On the Privacy-Preserving Capabilities of PAVE Specifications in Learnware*

Hao-Yi Lei[1,2], **Jin-Hui Wu**[3], **Zhi-Hao Tan**[1,2], **Zhi-Hua Zhou**[1,2]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, China
[2] School of Artificial Intelligence, Nanjing University, China
[3] School of Computer Science and Technology, Soochow University, China
{leihy, tanzh, zhouzh}@lamda.nju.edu.cn, wujinhui@suda.edu.cn

Abstract

The learnware paradigm supports model reuse by pairing each submitted model with a *specification*, a lightweight representation used by the learnware dock system to identify, match, and reuse models without accessing raw data. While specifications are essential for learnware identification, they are also data-dependent public artifacts and it is not clear whether they reveal private information. Recently, the *Parameter Vector* (PAVE) specification has been proposed and shown to be effective for learnwares, yet its privacy properties remain largely unexplored. In this paper, we provide the first theoretical privacy analysis for PAVE. Specifically, we first formalize two specification-induced risks in the learnware paradigm: the *disclosure risk* of the released specification and the *amplification risk* that the specification may strengthen attacks against the released model. Second, we characterize when compact PAVE releases admit intrinsic differential privacy (DP): under natural structural conditions of learnware docks, the compact PAVE specification satisfies an (ε, δ)-DP guarantee without explicit additive noise through a Gaussian-sketch view of stable parameter variations, and for regimes outside these conditions, we further provide DP-Stabilized-PAVE as a certified differentially private variant. Third, we show that the resulting DP guarantees control both disclosure risk and amplification risk, and we analyze the induced privacy–utility trade-off to guide effective learnware identification while preserving privacy.

1 Introduction

Reusing existing models is often more practical than training a new model from scratch. In many applications, developers already hold trained models that may be useful for future tasks, while users often have only limited data and limited computational resources. The difficulty is that model reuse is not only a matter of collecting models. A user must first identify which existing models are relevant to her task. Simple metadata, such as model names, architectures, tags, or benchmark scores, can only provide a rough description of model capability. Directly evaluating many candidate models on the user's raw data is more informative, but it is costly and may expose private task information. Asking developers to reveal their training data would also help model identification, but is usually infeasible due to privacy or proprietary concerns.

The learnware paradigm [Zhou, 2016; Zhou and Tan, 2024] addresses this problem by making the specification an explicit part of model reuse. A learnware is a model accompanied by a specification that describes its reusable capability. Such a specification can be generated from the model, the training data, or task information, and the dock system can use it to match developer-side models with user-side requirements without raw-data exchange. In this sense, the specification is the public artifact that makes learnware identification possible. However, this specification is also data-dependent. The more accurately it reflects model capability, the more likely it is to carry information about the data or task from which it was generated. Thus, the privacy question in learnware is not only about releasing models, but also about releasing specifications: what information does this public capability specification reveal, and how can such leakage be controlled while keeping the specification useful?

The parameter vector (PAVE) specification for learnwares has shown strong empirical effectiveness for learnware identification and reuse [Tan *et al.*, 2025; Shi *et al.*, 2026] recently. Unlike reduced-set specifications such as RKME specifications [Zhou and Tan, 2024], PAVE summarizes task and capability information through parameter variations induced by fine-tuning. The released specification is a compact low-rank parameter rather than a synthetic representative set in the data space. This difference makes existing privacy analyses for RKME insufficient for PAVE. On the one hand, gradient-induced parameter variations may themselves carry information about the underlying data. On the other hand, a PAVE specification is released together with a model, so it may also provide additional evidence for attacks against the released model. These observations motivate a privacy analysis tailored to the PAVE generation and release procedure.

Differential privacy (DP) [Dwork, 2006] is a natural tool for such an analysis, since it provides a robust worst-case

*A technical appendix with detailed proofs is available at https://default-anno-bucket.s3.us-west-1.amazonaws.com/Proof.pdf.

guarantee against inference attacks. However, applying DP to learnware specifications is non-trivial. Most DP mechanisms rely on explicit noise injection, while learnware specifications are intended to be compact and useful for identification. Excessive perturbation can therefore damage the matching utility of the learnware dock system, as also observed in prior studies on learnware privacy [Lei *et al.*, 2024]. Moreover, a specification-level privacy guarantee should be connected to learnware-specific risks: releasing a specification may leak information by itself, and may also strengthen attacks when combined with the released model.

In this work, we characterize the privacy-preserving capabilities of PAVE specifications. The main contributions are summarized as follows:

- We formalize two specification-induced privacy risks for PAVE, namely the disclosure risk of the released specification and the amplification risk that the specification may strengthen attacks against the released model.
- We show that compact PAVE releases admit an intrinsic (ε, δ)-DP guarantee under natural structural conditions of learnware docks, through a Gaussian-sketch view of stable parameter variations. For regimes outside these conditions, we further provide DP-Stabilized-PAVE as a certified differentially private variant.
- We prove that the resulting DP guarantees can control both disclosure risk and specification-side amplification risk, and analyze the induced privacy–utility trade-off to guide privacy-aware deployment while preserving effective learnware identification.

2 Preliminaries and Methodology

In this section, we introduce the basic workflow of the learnware paradigm with PAVE specifications, and formalize the privacy concerns that arise from releasing such specifications.

2.1 Background of Learnware Paradigm

We first recall the two-stage learnware workflow.

Submitting stage. For a developer, a learnware consists of a trained model h together with its specification. PAVE constructs a *model vector* τ_h by fine-tuning a shared pre-trained model $f(\cdot;\theta_0)$ on the developer's training dataset D_t with a task-specific loss L_t. When the output space of the pre-trained model differs from that of the task, a lightweight mapping g_t is used to bridge the two spaces. Concretely, PAVE is obtained as the parameter update that minimizes

$$\tau_h \;=\; \arg\min_{\tau} \sum_{(x,y)\in D_t} L_t\big(g_t \circ f(x;\theta_0+\tau),\, h(x)\big). \quad (1)$$

The learnware (h, τ_h) is then submitted to the dock system.

