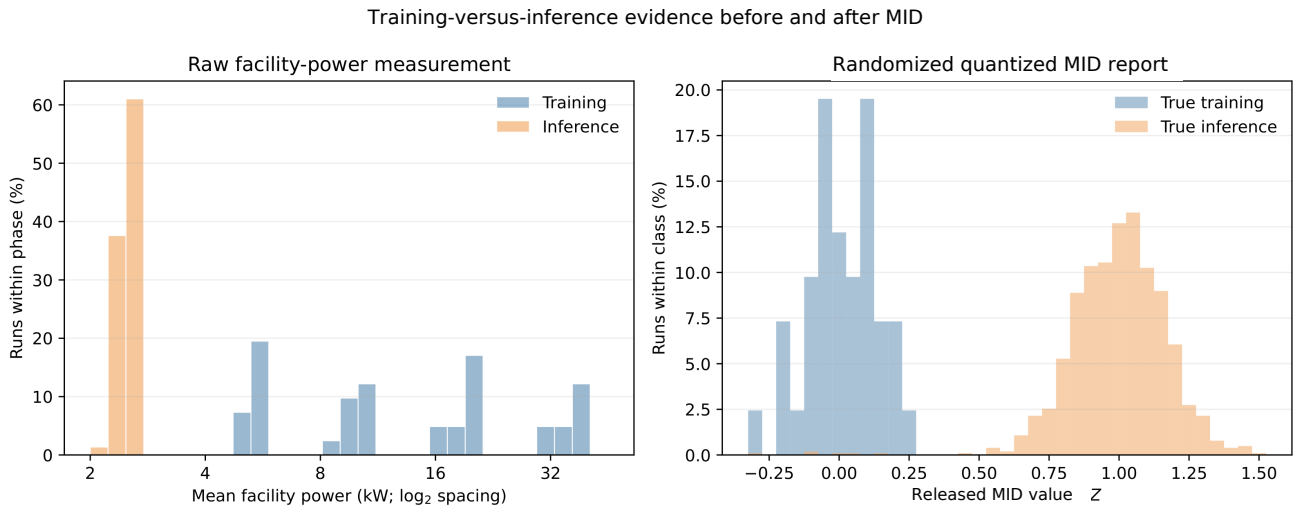


Figure 3: Training-versus-inference evidence before and after MID. The selected randomized report retains separated, multi-bin distributions for the authorized classes.

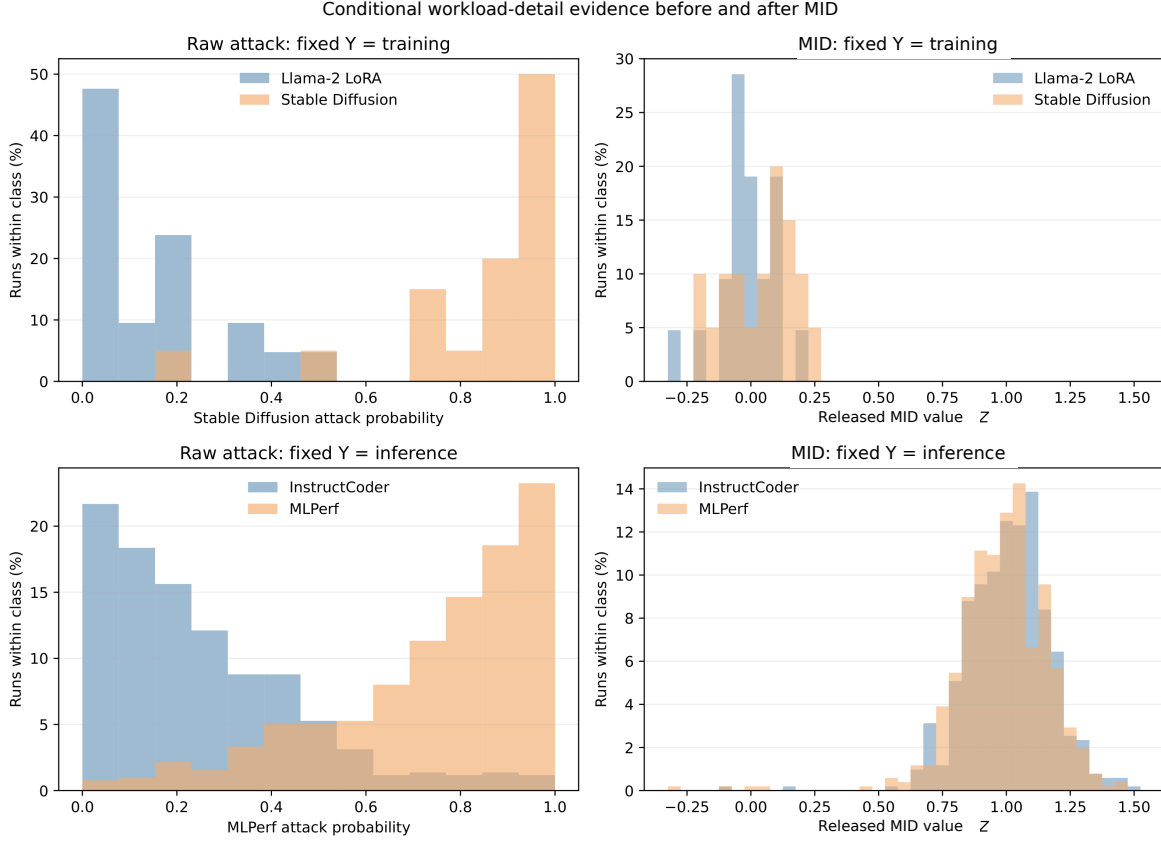


Figure 4: Conditional workload evidence. The top row fixes training and compares Llama-2 LoRA with Stable Diffusion; the bottom row fixes inference and compares InstructCoder with MLPerf. Raw trace features distinguish each sensitive pair, whereas the selected report makes the corresponding distributions nearly identical.

Table 3: RL-versus-non-RL results. The sensitive target is the four-way exact workload, evaluated only for non-RL sessions.

Evidence	Y BA	$I(Y; E)$	S BA	$I(S; E \mid Y=\text{non-RL})$
Raw telemetry	1.0000	1.0000	1.0000	2.0000
Selected GPU-utilization report	1.0000	1.0000	0.2500	0

Each colored candidate in Figure 5 is a one-bit hard report constructed from a different telemetry channel; the diamond is the rich-telemetry decoder lower bound. Several one-bit candidates still reveal exact-workload information because their errors vary across workloads at fixed non-RL. A small output alphabet therefore does not by itself control disclosure. The frontier compares measurement channels, not only sample rates or noise levels within one predetermined channel.

Figures 6 and 7 explain why the MID selected report reaches the empirical ideal. It is perfectly correct for RL versus non-RL (Figure 6) and returns the *same* non-RL answer for forecasting, image captioning, image classification, and text generation (Figure 7). Other candidate one-bit reports make workload-dependent errors and therefore retain exact-workload information. The result is not that one-bit reports are inherently private; it is that MID identifies which one-bit report preserves the authorized answer without encoding the protected workload through its behavior.

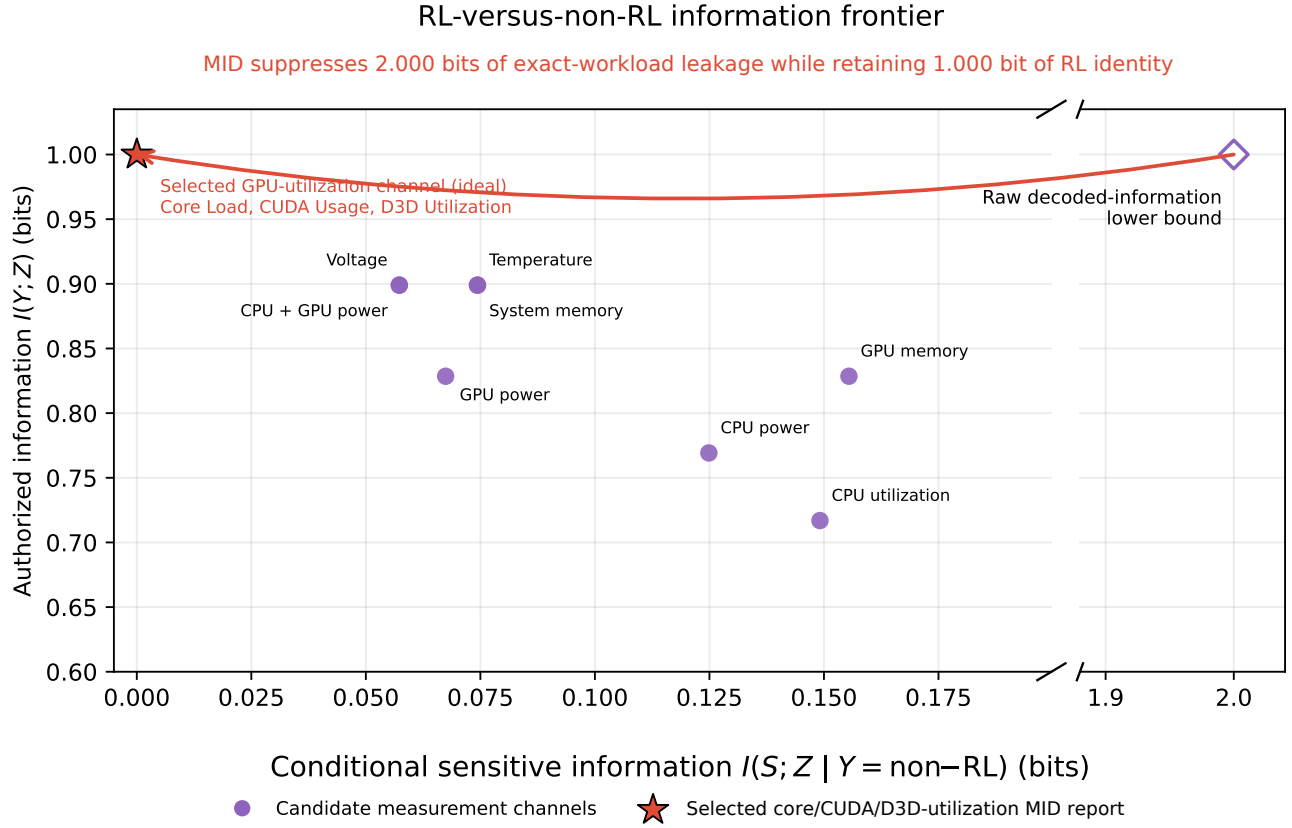


Figure 5: RL-versus-non-RL information frontier across telemetry-channel mechanisms. Every colored candidate is a one-bit RL-status report, yet several have nonzero exact-workload leakage. On the held-out sessions, the selected GPU-utilization report retains one bit of authorized information with zero measured exact-workload information at fixed non-RL.