Deploying stage. Given a user task with a few-shot dataset D_u and loss L_u, PAVE constructs a *task vector* τ_u using the same pre-trained model $f(\cdot;\theta_0)$, but replaces the model prediction target by the ground-truth label. Thus the vector reflects the capability required by the user task:

$$\tau_u \;=\; \arg\min_{\tau} \sum_{(x,y)\in D_u} L_u\big(g_u \circ f(x;\theta_0+\tau),\, y\big). \quad (2)$$

The dock system identifies helpful learnwares by comparing cosine similarity in the parameter-vector space:

$$\cos(\tau_h, \tau_u) \;=\; \frac{\langle \tau_h, \tau_u \rangle}{\|\tau_h\|_2 \, \|\tau_u\|_2}, \quad (3)$$

and selects learnwares with large similarity for reuse.

To make PAVE efficient to store and compare, it adopts a LoRA-style low-rank parameterization. For each selected weight matrix $\mathbf{W}_\ell$ in the pre-trained backbone, the update is represented as

$$\Delta \mathbf{W}_\ell = \mathbf{B}_\ell \mathbf{A}_\ell, \quad (4)$$

where $\mathbf{A}_\ell \in \mathbb{R}^{r\times n_\ell}$ is randomly initialized and then frozen, while $\mathbf{B}_\ell \in \mathbb{R}^{m_\ell \times r}$ with $r \ll \min\{m_\ell, n_\ell\}$ is learned from data. Writing $\mathbf{A}$ for the collection of frozen factors, PAVE learns the compact factors by

$$\mathbf{B}^\star = \arg\min_{\mathbf{B}} \sum_{(x,y)\in D} \mathcal{L}\big(g \circ f(x, \theta_0 + \mathbf{B}\mathbf{A}),\, h(x)\big). \quad (5)$$

The expanded update $\tilde{\tau} = \mathbf{B}\mathbf{A}$ is the corresponding full-space parameter variation, while the concatenated matrix

$$\mathbf{B} = [\mathbf{B}_1, \ldots, \mathbf{B}_L] \in \mathbb{R}^{m\times(rL)} \quad (6)$$

is used as the compact PAVE specification. For identification, the learnware dock system can directly compute cosine similarity in the low-rank space via $\cos(\mathbf{B}, \mathbf{B}')$, which efficiently approximates $\cos(\mathbf{B}\mathbf{A}, \mathbf{B}'\mathbf{A})$ with high probability. In the compact-release regime analyzed in this paper, the public specification is the compact factor $\mathbf{B}$; the realized random factors $\mathbf{A}_\ell$ are not part of the public specification.

2.2 Formalization of Learnware Privacy Concerns

Since specifications are generated from the raw data of developers or users, a central privacy question is: *what additional risks are introduced by attaching and releasing a specification?* We distinguish two complementary concerns below.

(i) Disclosure risk. The released specification itself may leak information about the underlying dataset. This concern is particularly relevant for PAVE, because it summarizes the fine-tuning dataset through gradient-induced parameter variations. Prior studies on gradient and update inversion show that such information can be used to infer sensitive properties of training examples [Zhu *et al.*, 2019; Geiping *et al.*, 2020].

(ii) Amplification risk. Even without a specification, the released model may already leak information about its training data through model inversion attacks, which attempt to recover sensitive attributes or representative training examples from model outputs or parameters [Fredrikson *et al.*, 2014; Fredrikson *et al.*, 2015; Zhang *et al.*, 2020]. In a learnware system, publishing a specification alongside the model introduces an additional evidence channel. The learnware-specific risk we study is whether this extra channel can strengthen such attacks beyond what the model alone already enables.

Figure 1 summarizes the PAVE identification workflow and the two specification-induced risk studied in this paper.

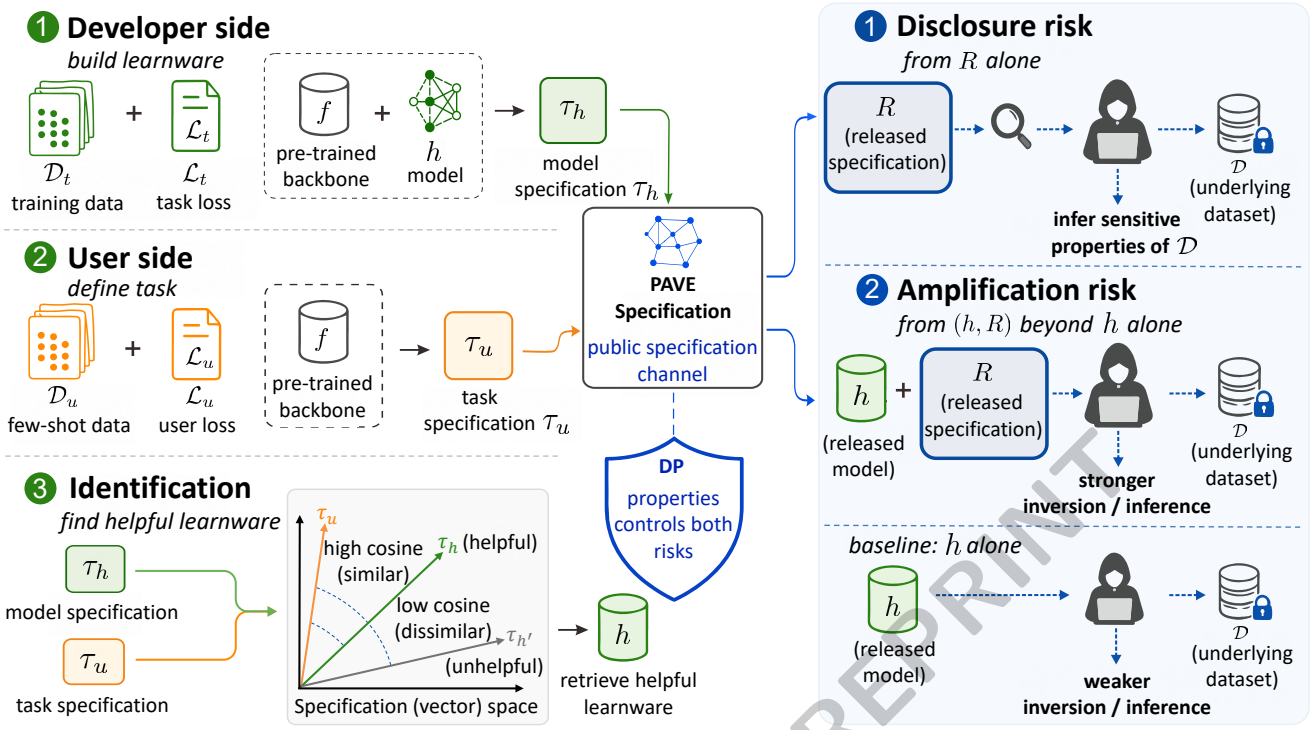

Figure 1: Overview of PAVE-based learnware identification, reuse and specification-induced privacy risks.

A generic inference game. Let D denote the developer's training dataset, drawn from an underlying distribution $\mathcal{P}$. A learnware consists of a released model h together with a released specification R. We model an attack by a tuple $(\mathcal{A}, \tau, \nu)$, where $\mathcal{A}$ is a possibly randomized adversary, τ is a target function encoding the inference objective, and ν denotes the side information provided to the adversary. The interaction proceeds as follows:

(1) The challenger samples a record s from $\mathcal{P}$, or equivalently from D when modeling membership.

(2) The challenger reveals the side information $\nu(s)$ to the adversary.

(3) The adversary is given a released interface, specified below, and outputs a guess.

The adversary's success is defined as

$$\mathrm{gain}(\mathcal{A}; \mathcal{V}, \tau, \nu) \triangleq \Pr\left[\mathcal{A}(\mathcal{V}, \nu(s)) = \tau(s)\right], \tag{7}$$

where the probability is over the randomness of the sampled record, the released interface, and the adversary. Different privacy concerns correspond to different adversarial views $\mathcal{V}$.

(i) Disclosure risk of releasing the specification. We first quantify what the specification R reveals by itself. In this game, the adversary observes the released specification, denoted by $\mathcal{V}_{\mathrm{spec}} = R$. Since side information may already reveal part of the target, we define disclosure risk as the incremental advantage brought by releasing R:

$$\mathrm{Risk}_{\mathrm{dis}}(R) \triangleq \sup_{\mathcal{A}\in\mathfrak{A}} \mathrm{gain}(\mathcal{A}; R, \tau, \nu) - \sup_{\mathcal{A}\in\mathfrak{A}} \mathrm{gain}(\mathcal{A}; \varnothing, \tau, \nu), \tag{8}$$

where $\mathfrak{A}$ denotes the class of admissible adversaries, and $\varnothing$ denotes the baseline view in which the adversary receives only the side information. A small $\mathrm{Risk}_{\mathrm{dis}}(R)$ indicates that observing the released specification does not substantially improve the adversary's ability to infer $\tau(s)$ beyond what is already possible from side information.

(ii) Amplification risk for attacks against the released model. Releasing a specification may also help an attacker exploit the *model* more effectively. To capture this, we compare adversaries that observe the model alone with adversaries that observe both the model and the specification. Let $\mathcal{V}_{\mathrm{model}} = h$ denote access to the released model alone, and let $\mathcal{V}_{\mathrm{joint}} = (h, R)$ denote access to both. We define the *amplification risk* induced by R for the model h as

$$\mathrm{Risk}_{\mathrm{amp}}(h; R) \triangleq \sup_{\mathcal{A}\in\mathfrak{A}} \mathrm{gain}(\mathcal{A}; (h, R), \tau, \nu) - \sup_{\mathcal{A}\in\mathfrak{A}} \mathrm{gain}(\mathcal{A}; h, \tau, \nu). \tag{9}$$

This quantity isolates the incremental privacy risk introduced by publishing the specification on top of the released model.

Binary instantiation. The gain-based definitions above describe the operational meaning of the two risks for general inference objectives. To obtain attack-agnostic theoretical guarantees, we use their standard binary neighbouring-dataset instantiation. In this instantiation, the adversary distinguishes whether a view is generated from a dataset D or from a neighbouring dataset D'.

For a released view V, define

$$p^{V}_{\mathcal{A}}(D) \triangleq \Pr\left[\mathcal{A}(V(D)) = 1\right], \tag{10}$$

and the optimal distinguishing advantage

$$\mathrm{Adv}^{\star}(V; D, D') \triangleq \sup_{\mathcal{A}\in\mathfrak{A}} \left| p^{V}_{\mathcal{A}}(D) - p^{V}_{\mathcal{A}}(D') \right|. \tag{11}$$

Equivalently, in the balanced binary game between D and D', the optimal success probability is

$$\mathrm{gain}^{\star}_{\mathrm{bin}}(V; D, D') = \frac{1}{2} + \frac{1}{2}\,\mathrm{Adv}^{\star}(V; D, D'). \tag{12}$$

Thus, controlling $\mathrm{Adv}^{\star}$ also controls the corresponding binary inference gain. Let $[x]_+ \triangleq \max\{x, 0\}$. The advantage-style disclosure risk $\mathrm{Risk}^{\mathrm{adv}}_{\mathrm{dis}}(R \mid H_0)$ is

$$\sup_{D\sim D'} \Big[\mathrm{Adv}^{\star}((H_0, R); D, D') - \mathrm{Adv}^{\star}(H_0; D, D') \Big]_+, \tag{13}$$

where H_0 denotes the baseline side information before observing the specification. Similarly, the advantage-style amplification risk $\mathrm{Risk}^{\mathrm{adv}}_{\mathrm{amp}}(H; R)$ is

$$\sup_{D\sim D'} \Big[\mathrm{Adv}^{\star}((H, R); D, D') - \mathrm{Adv}^{\star}(H; D, D') \Big]_+, \tag{14}$$

where H denotes the model-only view.