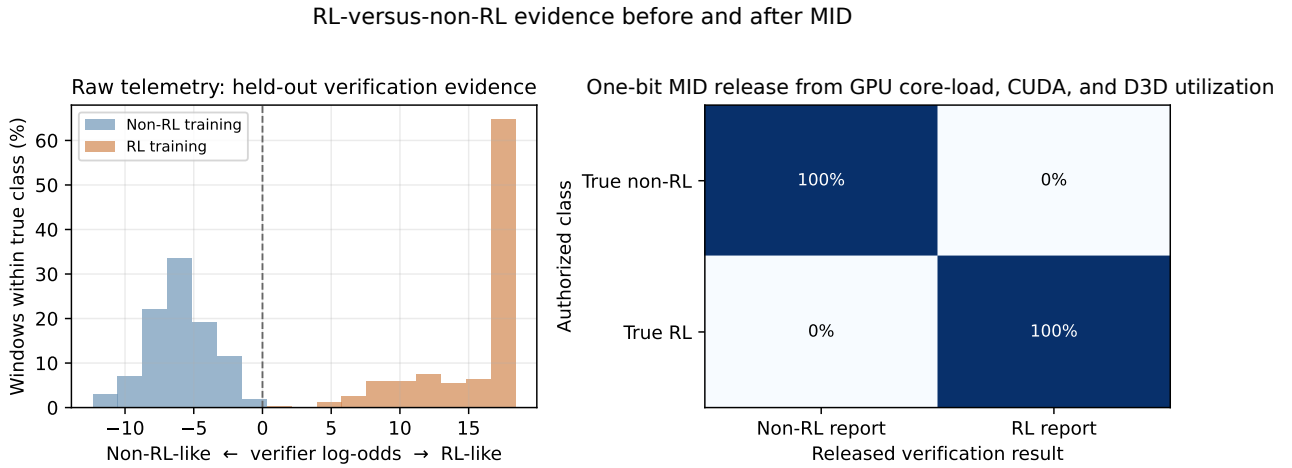


Figure 6: RL-versus-non-RL held-out verification evidence and the selected one-bit report. The left panel shows a classifier score (log-odds) computed from rich telemetry; the right panel shows the released RL/non-RL bit.

Exact-workload evidence at fixed Y = non-RL

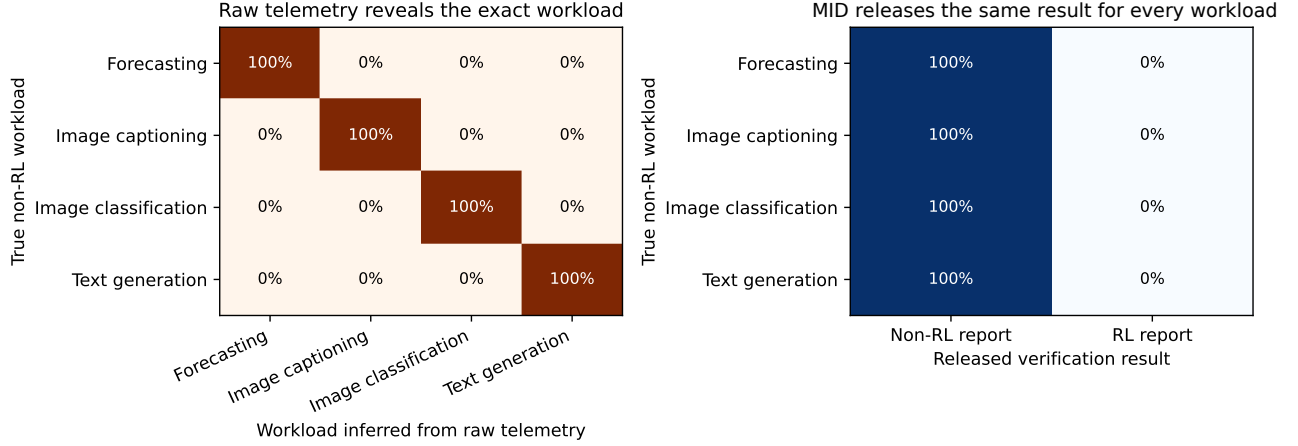


Figure 7: Exact-workload evidence at fixed non-RL. Raw telemetry identifies forecasting, image captioning, image classification, and text generation; MID releases the same non-RL result for all four workloads.

4.3 Hardware Identity

We next apply MID to hardware identity: whether an approved accelerator executed the workload without revealing the workload class. After removing byte-identical copies, we use diffusion and large language model (LLM) configurations from Elsayed et al. that were measured on both eight-GPU H100 and B200 nodes [62]. RTX 3060 traces lack matched eight-GPU configurations and are excluded. Each H100/B200 pair shares one of five held-out folds. The authorized target is chip identity; the sensitive target is diffusion versus LLM within each chip.

The raw baseline contains 183 summaries of power, utilization, memory, and temperature. Candidate mechanisms use full telemetry, dynamic telemetry without static capacity, GPU power, utilization, memory activity, temperature, scale-independent temporal shape, or reported memory capacity. MID selects total reported GPU memory capacity

$$V_i = \text{mean}_t \sum_{g=1}^8 \text{memTotal}_{i,g}(t).$$

Inside fold f , the threshold is fitted using training sessions only:

$$b_f = \frac{1}{2} \left(\max_{i \in \text{train}_f: Y_i = \text{H100}} V_i + \min_{i \in \text{train}_f: Y_i = \text{B200}} V_i \right).$$

The released evidence is

$$Z_i = \begin{cases} 0, & V_i \leq b_f \quad (\text{H100}), \\ 1, & V_i > b_f \quad (\text{B200}). \end{cases} \quad (24)$$

When each matched group is scored by a model trained on the other groups, the telemetry verifies the chip perfectly but identifies diffusion versus LLM with balanced accuracy 0.8611 on H100 and 0.8889 on B200. The one-bit report preserves perfect chip verification and is constant across workloads within each chip. This analysis assumes that reported capacity is authentic; deployment requires the authenticated source specified in Section 2.1.

Table 4 and Figure 8 show that the dynamic behavior recorded in rich telemetry is unnecessary for this authorized claim. Raw telemetry and authenticated memory capacity both retain one full bit of

Table 4: H100-versus-B200 results. Raw telemetry verifies chip identity perfectly but also distinguishes diffusion from LLM workloads with 0.8750 BA. The MID-selected memory-capacity report preserves perfect chip verification while reducing workload inference to chance and measured conditional leakage from 0.5016 bits to zero.

Evidence	Y BA	$S \mid Y=\text{H100 BA}$	$S \mid Y=\text{B200 BA}$	Mean $S \mid Y$ BA	$I(Y; E)$	$I(S; E \mid Y)$
Raw telemetry	1.0000	0.8611	0.8889	0.8750	1.0000	0.5016
Memory-capacity MID report	1.0000	0.5000	0.5000	0.5000	1.0000	0

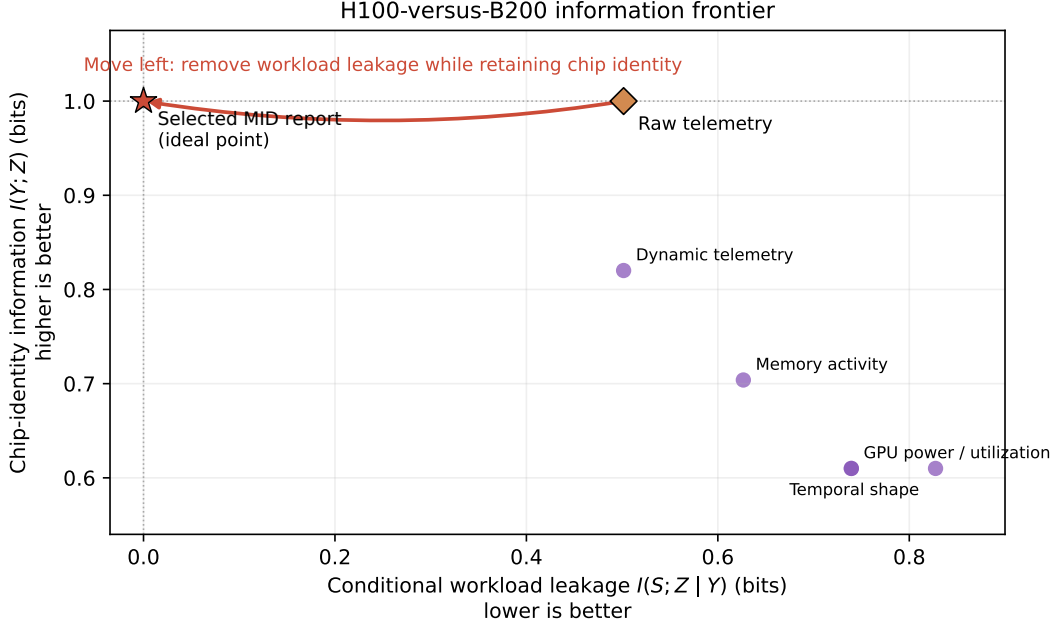


Figure 8: H100-versus-B200 information frontier. Each candidate uses a different telemetry channel or channel family. The experiment shows that MID can choose what to measure, not only how to alter a fixed measurement.

chip-identity information, but only the dynamic telemetry exposes substantial workload information. By searching across evidence channels, MID finds that a static capacity report reaches the empirical ideal, whereas several dynamic channels lose chip utility while retaining more workload leakage.

Figures 9 and 10 show why this channel choice works. The capacity report remains perfectly separated by H100 versus B200, but within either chip it is identical for diffusion and LLM workloads. MID therefore removes the workload distinction without weakening the hardware-identity claim.

4.4 Compute Scale

We then turn to compute-scale verification, where the policy may target a coarse resource class or an exact declared allocation. We evaluate both using processor-level and facility-level power.

Coarse execution tier using DeepTheft traces. Compute-monitoring proposals seek to verify resource limits without exposing models, data, or hyperparameters [7, 6]. The dataset accompanying DeepTheft contains variable-length processor-package and dynamic random-access memory (DRAM) power traces collected through Intel’s Running Average Power Limit (RAPL) interface during deep neural network executions [49]. We define Y from trace duration: the bottom 30% is low tier, the top 30% high tier, and the middle 40% is excluded. This laboratory target tests disclosure control for a coarse duration class; it is not a proxy for statutory training-compute thresholds expressed as a number of floating-point operations (FLOP) or for frontier-training scale.

H100-versus-B200 evidence before and after MID

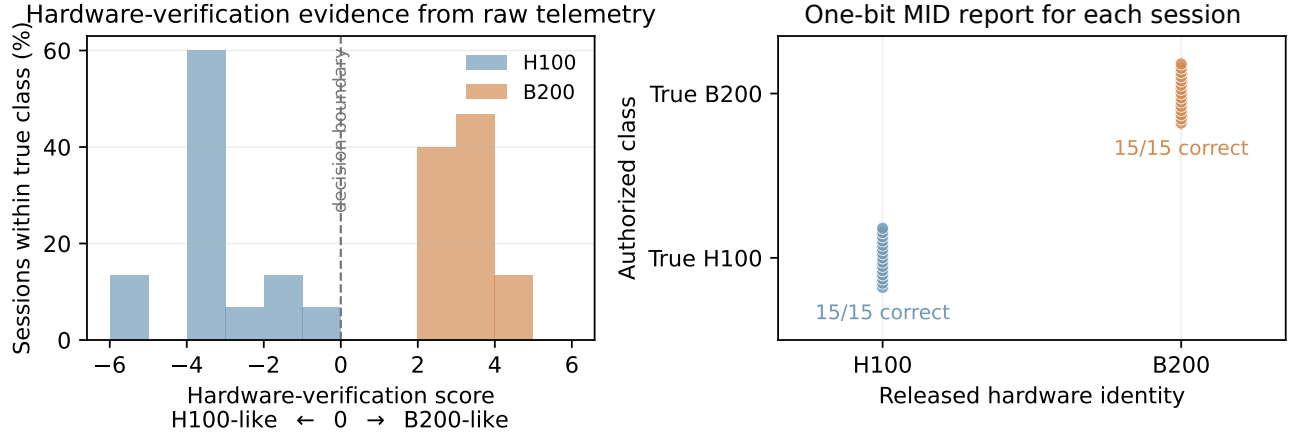


Figure 9: H100-versus-B200 verification before and after MID. The left panel visualizes the multivariate raw telemetry with a held-out log-odds score; the right panel shows the actual one-bit hardware report. Both rich telemetry and the selected one-bit capacity report identify the chip perfectly, showing that the richer dynamic trace is unnecessary for the authorized hardware claim.

Conditional workload leakage before and after MID

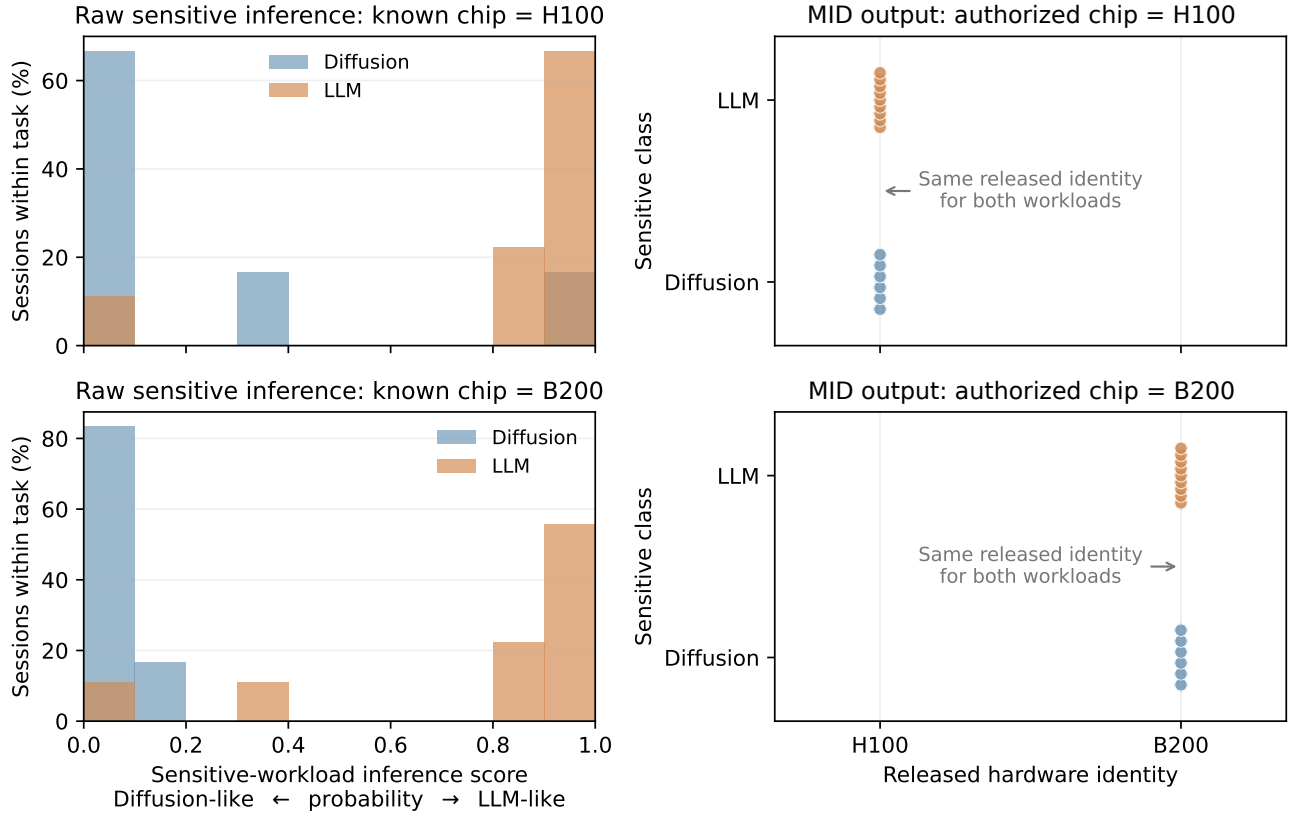


Figure 10: Conditional diffusion-versus-LLM evidence within each authorized chip. Rich telemetry distinguishes the workloads, while both workloads induce the same MID released chip result, hence zero leakage.

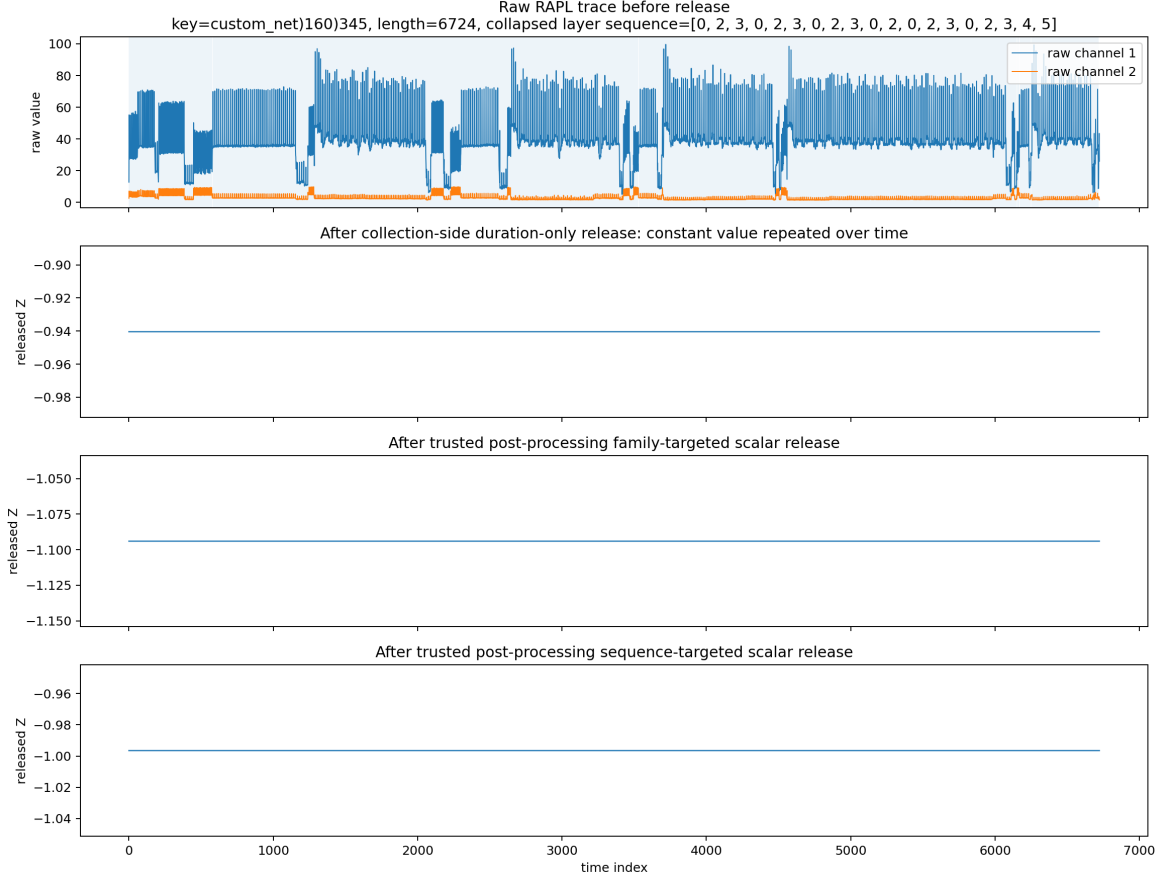


Figure 11: One execution from the DeepTheft dataset before and after release design.

We treat model family and architecture as separate sensitive targets and design a release for each. The first sensitive target contains six model families,

$$S_{\text{family}} \in \{\text{custom_net}, \text{custom_net_bn}, \text{resnet_basicblock}, \text{resnet_bottleneck}, \text{vgg}, \text{vgg_bn}\}.$$

The second is the architecture tuple

$$S_{\text{arch}} = (n_{\text{layers}}, n_{\text{conv}}, n_{\text{pool}}, n_{\text{linear}}, \text{max output channels}),$$

which yields five classes after retaining only architecture categories represented in both compute tiers. Architecture metadata supplies labels only; no architecture field is an attack input or released feature.

The rich baseline is a 256-dimensional downsampled trace. The collection-time alternative is normalized log duration

$$d_i = \frac{\log(1 + T_i) - \mu_T}{\sigma_T},$$

where T_i is trace length and μ_T, σ_T are fitted on the design split. Candidate scalar releases computed privately after collection use the trace representation together with d_i , giving 257 private features. Figure 11 makes the disclosure difference concrete: the raw RAPL waveform exposes detailed temporal structure used to recover model architecture, whereas each MID interface releases only one scalar. The horizontal repetition in the figure only visualizes that scalar on the waveform’s time axis; the verifier receives one value rather than a time series.

Table 5 compares MID’s two deployment paths. Duration can be collected directly and verifies the duration-defined compute tier perfectly while substantially reducing model-family leakage. When

Table 5: DeepTheft matched-support frontier. Sensitive balanced accuracy is averaged within compute tier.

Sensitive target	Evidence	Mechanism	β	σ	Y BA	$S \mid Y$ BA	D_S^{cond}
Model family (6 classes)	Raw features	raw baseline	–	–	0.9099	0.7211	1.8438
	Duration only	collection time	–	–	1.0000	0.2929	0.0829
	Minimum D_S^{cond}	trusted scalar	1000	0.10	0.9897	0.1419	0.0024
	Maximum utility	trusted scalar	3	0	0.9989	0.2875	0.0416
Architecture bucket (5 classes)	Raw features	raw baseline	–	–	0.8893	0.4754	2.6191
	Duration only	collection time	–	–	1.0000	0.3469	0.1113
	Minimum D_S^{cond}	trusted scalar	10	1.00	0.8661	0.2021	0.0167
	Maximum utility	trusted scalar	300	0.10	1.0000	0.2163	0.0095

Table 6: DeepTheft attack evaluation with the attack models, labels, splits, and training schedule held fixed. Replacing only the raw RAPL input with a MID release reduces layer-sequence recovery from 99.1% to 11.5% and macro F1 from 99.8% to 25.8%.

Attack input	Mechanism type	Epochs	Step 1 sequence score (%)	Step 2 acc. (%)	Step 2 F1 (%)
Raw RAPL trace	raw baseline	3	99.053	99.956	99.845
Duration only	collection time	3	11.611	63.149	25.804
Family-targeted scalar	trusted post-collection	3	15.668	63.149	25.804
Sequence-targeted scalar	trusted post-collection	3	11.543	63.149	25.804

that remaining disclosure is unacceptable, a post-collection scalar reduces family inference from 0.7211 balanced accuracy under raw evidence to 0.1419, near six-class chance, while retaining 0.9897 compute-tier accuracy. For architecture, the maximum-utility scalar retains perfect compute-tier verification while reducing conditional architecture inference to 0.2163, near five-class chance. The result shows when MID’s post-collection path adds value: it suppresses information that remains in the best available direct measurement.