3 DP Guarantees for PAVE Specifications

In this section, we characterize when PAVE specifications admit differential privacy. We first show that, under natural compact-release conditions of learnware docks, the compact PAVE factor satisfies an intrinsic (ε, δ)-DP guarantee without injecting additional additive noise. We then discuss the structural conditions behind this guarantee and introduce DP-S-PAVE, a certified variant for deployments outside the intrinsic compact-release regime.

We first recall some basic notions of differential privacy (DP). DP limits how much the output distribution can change when a member of the original dataset is modified.

Definition 3.1 (Neighbouring datasets)**.** *Let $D = \{z_i\}_{i=1}^N$ and $D' = \{z'_i\}_{i=1}^N$ be two datasets of the same size. We write $D \sim D'$ if they differ in exactly one record, i.e., there exists an index j such that $z_j \neq z'_j$ and $z_i = z'_i$ for all $i \neq j$.*

Definition 3.2 (Differential privacy)**.** *A randomized mechanism $\mathcal{M}$ is (ε, δ)-differentially private if for any measurable event E and any neighbouring datasets $D \sim D'$,*

$$\Pr\big(\mathcal{M}(D) \in E\big) \le e^{\varepsilon} \Pr\big(\mathcal{M}(D') \in E\big) + \delta, \tag{15}$$

where the probability is taken over the randomness of $\mathcal{M}$.

Definition 3.3 (ℓ_2-sensitivity)**.** *Let q be a matrix-valued query. Its ℓ_2-sensitivity is*

$$\Delta(q) \triangleq \max_{D\sim D'} \big\| q(D) - q(D') \big\|_2, \tag{16}$$

where $\|\cdot\|_2$ denotes the spectral norm.

From PAVE to a compact Gaussian-sketch mechanism. For clarity, we first consider a single layer and omit the layer index. Let $\mathbf{A} \in \mathbb{R}^{r\times m}$ be sampled with i.i.d. entries $A_{ij} \sim \mathcal{N}(0, 1/r)$, independently of the dataset. In the compact privacy regime, the distribution of $\mathbf{A}$ is public, while its realized value and random seed are maintained as internal randomness and are not part of the public specification.

For one-step, linearized, or trajectory-decoupled PAVE, the learned compact factor can be written as

$$\mathbf{B}(D) = \mathbf{F}(D)\mathbf{A}^{\top}, \tag{17}$$

where $\mathbf{F}(D) \in \mathbb{R}^{d\times m}$ is the matrix query induced by the fine-tuning trajectory. For example, unrolling gradient steps of the form $\mathbf{B}_{t+1} = \mathbf{B}_t - \eta\,\mathbf{G}_t(D)\mathbf{A}^{\top}$ gives $\mathbf{B}_T(D) = -\eta \sum_{t=0}^{T-1} \mathbf{G}_t(D)\mathbf{A}^{\top}$, so that $\mathbf{F}(D) = -\eta \sum_{t=0}^{T-1} \mathbf{G}_t(D)$ when the cumulative query is independent of the realized $\mathbf{A}$.

The mechanism in Eq. (17) is a Gaussian sketch. To avoid support separation for matrix-valued releases, we analyze the standard fixed-subspace regime. Assume that there exists a fixed and public matrix $\mathbf{U} \in \mathbb{R}^{d\times k}$, $\mathbf{U}^{\top}\mathbf{U} = \mathbf{I}_k$, such that the released query lies in this specification subspace:

$$\mathbf{F}(D) = \mathbf{U}\mathbf{H}(D) \quad \text{with} \quad \mathbf{H}(D) \in \mathbb{R}^{k\times m}. \tag{18}$$

Equivalently, the released compact factor can be written as

$$\mathbf{B}(D) = \mathbf{U}\mathbf{C}(D) \quad \text{with} \quad \mathbf{C}(D) = \mathbf{H}(D)\mathbf{A}^{\top}. \tag{19}$$

Since $\mathbf{U}$ is fixed and public, it suffices to analyze the coordinate release $\mathbf{C}(D)$. We define the covariance matrix

$$\mathbf{\Sigma}_D \triangleq \mathbf{H}(D)\mathbf{H}(D)^{\top} \in \mathbb{R}^{k\times k}. \tag{20}$$

We impose the following covariance-stability condition on admissible neighbouring datasets:

$$(1-\gamma)\mathbf{\Sigma}_{D'} \preceq \mathbf{\Sigma}_D \preceq (1+\gamma)\mathbf{\Sigma}_{D'} \quad \text{with} \quad 0 < \gamma < 1, \tag{21}$$

where $\mathbf{\Sigma}_D$ and $\mathbf{\Sigma}_{D'}$ are positive definite. This condition states that changing one record cannot substantially rotate or rescale the covariance inside the specification subspace.

For a concise privacy accounting, we use Rényi differential privacy (RDP).

Definition 3.4 (Rényi differential privacy)**.** *For $\alpha > 1$, a mechanism $\mathcal{M}$ is said to satisfy (α, ρ)-RDP if*

$$D_{\alpha}(\mathcal{M}(D) \| \mathcal{M}(D')) \le \rho \tag{22}$$

for all neighbouring datasets $D \sim D'$, where $D_{\alpha}(\cdot\|\cdot)$ is the order-α Rényi divergence.

An (α, ρ)-RDP mechanism is also $\left(\rho + \frac{\log(1/\delta)}{\alpha-1},\, \delta\right)$-DP for any $\delta \in (0, 1)$. In particular, a $(2, \rho_2)$-RDP guarantee gives $(\rho_2 + \log(1/\delta), \delta)$-DP.

Theorem 3.5 (Intrinsic DP for compact PAVE specifications)**.** *Consider the compact PAVE mechanism*

$$\mathcal{M}_{\mathbf{B}}(D) = \mathbf{B}(D) = \mathbf{U}\mathbf{H}(D)\mathbf{A}^{\top}, \tag{23}$$

where $A_{ij} \sim \mathcal{N}(0, 1/r)$ is sampled independently of D, and the realized $\mathbf{A}$ is not released. Assume the common-support condition in Eq. (18) *and the covariance-stability condition in Eq.* (21)*. Then $\mathcal{M}_{\mathbf{B}}$ satisfies $(2, \rho_2)$-RDP with*

$$\rho_2 \le \frac{rk}{2} \log \frac{1}{1-\gamma^2} \le \frac{rk\gamma^2}{2(1-\gamma^2)}. \tag{24}$$

Consequently, for any $\delta \in (0, 1)$, $\mathcal{M}_{\mathbf{B}}$ is (ε, δ)-DP with

$$\varepsilon = \frac{rk}{2} \log \frac{1}{1-\gamma^2} + \log \frac{1}{\delta} \le \frac{rk\gamma^2}{2(1-\gamma^2)} + \log \frac{1}{\delta}. \tag{25}$$

Remark 3.6. *Theorem 3.5 characterizes an intrinsic DP regime of compact PAVE specifications. It relies on three structural conditions and one modeling scope condition.*

Compact secret release. *The public PAVE specification is the compact factor* $\mathbf{B} = \mathbf{F}(D)\mathbf{A}^\top$*, while the realized Gaussian matrix* $\mathbf{A}$*, its seed, and the expanded update* $\mathbf{BA}$ *are not part of the public release. The distribution of* $\mathbf{A}$ *can be public; what is not released is the particular realization used to generate the specification. This distinction is essential because* $\mathbf{A}$ *serves as the internal randomness of the compact Gaussian-sketch mechanism. If* $(\mathbf{A}, \mathbf{B})$ *is released, the algebraic relation* $\mathbf{B} = \mathbf{F}(D)\mathbf{A}^\top$ *becomes directly testable, and the Gaussian-sketch privacy argument no longer applies. The modeling scope condition is related but different: the matrix query* $\mathbf{F}(D)$ *should be independent of the realized* $\mathbf{A}$*. This holds for one-step or trajectory-decoupled PAVE, where the cumulative query is fixed once* D *is fixed. For fully adaptive PAVE training, the certified variant below should be used.*

Common specification subspace. *The second structural condition is the fixed public subspace in Eq.* (18)*. Its role is to place all released PAVE specifications in the same public coordinate system:* $\mathbf{F}(D) = \mathbf{U}\mathbf{H}(D)$ *and* $\mathbf{B}(D) = \mathbf{U}(\mathbf{H}(D)\mathbf{A}^\top)$*. This condition is natural in a learnware dock, because model-side and user-side specifications must be comparable for identification. It is automatically satisfied when the selected PAVE query has stable full row support, in which case one may take* $\mathbf{U} = \mathbf{I}$*. More generally, the system can enforce it by selecting fixed layers, using a fixed public projection subspace, choosing a subspace from public calibration data, or directly releasing the coordinate specification* $\mathbf{C}(D) = \mathbf{U}^\top\mathbf{B}(D) = \mathbf{H}(D)\mathbf{A}^\top$*. Without such a common support, neighbouring datasets may move the released matrix into different column supports, causing support separation.*

Covariance stability. *The third structural condition is covariance stability in Eq.* (21)*. It requires the Gaussian-sketch covariance* $\mathbf{\Sigma}_D = \mathbf{H}(D)\mathbf{H}(D)^\top$ *to change smoothly under a single-record modification. The positive definiteness of* $\mathbf{\Sigma}_D$ *is the non-degeneracy part of this condition: the retained public specification subspace should not collapse to a lower-dimensional direction. A convenient sufficient condition is* $\mu^2\mathbf{I}_k \preceq \mathbf{H}(D)\mathbf{H}(D)^\top \preceq L^2\mathbf{I}_k$ *and* $\|\mathbf{H}(D) - \mathbf{H}(D')\|_2 \le \Delta$ *for all neighbouring* $D \sim D'$*. Then* $\|\mathbf{\Sigma}_D - \mathbf{\Sigma}_{D'}\|_2 \le 2L\Delta$*, and Eq.* (21) *holds with*

$$\gamma = \frac{2L\Delta}{\mu^2}, \tag{26}$$

provided $\gamma < 1$*. For an average-type query* $\mathbf{H}(D) = n^{-1}\sum_{i=1}^n \mathbf{\Psi}(z_i)$ *with* $\|\mathbf{\Psi}(z)\|_2 \le G$*, one has* $\Delta \le 2G/n$*, and hence* $\gamma \le 4LG/(n\mu^2)$*. Thus large fine-tuning datasets, stable optimization, and clipping of unusually large updates all make the intrinsic compact guarantee more favorable.*