Balanced chance is $1/6 \approx 0.167$ for model family and $1/5 = 0.20$ for architecture. The held-out prediction accuracies of the minimum- D_S^{cond} candidates are 0.1419 for model family and 0.2021 for architecture, indicating little conditional discrimination by these models.

Figures 12 and 13 show the two requirements simultaneously. The duration and selected scalar releases retain separation between the authorized low- and high-compute tiers, while the model-family distributions that are clearly separated in the raw evidence become strongly overlapping. MID therefore preserves the coarse verification claim while removing the temporal and distributional structure used to infer the protected model family.

To test whether the low-information releases weaken the DeepTheft attack, we rerun Steps 1 and 2 using Gao et al.’s targets and splits [49]. We keep the three-epoch schedule fixed and replace only the attack input. Because the implementation expects a time series, we repeat each scalar release at every time step and add a small ϵ to UPloss to prevent NaN values. Step 1 recovers the ordered layer-type sequence; we report its Levenshtein Distance Accuracy as the sequence score. Step 2 recovers layer hyperparameters and reports macro F1. Table 6 shows near-perfect recovery from raw RAPL traces. Low-information releases reduce the sequence score to 11.5–15.7% and macro F1 to 25.8%, with the sequence-targeted scalar performing best on its intended label.

Step 2 is class-imbalanced: its 63.149% accuracy under every low-information release accompanies macro F1 25.804%, consistent with majority-class behavior. Macro F1 is therefore the informative Step 2 metric. Because the attack implementation is unchanged and only its evidence is replaced, this collapse shows that MID’s disclosure reduction transfers to the original DeepTheft attack, not only to the information estimator used during mechanism selection. This is fundamentally due to MID being an attack-agnostic information-theoretic framework.

Released evidence distribution by authorized target Y

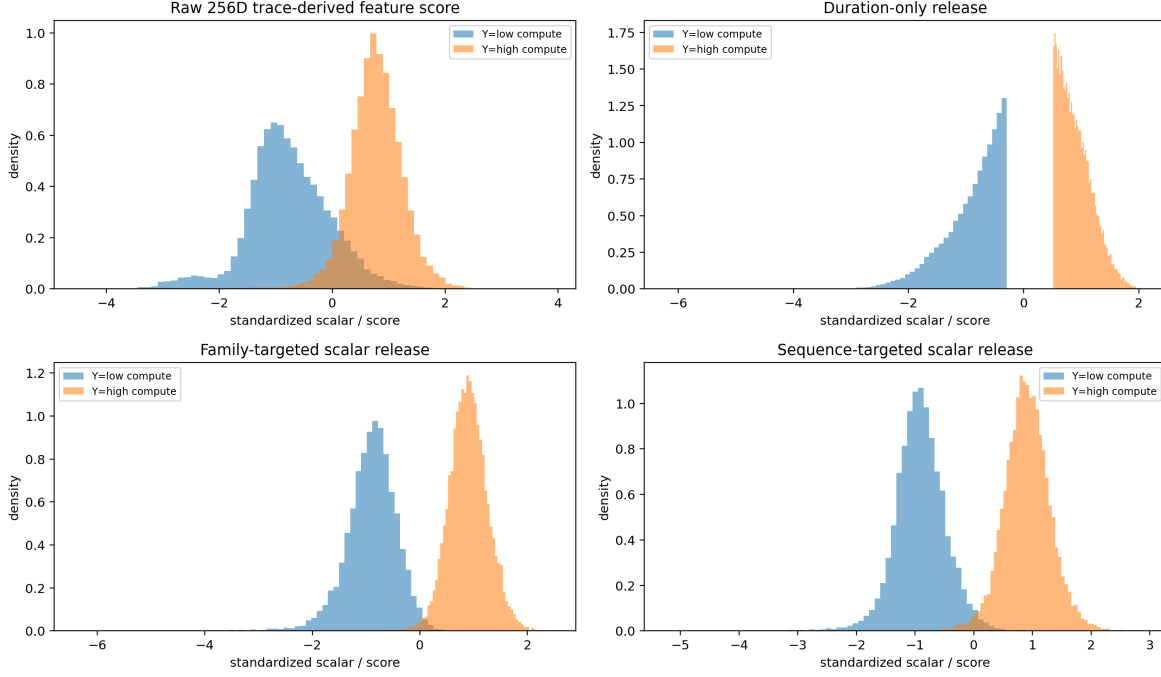


Figure 12: DeepTheft evidence grouped by authorized compute tier. The duration and scalar releases preserve the low-versus-high distinction after removing the detailed waveform.

Table 7: Groth16 zk-SNARK arithmetic implementation for one scalar report derived from DeepTheft traces.

Quantity	Value	Quantity	Value
Curve	BN254 (bn128)	Circuit constraints	258
Private inputs	258	Public inputs	259
Serialized proof	4 KB	Verification key	52 KB
Proving time	1.111 s	Verification time	1.006 s

ZKP-backed release. We compile the family-targeted scalar into a Groth16 circuit [23] for Eq. (22). The private input contains 256 RAPL features, log duration, and one noise value; the fixed-point scale is 1,000 and $\sigma = 0.10$. For the demonstrated instance, $\tilde{\sigma} = 100$, $\tilde{\xi} = 126$, $\sum_j \tilde{w}_j \tilde{X}_j = -2,181,841$, and $\tilde{Z} = -2,169,241$ —Table 7 reports its cost. Verification accepts the scalar field-arithmetic relation without revealing $\tilde{X}$ or $\tilde{\xi}$.

Declared node allocation on NLR. The NLR dataset combines measured workload profiles with facility power estimated by a bottom-up model of the power-delivery components [61]. Our processed training corpus contains modeled profiles for Llama-2 70B LoRA and Stable Diffusion runs across 2, 4, 8, and 16 nodes. Y is exact node count and S is workload identity within node count. Here, a node means one Kestrel GPU compute server participating in the distributed job—not one GPU. Each Kestrel GPU node contains four NVIDIA H100 GPUs. The raw baseline contains 27 summaries of duration, energy, power levels and quantiles, temporal variation, autocorrelation, and spectral entropy. Workload identity is a meaningful sensitive target because GPU power can vary with input values even when operation shapes are fixed [31]; an infrastructure signal that verifies aggregate allocation may therefore reveal computation details.

Conditional histograms for sensitive model family S inside fixed authorized class Y

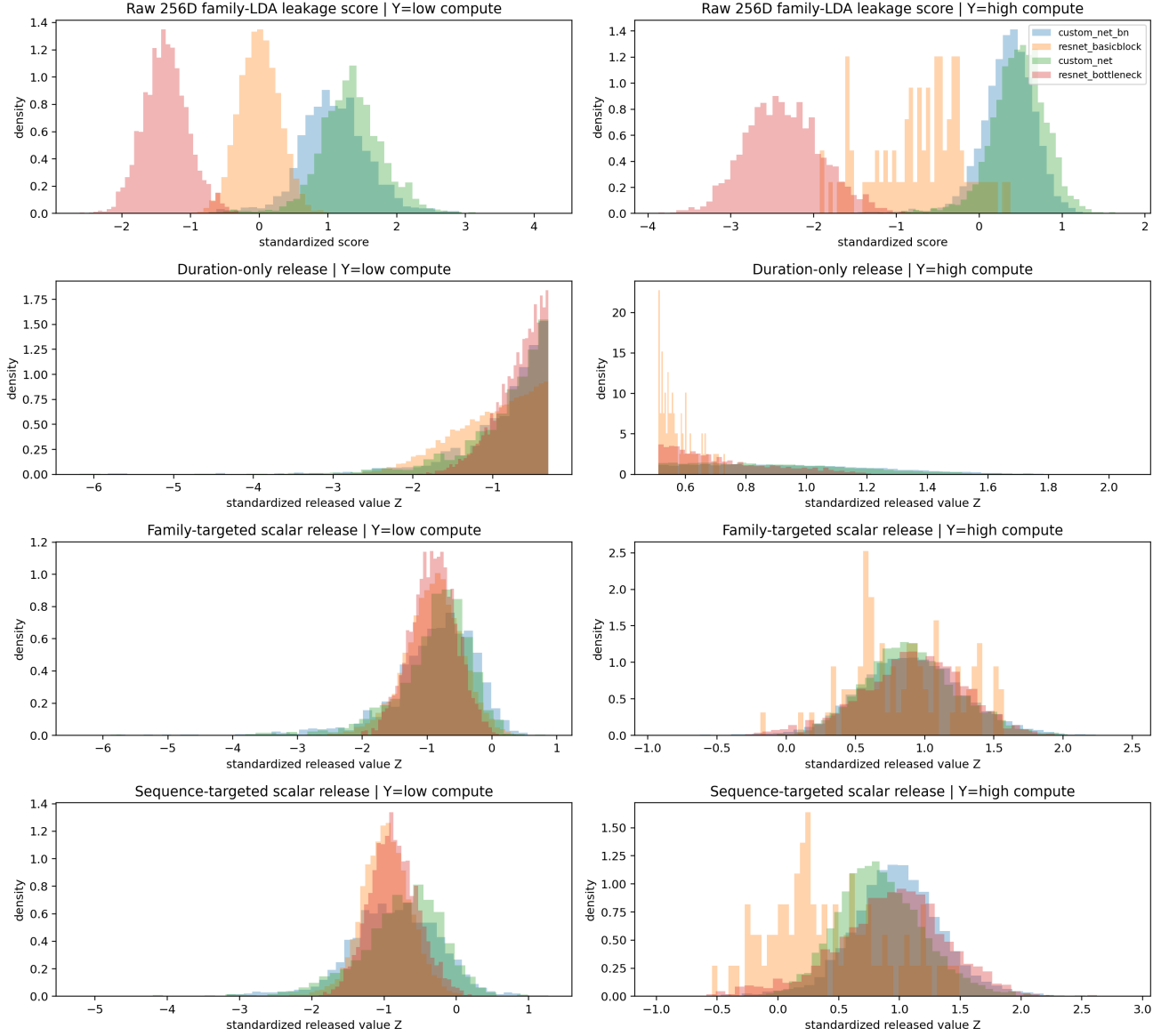


Figure 13: DeepTheft model-family evidence within fixed compute tier. Raw trace features contain strong family-separating directions. Duration reduces but does not eliminate that structure; the family-targeted scalar moves conditional inference close to six-class chance. In the plot label, LDA denotes the linear discriminant analysis model used only to create the visualization score.

Table 8: NLR node-allocation verification and conditional workload leakage. Mutual information is reported only for the discrete MID channel and is a held-out plug-in estimate.

Evidence	Y BA	$S \mid Y$ BA	$I(Y; E)$	$I(S; E \mid Y)$
Raw trace features (27)	0.9773	0.9750	–	–
Raw $\log_2$ mean power	1.0000	0.8729	–	–
MID discrete report	1.0000	0.5000	1.9952	1.7×10^{-16}
Balanced chance	0.2500	0.5000	–	–

We apply candidate meter reports to $\log_2$ mean power over the following values:

$$q \in \{0, 0.0625, 0.125, 0.25, 0.50, 0.75, 1.00\}, \quad \sigma \in \{0, 0.025, 0.05, 0.10, 0.20, 0.30, 0.40\},$$

where $q = 0$ denotes no quantization and σ is the Gaussian-noise standard deviation in log-power units. Selection minimizes conditional workload leakage subject to node-count balanced accuracy of at least 0.90. The selected deterministic report is

$$Z = \text{round}(\log_2 \bar{P} - b), \quad b = 0.134, \quad (25)$$

where $\bar{P}$ is mean facility power. It maps 2, 4, 8, and 16 nodes to codes 12, 13, 14, and 15. Nested four-fold selection chooses $q = 1$ and $\sigma = 0$ independently in every outer fold using training data only; fitted offsets range from 0.1329 to 0.1778. The report is correct on every held-out run, while workload balanced accuracy is 0.5 within every node count.

Four equally likely node classes carry two bits. For this experiment we instead use the observed run frequencies, whose entropy is $H(Y) = 1.9952$ bits. On the held-out runs, the discrete report retains all of it: $I(Y; Z) = 1.9952$ bits, while the plug-in estimate $I(S; Z \mid Y) = 1.7 \times 10^{-16}$ bits is numerically zero.

Figure 14 places the selected report on the measured privacy–utility frontier: it retains all available node-count information while reaching numerically zero measured conditional workload information.

When every power value is multiplied by the same calibration factor, the fixed quantizer remains perfect from -15.5% to $+18.5\%$ error; failures begin at -16% and $+19\%$. This sweep measures tolerance to uniform calibration error.

Figure 15 visualizes this tolerance: verification remains perfect throughout the stated interval and begins to fail only after the calibration error crosses a quantization boundary.

Figure 16 shows that the zero-leakage result follows directly from the released code. Each node allocation maps to a distinct value, so the verifier recovers the complete authorized allocation. Within each fixed node count, however, Llama-2 LoRA and Stable Diffusion map to exactly the same value. The release therefore preserves all measured allocation information while providing no measured basis for distinguishing the protected workloads.

4.5 Model Identity

Finally, we apply MID to model identity: whether an approved model type executed without exposing its internal structure. Our corpus, filtered from the ModelSpy dataset, contains GPU EM traces from CNN and Transformer executions labeled with model structure [48]. We authorize $Y \in \{\text{CNN}, \text{Transformer}\}$ and protect the finer layer signature S , which records the model’s ordered layer structure. Candidate collection mechanisms use downsampling factors

$$\alpha \in \{1, 32, 48, 64, 80, 96, 112, 128, 160, 192, 256, 320\}.$$

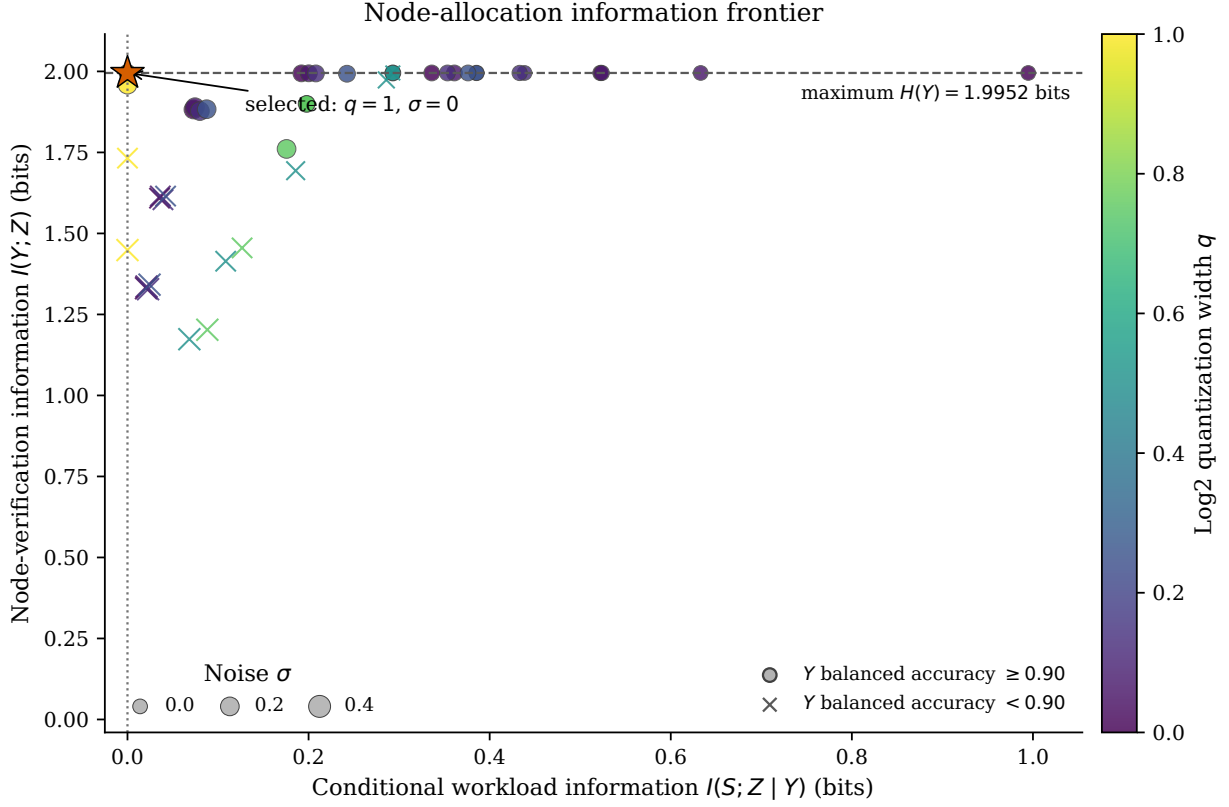


Figure 14: NLR node-allocation information frontier. On the held-out runs, the selected quantized report retains all available node-count information and has numerically zero measured conditional workload information.

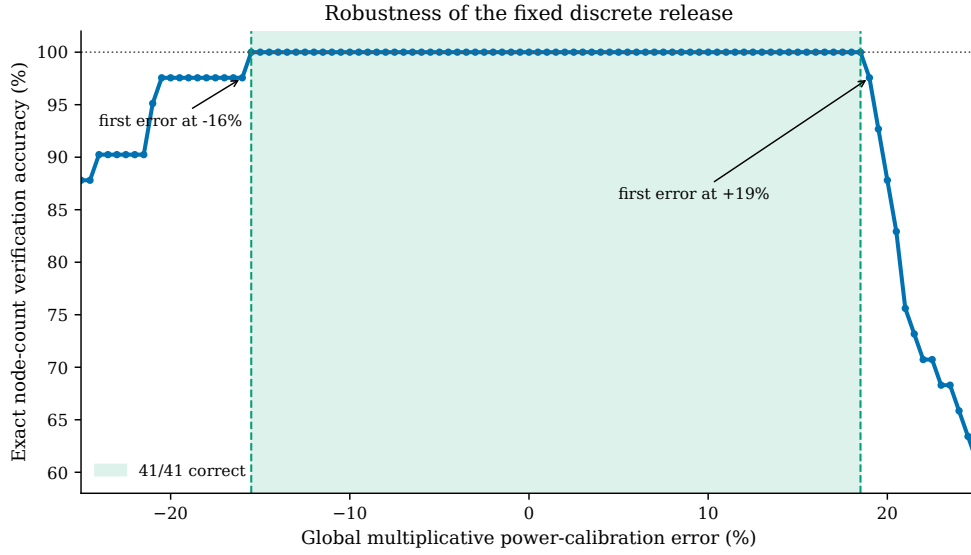


Figure 15: Robustness of the selected node-count report to global power-calibration error. Verification remains perfect from -15.5% to $+18.5\%$ error, showing that the low-information discrete release does not depend on exact power calibration within this interval.

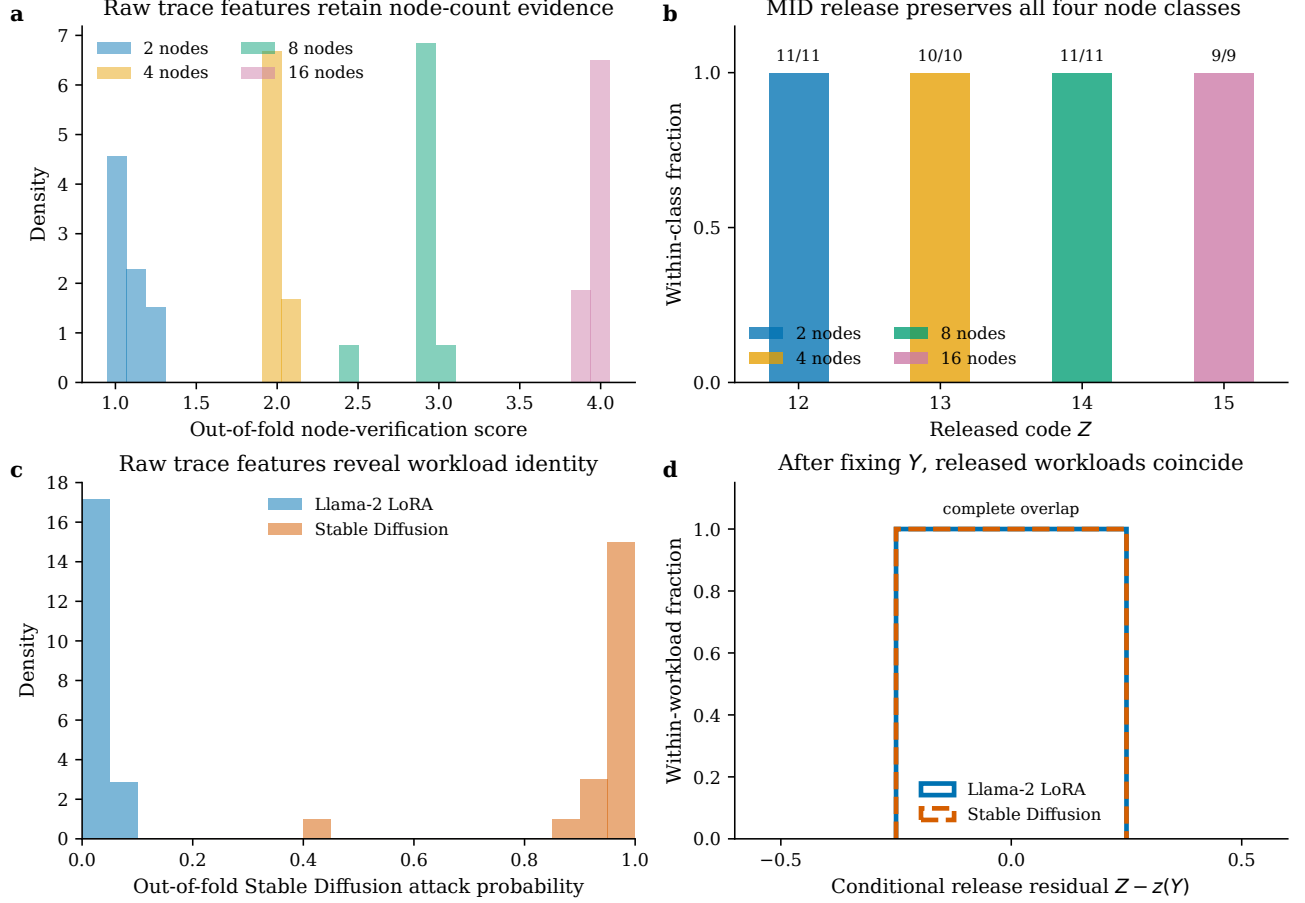


Figure 16: NLR node-allocation evidence. The MID-selected report retains four non-overlapping node codes. At each fixed node count, Llama-2 LoRA and Stable Diffusion induce the same released value, hence zero leakage.