Proposition 3.7 (A public-$\mathbf{A}$ compact release is not generally DP)**.** *Consider the mechanism*

$$\mathcal{M}_{\mathrm{pub}}(D) = (\mathbf{A},\, \mathbf{F}(D)\mathbf{A}^\top), \tag{27}$$

where $\mathbf{A} \in \mathbb{R}^{r\times m}$ *has i.i.d. Gaussian entries and the realized* $\mathbf{A}$ *is released. If there exist neighbouring datasets* $D \sim D'$ *such that* $\mathbf{F}(D) - \mathbf{F}(D') \neq \mathbf{0}$*, then, for such a neighbouring pair,* $\mathcal{M}_{\mathrm{pub}}$ *is not* (ε, δ)*-DP for any finite* ε *and any* $\delta < 1$*.*

Remark 3.8. *The quantities in Theorem 3.5 have direct learnware interpretations. The rank* r *is the low-rank width of the PAVE factor, and* k *is the dimension of the public specification subspace. Their product* rk *is the effective number of Gaussian sketch coordinates released to the system. The parameter* γ *measures the relative change of the specification covariance under a neighbouring-dataset modification. Thus, smaller effective specification dimension and more stable task-level parameter variations lead to tighter privacy. These quantities also correspond to design choices of the system: the rank controls how much parameter-variation signal is retained, the public subspace controls the common coordinate system for matching, and the stability parameter reflects the influence of individual records on the specification.*

Algorithm 1 DP-Stabilized PAVE (DP-S-PAVE)

Input: dataset D; pretrained parameters θ_0; selected layers set $\mathcal{I}$ with $\mathbf{W}_\ell \in \mathbb{R}^{m_\ell\times n_\ell}$; ranks $\{r_\ell\}_{\ell\in\mathcal{I}}$; per-example fine-tuning loss $\mathcal{L}$; steps T; step size η; clipping threshold C; sampling rate q; noise multiplier σ.
Output: private compact specification $R = \{\mathbf{B}_{\ell,T}\}_{\ell\in\mathcal{I}}$.
for $\ell \in \mathcal{I}$ **do**
  Initialize $\mathbf{A}_\ell \in \mathbb{R}^{r_\ell\times n_\ell}$ independently of D and freeze it; and set $\mathbf{B}_{\ell,0} \leftarrow \mathbf{0}_{m_\ell\times r_\ell}$.
end for
for $t = 0, \dots, T-1$ **do**
  Sample a mini-batch $S_t \subseteq D$ with rate q or $|S_t| = \lfloor q|D| \rfloor$.
  for $\ell \in \mathcal{I}$ **do**
    for $z_i \in S_t$ **do**
      $\mathbf{Z}_{\ell,t}^{(i)} \leftarrow \nabla_{\mathbf{W}_\ell}\mathcal{L}(\mathbf{W}_t; z_i)\mathbf{A}_\ell^\top$;
      $\widetilde{\mathbf{Z}}_{\ell,t}^{(i)} \leftarrow \mathrm{Clip}(\mathbf{Z}_{\ell,t}^{(i)}, C) = \mathbf{Z}\cdot\min\{1, C/\|\mathbf{Z}\|_{\mathrm{F}}\}$.
    end for
    Draw $\mathbf{N}_{\ell,t} \in \mathbb{R}^{m_\ell\times r_\ell}$ with entries from $\mathcal{N}(0, \sigma^2C^2)$.
    $\bar{\mathbf{Z}}_{\ell,t} \leftarrow b^{-1}\Big(\sum_{z_i\in S_t}\widetilde{\mathbf{Z}}_{\ell,t}^{(i)} + \mathbf{N}_{\ell,t}\Big)$.
    $\mathbf{B}_{\ell,t+1} \leftarrow \mathbf{B}_{\ell,t} - \eta\bar{\mathbf{Z}}_{\ell,t}$.
  end for
end for
return $R = \{\mathbf{B}_{\ell,T}\}_{\ell\in\mathcal{L}}$.

A certified variant beyond the intrinsic regime. The above structural conditions are natural in a compact-release regime, but they need not hold in every deployment. For example, a system may require public reproducibility of $\mathbf{A}$, may need to release the expanded update $\mathbf{BA}$, may not enforce a common specification subspace, or may use fully adaptive nonlinear PAVE training in which the cumulative query depends on the realized $\mathbf{A}$. To cover such cases, we introduce a certified differentially private variant, called DP-Stabilized PAVE (DP-S-PAVE). The idea is to apply per-example clipping and calibrated Gaussian noise directly in the low-rank PAVE update space. We use the same per-example loss as in the original PAVE generation procedure. In contrast to Theorem 3.5, DP-S-PAVE does not rely on secret $\mathbf{A}$, common support, or covariance stability.

Theorem 3.9 (Certified DP of DP-S-PAVE)**.** *Fix the matrices* $\{\mathbf{A}_\ell\}_{\ell\in\mathcal{L}}$ *independently of the dataset. In Algorithm 1, assume that every per-example projected update is clipped to Frobenius norm at most* C*, and that independent Gaussian noise with standard deviation* σC *is added before the deterministic scaling by* $1/b$*. Let* $\rho_\alpha^{\mathrm{subG}}(q, \sigma)$ *denote the order-*α *RDP parameter of one Poisson-subsampled Gaussian update*

with replacement-neighbour sensitivity $2C$ and noise standard deviation σC. Then the final release $R = \{\mathbf{B}_{\ell,T}\}_{\ell\in\mathcal{L}}$ satisfies (α, ρ_α)-RDP with

$$\rho_\alpha \le \sum_{t=0}^{T-1} \sum_{\ell\in\mathcal{L}} \rho_\alpha^{\mathrm{subG}}(q, \sigma). \tag{28}$$

Thus for any $\delta \in (0, 1)$, DP-S-PAVE is $(\varepsilon_{\mathrm{dp}}, \delta)$-DP with

$$\varepsilon_{\mathrm{dp}}(\delta) = \inf_{\alpha>1} \left\{ \rho_\alpha + \frac{\log(1/\delta)}{\alpha - 1} \right\}. \tag{29}$$