We simulate lower-rate collection by retaining every α -th recorded sample. This assumes the same sensor and analog electronics. In deployment, the sensor would collect at the selected lower rate, so the high-rate digital trace would never be created.

Among candidates with Y balanced accuracy at least 0.90, $\alpha = 192$ minimizes D_S^{cond} . Relative to the raw rate, authorized balanced accuracy changes from 0.9997 to 0.9046, D_S^{cond} falls from 64.2547 to 7.2474, and scores from the ModelSpy layer and hyperparameter attacks [48] fall from 0.9785 and 0.9468 to 0.5458 and 0.5952, respectively.

Figure 17 shows what MID changes at collection time. The selected lower-rate channel retains the coarse execution pattern needed for CNN-versus-Transformer verification while avoiding most of the fine-grained digital samples available to ModelSpy. This is a privacy advantage of mechanism design at the evidence source rather than post-processing a high-rate trace after it has already been disclosed.

Figures 18 and 19 show the resulting tradeoff. At $\alpha = 192$, the CNN and Transformer score distributions remain sufficiently separated to satisfy the required 0.90 authorized balanced accuracy. Within the fixed CNN class, however, the layer-signature distributions overlap substantially more, corresponding to large reductions in the ModelSpy layer and hyperparameter attack scores. MID therefore removes fine model-structure evidence while retaining the coarser model-type distinction.

Table 9 gives the complete sweep. Leakage does not decrease smoothly between $\alpha = 160$ and 192. MID therefore selects from the measured frontier rather than assuming that more downsampling always improves privacy.

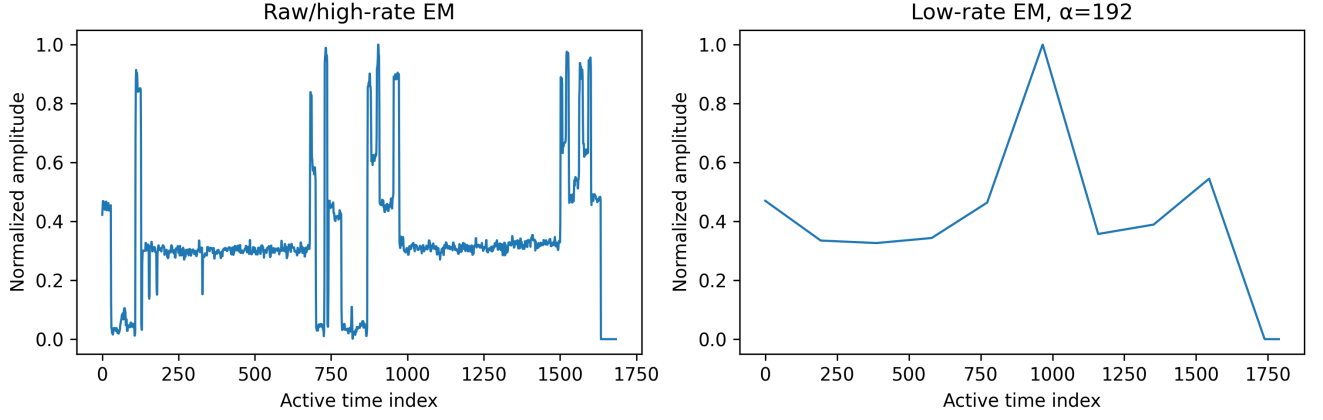


Figure 17: Representative ModelSpy EM evidence at the raw rate and selected lower-rate factor $\alpha = 192$. The selected collection policy retains the coarse activity pattern used for model-type verification while avoiding most of the fine temporal detail exploited for architecture inference.

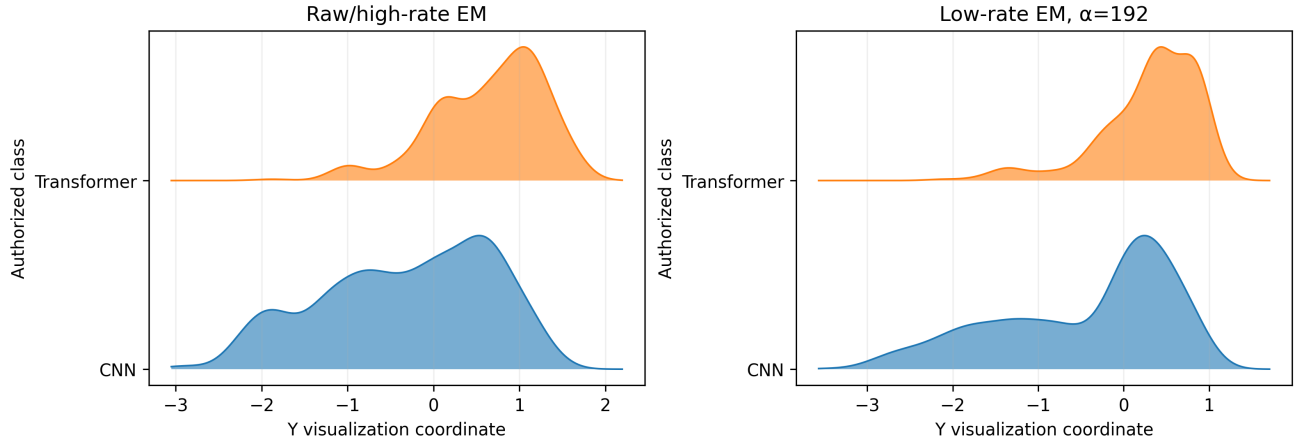


Figure 18: Authorized CNN-versus-Transformer evidence before and after lower-rate collection. The selected $\alpha = 192$ mechanism retains sufficient class separation to meet the verification requirement, so its privacy improvement is not obtained by making the evidence useless for model-type verification.

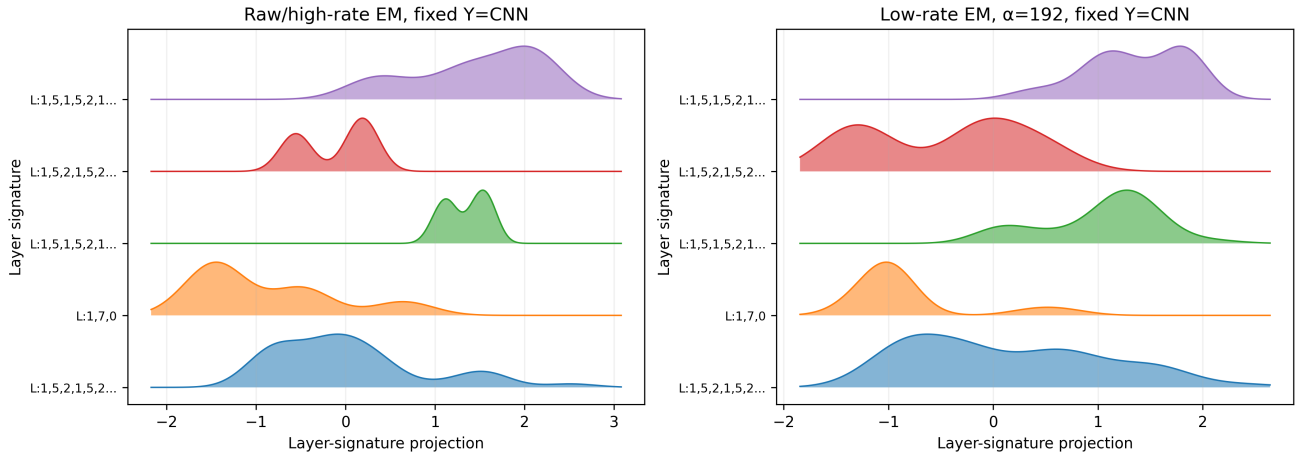


Figure 19: Protected layer-signature evidence for the five most frequent layer signatures at fixed $Y = \text{CNN}$. Lower-rate acquisition makes the signature distributions substantially more overlapping, consistent with the reduction in ModelSpy’s layer and hyperparameter attack scores.

Table 9: ModelSpy lower-rate EM sweep. MID selects $\alpha = 192$ because it has the lowest measured conditional disclosure among candidates satisfying Y balanced accuracy of at least 0.90. Candidates with still lower disclosure fail the verification requirement, showing why the mechanism must be selected from the measured privacy–utility information frontier rather than by maximizing downsampling.

α	Y acc.	Y bal. acc.	D_Y	D_S^{cond}	Layer-attack score	Hyperparameter-attack score
1	0.9994	0.9997	10.4196	64.2547	0.9785	0.9468
32	0.9889	0.9889	5.5503	42.7931	0.7335	0.7329
48	0.9823	0.9820	5.0442	30.6026	0.6620	0.6905
64	0.9768	0.9848	3.0585	22.6998	0.6408	0.6723
80	0.9453	0.9617	3.3963	17.6194	0.6163	0.6469
96	0.9558	0.9557	2.0929	14.9535	0.6080	0.6421
112	0.9176	0.9440	1.7084	11.9663	0.5865	0.6186
128	0.9022	0.9333	1.5643	11.0527	0.5766	0.6176
160	0.8530	0.8922	1.5446	8.0075	0.5617	0.5941
192	0.8507	0.9046	1.8943	7.2474	0.5458	0.5952
256	0.7584	0.8397	0.8321	5.2068	0.4857	0.5628
320	0.6987	0.7970	0.6889	4.0702	0.4430	0.5337

5 Discussion

The evaluation shows that MID is not a single release mechanism but a way to choose what evidence should cross the verification boundary under a stated policy. We now discuss the scope of that protection and how the selected interface integrates with existing verification systems.

MID designs what the evidence reveals. Across the experiments, MID controls disclosure in different ways: telemetry channels for hardware identity and RL detection, lower-rate acquisition for ModelSpy EM traces [48], duration or a private projection for DeepTheft power traces [49], and compact quantized reports for the NLR facility-power data [61]. Searching only over noise added after collection would miss several of these solutions. No construction is universally optimal: the allowed controls and required utility determine the best achievable tradeoff.

What MID protects. MID protects a declared property S against disclosure beyond an authorized result Y under the design distribution represented by the development data. Conditional mutual information measures that disclosure; for the corresponding population quantity, the data-processing inequality ensures that later processing of the release cannot increase its information about S . A deployment specifies (Y, S, τ_Y) before mechanism selection, validates on grouped physical runs, and monitors distribution shift. Once policy specifies the authorized claim and protected properties, MID makes the disclosure boundary quantitative and auditable. Differential privacy addresses a different question by bounding worst-case changes under a neighboring-execution relation.