In particular, without subsampling, i.e., $q = 1$, a conservative closed-form bound is

$$\rho_\alpha \le \frac{2\alpha T|\mathcal{L}|}{\sigma^2}, \qquad \varepsilon_{\mathrm{dp}} \le \frac{2T|\mathcal{L}|}{\sigma^2} + 2\sqrt{\frac{2T|\mathcal{L}|}{\sigma^2} \log \frac{1}{\delta}}. \tag{30}$$

Remark 3.10. *DP-S-PAVE complements the intrinsic compact guarantee. When the compact secret-sketch conditions hold, Theorem 3.5 gives a noise-free DP route. When these conditions are not enforced, Theorem 3.9 provides a certified route by adding calibrated noise in the low-rank PAVE update space. Since the guarantee of DP-S-PAVE is conditioned on fixed $\mathbf{A}_\ell$ and comes from explicit Gaussian perturbation, the realized $\mathbf{A}_\ell$, the compact factors $\mathbf{B}_\ell$, the expanded updates $\mathbf{B}_\ell\mathbf{A}_\ell$, normalized specifications, cosine similarities, and top-K retrieval outputs are all post-processing of a DP mechanism and therefore remain differentially private.*

4 Risk Analysis

We now connect the DP guarantees in Section 3 to the two learnware privacy risks formalized in Section 2.2, and then discuss the resulting privacy–utility trade-off.

4.1 Disclosure Risk of the Specification Channel

We first consider the disclosure risk of releasing the specification itself. For brevity, we write $\eta_{\varepsilon,\delta} \triangleq (e^\varepsilon - 1) + \delta$.

Lemma 4.1 (DP channels add bounded inference advantage)**.** *Let H be a baseline view, and let $R_D = \mathcal{M}(D)$ be an additional released channel. Suppose that R is conditionally (ε, δ)-DP given H, i.e., for every fixed value $H = u$, every measurable event S, and all $D \sim D'$,*

$$\Pr[R_D \in S \mid H = u] \le e^\varepsilon \Pr[R_{D'} \in S \mid H = u] + \delta.$$

Then, for all neighbouring datasets $D \sim D'$,

$$\mathrm{Adv}^\star((H, R); D, D') \le \mathrm{Adv}^\star(H; D, D') + \eta_{\varepsilon,\delta}. \tag{31}$$

Lemma 4.1 is the link between mechanism-level privacy and learnware-level risk. After the view $H = u$ is fixed, any attack using (u, R) is a post-processing of the released specification. The term $\mathrm{Adv}^\star(H; D, D')$ accounts for the distinguishing power already present in the baseline view, while the additional contribution of R is controlled by $\eta_{\varepsilon,\delta}$.

Theorem 4.2 (DP controls specification disclosure risk)**.** *Let $R = \mathcal{M}_{\mathbf{B}}(D)$ be the released compact PAVE specification. If R is conditionally (ε, δ)-DP given the baseline side information H_0, then*

$$\mathrm{Risk}_{\mathrm{dis}}^{\mathrm{adv}}(R \mid H_0) \le (e^\varepsilon - 1) + \delta. \tag{32}$$

In particular, the same bound holds when H_0 is fixed public side information and R satisfies ordinary (ε, δ)-DP.

Theorem 4.2 is an incremental-risk statement. It does not bound the raw success probability of an attack, because the baseline side information may already reveal part of the target. Instead, it bounds the extra binary distinguishing advantage attributable to observing the specification. Consequently, tighter compact-release privacy parameters, smaller effective dimension rk, stronger covariance stability, or the use of DP-S-PAVE directly reduce the worst-case disclosure advantage.

4.2 Specification-Side Amplification Risk

We next consider amplification risk: whether publishing a PAVE specification can strengthen attacks against the released model. Let H denote the model-only view available to the adversary, including the released model h and any side information shared across the two games. The joint view is (H, R). The model view H may already contain information about the underlying dataset; our goal is to isolate the additional contribution of the specification channel.

We assume that the compact specification channel R is conditionally (ε, δ)-DP given H. This is satisfied, for example, when H is treated as fixed public side information and the compact PAVE release satisfies Theorem 3.5 uniformly under this conditioning. If the model h is trained by a non-private procedure, it may still leak information, but the result below controls the incremental leakage caused by adding R.

Theorem 4.3 (No material specification-side amplification)**.** *Let $R = \mathcal{M}_{\mathbf{B}}(D)$ be the released compact PAVE specification, and let H be the model-only view. If R is conditionally (ε, δ)-DP given H, then*

$$\mathrm{Risk}_{\mathrm{amp}}^{\mathrm{adv}}(H; R) \le (e^\varepsilon - 1) + \delta. \tag{33}$$

Although Theorem 4.3 has the same numerical margin as the disclosure bound, it controls a different learnware-specific risk. The disclosure bound compares the baseline side information H_0 with the joint view (H_0, R), and therefore measures what the specification reveals by itself. In contrast, the amplification bound takes the model-only view H as the baseline. This view may already be data-dependent and non-private, and the theorem does not attempt to remove such model-side leakage. Instead, it asks whether the additional specification channel R can further increase the adversary's distinguishing power beyond what is already possible from H. Technically, both results rely on Lemma 4.1, but they instantiate it with different baselines: H_0 for disclosure and H for amplification. The identical margin $(e^\varepsilon - 1) + \delta$ reflects that, in both cases, the only newly added channel is the DP-protected specification. Thus Theorem 4.3 should be read as a specification-side amplification guarantee: publishing PAVE may be combined with model access, but the extra inference advantage attributable to the specification is bounded by the DP-controlled margin in Eq. (33).