What the experiments show. The experiments evaluate the complete mechanism-selection process on held-out physical sessions or runs. In three tasks, the selected discrete report preserves perfect verification and has zero measured conditional leakage on held-out data. The remaining tasks yield explicit privacy–utility frontiers, while the ModelSpy and DeepTheft evaluations show how the selected releases weaken concrete inference attacks [48, 49]. The same framework can search richer mechanism catalogs and apply Eq. (4) when a deployment requires several protected properties.

Integration with existing verification methods. MID complements systems that authenticate evidence, attest execution, or prove computation over private inputs. A deployment first states the authorized result Y , the protected property S , and the required verification utility. The existing verification system then supplies the mechanisms it can implement with its integrity guarantees: available measurements, collection rules, reports, and required verifier-visible transcript. MID treats each complete verifier-facing interface as E_m , including the report and any metadata, attestation, proof, or failure signal the verifier receives, and selects the one that meets the verification requirement while revealing the least about S beyond Y . Source authentication or attestation binds evidence to the claimed device or execution; when a release is computed from committed private evidence, a ZKP certifies that computation.

Protected hardware and confidential execution. The flexHEG guarantee processor and Guaranteeable Memory chiplet are designed to observe accelerator activity locally and authenticate detailed receipts or higher-level claims [4, 16]. Attestable Audits runs an agreed benchmark over a confidential model and audit data and publishes an attested aggregate result, while PAL*M attests properties of generative-model operations without exposing the underlying models or datasets [44, 26]. MID can compare the measurements, sampling policies, and report functions that each platform can implement. When the selected low-information measurement can be formed at collection time, the monitor need not create or retain a richer trace; otherwise the detailed measurement remains inside the protected boundary and only the selected report and required attestation fields leave it.

Network taps. Cankaya’s architecture captures network traffic while keeping its plaintext inside the monitored facility. The verifier receives signed commitments to that traffic and may challenge selected records, which are checked inside an air-gapped auditing environment. The verifier then receives only a report agreed in advance [8]. MID treats the commitments and report as the verifier-facing evidence and selects the report design that meets the verification requirement while revealing the least about the protected property. If the report exactly communicates the authorized result and the remaining evidence reveals nothing further, the interface is already ideal under MID. The underlying system remains responsible for the integrity of the network tap and auditing process.

ZKPs and confidential verification. ZkAudit proves an agreed audit function over committed private data or model weights, while South et al. prove inference and aggregate evaluation metrics for models with committed private weights [41, 22]. Peigné et al. propose a dense-training architecture whose public commitments, network anchors, policy bounds, and proof transcript verify a private training specification [33]. MID can select among the audit functions, aggregate metrics, or policy claims supported by these protocols and treats every field revealed to the verifier as part of the evidence; the ZKP then proves the selected statement. In our DeepTheft demonstration, a Groth16 zk-SNARK certifies one MID-selected linear projection of a private measurement; in deployment, the surrounding evidence system authenticates the measurement’s source [23].

6 Related Work

MID sits at the intersection of constrained disclosure, confidential verification, evidence generation, and information-theoretic privacy. The central distinction is that MID selects verifier-facing evidence according to what it reveals about a protected property beyond the authorized result.

Privacy requirements in AI verification. Prior work constrains what reaches the verifier in several ways: single-bit or yes/no reports [6, 3, 9], highly specified disclosures [2], and on-device aggregation of detailed receipts into higher-level claims [4]. The Agentic Witnessing protocol restricts each answer to four values and budgets the number of queries, yielding a worst-case bound on the capacity of a session [10]. These approaches constrain the form or maximum amount of disclosure, but they do not decide which feasible interface reveals the least about a declared protected property beyond the authorized answer. MID solves that selection problem subject to the required verification utility.

Confidential verification. Attestable Audits runs a model and benchmark inside a TEE, publishes an attested aggregate result, and keeps the model weights and audit code and data confidential; PAL*M attests properties of generative-model operations using confidential CPU–GPU execution without exposing the underlying models or datasets [44, 26]. ZkAudit proves a verifier-supplied audit function over committed private data and weights, whereas South et al. prove inference and aggregate evaluation metrics for models with committed private weights [41, 22]. Peigné et al. propose an architecture that combines a precommitted training specification, network observations, Merkle commitments, and ZKPs to verify predeclared training claims [33]. These systems protect the inputs to a selected computation but leave the verifier-facing result to the audit designer; MID supplies a quantitative criterion for selecting that result and the mechanism that produces it. Li et al. instead protect individual records with local differential privacy and infer preprocessing errors by comparing black-box model behavior with reference models trained under candidate preprocessing pipelines, explicitly measuring the resulting privacy–verification tradeoff [40]. MID applies to a different object: the information disclosed by verifier-facing execution evidence about a declared protected property beyond the authorized result.

Evidence generation, soundness, and information flow. Proof-of-learning and training-data-verification methods provide evidence about training history or provenance, while subsequent work demonstrates serious soundness weaknesses in current proof-of-learning designs [42, 43, 32]. Inference-verification methods use recomputation, token divergence, or activation fingerprints [17, 18]. Guaranteeable Memory places a chiplet at the HBM interface to attest memory snapshots, sampled operations, selected memory regions, or locally checked workload claims; Cankaya instead commits network traffic through signed hashes and evaluates challenged preimages inside an air-gapped auditing environment [16, 8]. Monfared et al. expose GPU timing and memory telemetry and discuss timing leakage, while Rahman and Tajdari use telemetry to classify hidden training workloads [28, 27]. These methods define candidate evidence channels and integrity paths for MID; MID determines which authenticated interface should reach the verifier. GPU-verification systems also select telemetry features to support reliable verification. WAVE selects performance-counter events to reconstruct model structure and size and proposes a TEE or in-GPU verifier when only the verification result should leave. ShadowScope chooses performance-monitoring events and evaluates how sampling rate affects robust kernel validation [29, 30]. These systems design evidence for identification and robustness; MID additionally selects it according to what it reveals about a declared S beyond Y . Another information-theoretic line in AI verification controls deliberate exfiltration through operational outputs. Rinberg et al. bound steganographic model-weight exfiltration through inference responses. Petrie et al. define unexplained information in a prover’s network outputs relative to a declared computation, budget that channel across content, timing, and metadata, and allow the prover to optimize its implementation under that budget [19, 20]. Those objectives limit information deliberately transmitted through operational outputs. MID measures a different flow: collateral disclosure from legitimate verifier-facing evidence.

General disclosure and information-flow context. Outside AI-execution verification, structured transparency distinguishes input from output privacy and calls non-essential information revealed by a flow “collateral information leakage” [21]. The information bottleneck compresses a representation while preserving task-relevant information [35]. The privacy funnel and subsequent privacy–utility work optimize release mechanisms to suppress designated sensitive variables under utility and observation constraints [36, 37, 38, 39]. These works provide the general privacy and information-theoretic context; MID formulates the evidence-design problem for AI verification and links the selected release to an authenticated direct measurement or a ZKP-certified private computation. The breadth of inference attacks on model interfaces and physical traces motivates this information objective rather than a separate defense for each attack [51, 52, 53, 55, 54, 48, 49, 50, 46, 31, 47].

7 Conclusion

AI verification must reveal enough to establish a claim without exposing unnecessary details about the execution. MID turns this tension into a quantitative design problem. Given an authorized target Y , a protected property S , and a verification requirement τ_Y , MID searches across evidence channels, collection policies, and release transformations for an interface that preserves verification while minimizing collateral leakage. Across six physical-measurement tasks, this process selects restricted channels, lower-rate collection, discrete reports, or private projections and reports the resulting privacy–utility frontiers. The selected interface can be deployed through an authenticated direct measurement or a ZKP-backed transformation of a private measurement; we demonstrate the scalar-release arithmetic with a Groth16 zk-SNARK. MID thus makes what verification evidence reveals an explicit, testable design choice.

A Output-Privacy Analysis of Auditor-in-a-Box

Penchas et al. propose a two-party protocol in which both parties sign a Plan that an LLM executes inside an attested TEE [11]. The authors explicitly present a deterministic output filter intended to impose a single-bit worst-case disclosure bound [12]. The reference implementation’s Monitor Query Validation example applies this idea through a restricted verdict interface.¹ In the published evaluation analyzed below, every reported decision is VALID or INVALID. This limits how much leaves the protected system, but does not determine what the observed verdict reveals.

We test this distinction using only the authors’ published evaluation of the usage-monitor classifier.² We analyze only requests that human reviewers marked VALID, so the intended answer is the same for every case. The authors’ case identifiers begin with either B or V. Many B-case descriptions call the request adjacent, proxy, or borderline, while V cases largely describe ordinary aggregate-research tasks. Although all of these requests have the same human judgment, the monitor returns INVALID for 47.4% of B cases and only 2.5% of V cases. The one-bit decision therefore reveals information about what kind of private request was submitted.

Formally, Y is the human judgment, fixed here at VALID; Z is the monitor’s released VALID/INVALID decision; and S records whether the case identifier begins with B or V. For each case, a simple predictor learns from all the other cases which prefix is more likely after each decision. It then receives only the decision for the case being tested and predicts B or V; it does not receive the request text, identifier, or purpose. The case identifiers, descriptions, human judgments, and model decisions all come from the authors’ evaluation; we add no labels or examples.

Table 10: Disclosure and decision-only inference at the fixed authorized answer $Y = \text{VALID}$. Although the interface releases only one bit, 0.176 bits of measured disclosure increases inference of the protected request group from 50.0% to 72.4% balanced accuracy.

Metric	Authorized answer only	With binary decision Z
Measured disclosure beyond Y	0	0.176 bits
Estimated B-group probability	19.4%	81.8% after INVALID
Inference-attack balanced accuracy	50.0%	72.4%

The important result is not merely that the decision carries 0.176 bits. It is that this apparently small disclosure supports a decision-only inference attack with 72.4% balanced accuracy, compared with 50.0% without the decision. The experiment therefore demonstrates the central motivation for MID on an existing verification interface: limiting output cardinality bounds how much can be released, but does not determine what the released bit reveals.

¹Pinned Monitor Query Validation source.

²Pinned published evaluation.

Acknowledgments

This work was supported by  Pivotal Research.

References

- [1] Pacing the Frontier. Pacing the frontier. July 2026. <https://www.pacingthefrontier.com>.
- [2] A. O’Gara, G. Kulp, W. Hodgkins, J. Petrie, V. Immler, A. Aysu, K. Basu, S. Bhasin, S. Picek, and A. Srivastava. Hardware-enabled mechanisms for verifying responsible AI development. arXiv:2505.03742, 2025.
- [3] M. Baker, G. Kulp, O. Marks, M. Brundage, and L. Heim. Verifying international agreements on AI: Six layers of verification for rules on large-scale AI development and deployment. arXiv:2507.15916, 2025.
- [4] J. Petrie, O. Aarne, N. Ammann, and D. Dalrymple. Flexible hardware-enabled guarantees for AI compute. arXiv:2506.15093, 2025.
- [5] A. Reuel, B. Bucknall, et al. Open problems in technical AI governance. *Transactions on Machine Learning Research*, 2025.
- [6] G. Sastry, L. Heim, H. Belfield, et al. Computing power and the governance of artificial intelligence. arXiv:2402.08797, 2024.
- [7] Y. Shavit. What does it take to catch a Chinchilla? Verifying rules on large-scale neural network training via compute monitoring. arXiv:2303.11341, 2023.
- [8] N. Cankaya. A system overview for near-term, low-trust AI compute verification. MIRI Technical Governance Team Working Paper, 2026. <https://techgov.intelligence.org/research/a-system-overview-for-near-term-low-trust-ai-compute-verification>.
- [9] B. Harack, R. F. Trager, A. Reuel, et al. Verification for international AI governance. Oxford Martin AI Governance Initiative, 2025. <https://www.oxfordmartin.ox.ac.uk/publications/verification-for-international-ai-governance>.
- [10] A. Rowstron. Agentic Witnessing: Pragmatic and scalable TEE-enabled privacy-preserving auditing. arXiv:2604.24203, 2026.
- [11] B. Pechas, G. Zhao, and R. Rinberg. Enabling verifiably-scoped monitoring through large language models and trusted compute. In *ICML Workshop on Technical AI Governance Research*, 2026. <https://openreview.net/forum?id=AWZJq607Cj>.
- [12] R. Rinberg and B. Pechas. Auditor-in-a-Box: Tools for third-party auditing. LessWrong, July 2026. <https://www.lesswrong.com/posts/uWYk7MM9hAf9GEBGe/auditor-in-a-box-tools-for-third-party-auditing>.
- [13] A. Scher and L. Thiergart. Mechanisms to verify international agreements about AI development. arXiv:2506.15867, 2025.
- [14] O. Aarne, T. Fist, and C. Withers. Secure, governable chips: Using on-chip mechanisms to manage national security risks from AI and advanced computing. Center for a New American Security, 2024.
- [15] S. Ansari. Hardware-level governance of AI compute: A feasibility taxonomy for regulatory compliance and treaty verification. arXiv:2604.04712, 2026.
- [16] J. Petrie. Guaranteeable Memory: An HBM-based chiplet for verifiable AI workloads. ICML Workshop on Technical AI Governance, 2025. <https://openreview.net/forum?id=uc79kOv0MV>.
- [17] N. Cankaya. Bit-exact AI inference verification without performance tradeoffs. arXiv:2606.00279, 2026.
- [18] A. Karvonen, D. Reuter, R. Rinberg, L. Marks, A. Garriga-Alonso, and K. Warr. DiFR: Inference verification despite nondeterminism. arXiv:2511.20621, 2025.
- [19] R. Rinberg, A. Karvonen, A. Hoover, D. Reuter, and K. Warr. Verifying LLM inference to detect model weight exfiltration. arXiv:2511.02620, 2025.

- [20] J. Petrie and Y. Mühlhäuser. Verifying AI compute by bounding unexplained information exfiltration. ICML Workshop on Technical AI Governance Research, 2026. <https://openreview.net/forum?id=qtgG5HZSsk>.
- [21] A. Trask, E. Bluemke, B. Garfinkel, C. G. Cuervas-Mons, and A. Dafoe. Beyond privacy trade-offs with structured transparency. arXiv:2012.08347, 2020.
- [22] T. South, A. Camuto, S. Jain, S. Nguyen, R. Mahari, C. Paquin, J. Morton, and A. Pentland. Verifiable evaluations of machine learning models using zkSNARKs. arXiv:2402.02675, 2024.
- [23] J. Groth. On the size of pairing-based non-interactive arguments. In *Advances in Cryptology—EUROCRYPT 2016*, volume 9666 of *Lecture Notes in Computer Science*, pages 305–326. Springer, 2016. DOI: 10.1007/978-3-662-49896-5_11.
- [24] iden3. Circom: A circuit compiler for zero-knowledge proving systems. <https://github.com/iden3/circom>.
- [25] iden3. snarkjs: A zk-SNARK implementation in JavaScript and WebAssembly. <https://github.com/iden3/snarkjs>.
- [26] P. Chantasantitam, A. I. Caulfield, V. Duddu, L. J. Gunn, and N. Asokan. PAL*M: Property attestation for large generative models. arXiv:2601.16199, 2026.
- [27] R. Rahman and S. Tajdari. Detecting hidden ML training with zero-overhead telemetry. arXiv:2606.19262, 2026.
- [28] S. K. Monfared, F. Ganji, D. E. Holcomb, and S. Tajik. Timing and memory telemetry on GPUs for AI governance. arXiv:2602.09369, 2026.
- [29] H. Xu, C. Gong, B. Liu, H. Zheng, B. Chen, and M. Li. WAVE: Leveraging architecture observation for privacy-preserving model oversight. In *Proceedings of the 31st ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 2212–2231, 2026. DOI: 10.1145/3779212.3790247.
- [30] G. Almusaddar, Y. Zhang, S. Ganjisaffar, B. Williams, Y. D. Liu, D. Ponomarev, and N. Abu-Ghazaleh. ShadowScope: GPU monitoring and validation via composable side channel signals. arXiv:2509.00300, 2025.
- [31] T. Gregersen, P. Patel, and E. Choukse. Input-dependent power usage in GPUs. arXiv:2409.18324, 2024.
- [32] C. Fang, H. Jia, A. Thudi, M. Yaghini, C. A. Choquette-Choo, N. Dullerud, V. Chandrasekaran, and N. Papernot. Proof-of-learning is currently more broken than you think. In *IEEE European Symposium on Security and Privacy*, pages 797–816, 2023.
- [33] P. Peigné, K. Nguyen, and P. Wang. Zero knowledge verification for frontier AI training is possible. arXiv:2606.05433, 2026.
- [34] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2nd edition, 2006.
- [35] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, 1999.
- [36] A. Makhdoumi, S. Salamatian, N. Fawaz, and M. Médard. From the information bottleneck to the privacy funnel. In *IEEE Information Theory Workshop*, 2014.
- [37] L. Sankar, S. R. Rajagopalan, and H. V. Poor. Utility–privacy tradeoffs in databases: An information-theoretic approach. *IEEE Transactions on Information Forensics and Security*, 8(6):838–852, 2013.
- [38] Y. Wang, Y. O. Basciftci, and P. Ishwar. Privacy–utility tradeoffs under constrained data release mechanisms. arXiv:1710.09295, 2017.
- [39] J. Liao, O. Kosut, L. Sankar, and F. P. Calmon. Tunable measures for information leakage and applications to privacy–utility tradeoffs. *IEEE Transactions on Information Theory*, 65(12):8043–8066, 2019.
- [40] W. Li, A. Halimi, J. Vaidya, X. Jiang, and E. Ayday. Privacy-preserving verification of ML preprocessing via model behavior indicators. *IEEE Transactions on Privacy*, 2:144–158, 2025. <https://doi.org/10.1109/TP.2025.3628998>.
- [41] S. Waiwitlikhit, I. Stoica, Y. Sun, T. Hashimoto, and D. Kang. Trustless audits without revealing data or models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 49808–49821, 2024.

- [42] H. Jia, M. Yaghini, C. A. Choquette-Choo, N. Dullerud, A. Thudi, V. Chandrasekaran, and N. Papernot. Proof-of-learning: Definitions and practice. In *IEEE Symposium on Security and Privacy*, pages 1039–1056, 2021. DOI: 10.1109/SP40001.2021.00106.
- [43] D. Choi, Y. Shavit, and D. K. Duvenaud. Tools for verifying neural models’ training data. In *Advances in Neural Information Processing Systems 36*, 2023.
- [44] C. Schnabl, D. Hugenroth, B. Marino, and A. R. Beresford. Attestable audits: Verifiable AI safety benchmarks using trusted execution environments. arXiv:2506.23706, 2025.
- [45] J. M. Ong, M. Di Ferrante, A. Pazdera, R. Garner, S. Jaghouar, M. Basra, M. Ryabinin, and J. Hagemann. TOPLOC: A locality-sensitive hashing scheme for trustless verifiable inference. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 47196–47211, 2025.
- [46] P. Horváth, D. Lauret, Z. Liu, and L. Batina. SoK: Neural network extraction through physical side channels. In *33rd USENIX Security Symposium*, 2024.
- [47] E. Debenedetti, G. Severi, N. Carlini, C. A. Choquette-Choo, M. Jagielski, M. Nasr, E. Wallace, and F. Tramèr. Privacy side channels in machine learning systems. In *33rd USENIX Security Symposium*, 2024.
- [48] R. Xiao, S. Feng, S. Ramesh, Jun Han, and Jinsong Han. Peering inside the black-box: Long-range and scalable model architecture snooping via GPU electromagnetic side-channel. In *NDSS Symposium*, 2026. DOI: 10.14722/ndss.2026.230141.
- [49] Y. Gao, H. Qiu, Z. Zhang, B. Wang, H. Ma, A. Abuadbbba, M. Xue, A. Fu, and S. Nepal. DeepTheft: Stealing DNN model architectures through power side channel. In *IEEE Symposium on Security and Privacy*, pages 3311–3326, 2024. DOI: 10.1109/SP54263.2024.00250.
- [50] H. T. Maia, C. Xiao, D. Li, E. Grinspun, and C. Zheng. Can one hear the shape of a neural network? Snooping the GPU via magnetic side channel. In *31st USENIX Security Symposium*, pages 4383–4400, 2022.
- [51] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, pages 3–18, 2017.
- [52] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE Computer Security Foundations Symposium*, pages 268–282, 2018.
- [53] M. Nasr, R. Shokri, and A. Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE Symposium on Security and Privacy*, pages 739–753, 2019.
- [54] L. Song and P. Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium*, pages 2615–2632, 2021.
- [55] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramèr. Membership inference attacks from first principles. In *IEEE Symposium on Security and Privacy*, 2022.
- [56] AI Futures Project. Get involved in verification. July 9, 2026. <https://ai-2040.com/supplements/verification-plan/get-involved>.
- [57] European Commission. General-purpose AI models in the AI Act: Questions and answers. Updated September 9, 2025. <https://digital-strategy.ec.europa.eu/en/faqs/general-purpose-ai-models-ai-act-questions-answers>.
- [58] Y. Bengio et al. International AI Safety Report 2026. UK Department for Science, Innovation and Technology, Research Series DSIT 2026/001, February 3, 2026. arXiv:2602.21012.
- [59] DeepSeek-AI et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. DOI: 10.1038/s41586-025-09422-z.
- [60] U.S. Department of Commerce, Bureau of Industry and Security. Policy statement on controls that may apply to advanced computing integrated circuits and other commodities used to train AI models. 2025. <https://www.bis.gov/media/1577>.

- [61] R. Vercellino, J. Willard, G. Campos, W. da Silva Pereira, O. Hull, M. Selensky, and J. Mueller. Dataset of generative AI workload power profiles. National Laboratory of the Rockies Data Catalog, 2026. DOI: 10.7799/3025227.
- [62] A. A. E. Elsayed, A. A. Al-Obaidi, and H. E. Z. Farag. Characterization of high-resolution AI data center training workloads on single and multiple GPU nodes. *Scientific Data*, published online June 3, 2026. DOI: 10.1038/s41597-026-07496-6.