This is a specification-side guarantee. It separates the information already present in the released model view from the additional risk introduced by the specification. If a deployment releases objects outside the compact secret-release regime, such as the realized $\mathbf{A}$, the expanded update $\mathbf{BA}$, or an adapted predictor constructed from a public $\mathbf{A}$, then Theorem 3.5 may no longer apply. In such cases, the same disclosure and amplification conclusions can be obtained by using

the certified DP-S-PAVE guarantee in Theorem 3.9. Once the specification channel is DP, downstream dock operations such as normalization, cosine similarity, ranking, and top-K retrieval remain protected by post-processing.

4.3 Privacy–Utility Trade-off of PAVE

The preceding results show that both disclosure risk and specification-side amplification risk are controlled by the DP parameters of the released specification channel. Theorem 3.5 makes these parameters explicit for compact PAVE. In the intrinsic regime, the order-2 RDP parameter satisfies

$$\rho_2 \leq \frac{rk}{2} \log \frac{1}{1-\gamma^2}. \tag{34}$$

Thus the effective released dimension rk and the covariance-stability parameter γ are the main privacy-relevant quantities. The same quantities also affect learnware identification. The rank r controls the width of the low-rank PAVE factor, while k controls the dimension of the public specification subspace. Larger r or k can preserve richer parameter-variation signals and improve matching, but they also release more Gaussian sketch coordinates and increase the privacy cost. Conversely, overly small r or k may improve privacy but remove directions needed to distinguish helpful learnwares.

The stability parameter γ captures another side of the trade-off. It measures how much the covariance $\mathbf{\Sigma}_D = \mathbf{H}(D)\mathbf{H}(D)^\top$ of the projected PAVE query changes when one record is modified. When $\mathbf{H}(D)$ is stable and non-degenerate, γ is small and the privacy bound becomes tighter. The sufficient condition in Section 3 gives a concrete interpretation: if $\mu^2\mathbf{I}_k \preceq \mathbf{H}(D)\mathbf{H}(D)^\top \preceq L^2\mathbf{I}_k$ and $\|\mathbf{H}(D) - \mathbf{H}(D')\|_2 \leq \Delta$, then $\gamma = 2L\Delta/\mu^2$; for average-type queries with bounded per-example contribution, $\Delta = O(1/n)$. Larger fine-tuning datasets, stable optimization, and clipping of unusually large updates therefore reduce the effect of any individual record on the specification.

This stability is also useful for identification. Learnware matching should depend on task-level variation rather than example-specific spikes. Suppressing unstable record-level influence helps the PAVE specification reflect reusable capability information instead of idiosyncratic noise, which can improve the robustness of cosine-based comparisons across learnwares. In this sense, privacy and utility are not always opposed: the same stabilization that reduces γ can also make the specification more faithful to the task-level signal.

The public specification subspace also has a dual role. From the privacy perspective, a fixed subspace $\mathbf{U}$ prevents support separation across neighbouring datasets and enables the compact Gaussian-sketch analysis. From the learnware perspective, $\mathbf{U}$ provides a common coordinate system in which model-side and user-side specifications can be compared. In practice, $\mathbf{U}$ may be chosen by fixed layer selection or a public projection, while the effective dimension k can be tuned according to the desired privacy–utility balance. When the compact secret-release conditions are not suitable, the privacy of DP-S-PAVE privacy is controlled by the clipping threshold, noise multiplier, sampling rate, and number of optimization steps through standard DP accounting.

5 Related Work

Most existing learnware systems [Tan *et al.*, 2024c] build with RKME as a specification choice [Zhou and Tan, 2024; Wu *et al.*, 2023]. This has led to a rich line of work on learnware identification, heterogeneous feature or label spaces, and evolvable learnware systems [Tan *et al.*, 2024b; Liu *et al.*, 2024; Tan *et al.*, 2024a]. On the privacy side, recent work provides rigorous guarantees for RKME-style specifications through geometric analysis of reduced sets [Lei *et al.*, 2024]. Our work studies the similar general objective but for a fundamentally different specification form. PAVE summarizes model capability through parameter variations induced by fine-tuning [Liu *et al.*, 2026], rather than through representatives in data space. This difference changes both the privacy surface and the mathematical tools needed for analysis.

Our analysis is also related to DP mechanisms for learning and sketching. Standard learning-based DP guarantees are usually obtained by explicit noise injection, such as DP-SGD, RDP, and subsampled-RDP accounting [Abadi *et al.*, 2016; Mironov, 2017; Wang *et al.*, 2019]. Another line shows that random projections or Gaussian sketches can themselves provide privacy guarantees [Blocki *et al.*, 2012; Lev *et al.*, 2025]. Recent work on LoRA privacy studies Wishart projection mechanisms and shows that matrix-valued LoRA-style updates are not generally noise-free DP [Hu *et al.*, 2026]. Our compact PAVE result is different: under dock-side compact-release conditions, the released factor is analyzed as a Gaussian sketch of stable parameter variations, and DP-S-PAVE recovers certified protection by explicit perturbation.

Finally, our risk formulation is motivated by privacy attacks on learning systems. Gradient and update inversion motivate the disclosure risk of releasing PAVE [Zhu *et al.*, 2019; Geiping *et al.*, 2020], while membership, attribute, and model inversion attacks motivate the amplification question when a specification is released together with a model [Fredrikson *et al.*, 2014; Fredrikson *et al.*, 2015; Shokri *et al.*, 2017; Yeom *et al.*, 2018; Salem *et al.*, 2019]. Our DP-to-advantage analysis connects the risks to the privacy of the specification.

6 Concluding Remarks

This paper provides the first formal theoretical analysis for privacy characterization of PAVE specifications in the learnware paradigm. We show that, under natural compact-release conditions of learnware docks, compact PAVE specifications admit an explicit (ε, δ)-differential privacy guarantee through a Gaussian-sketch view of stable parameter variations, without introducing explicit additive noise. For deployments outside this intrinsic regime, we further provide DP-S-PAVE as a certified differentially private variant. We connect these DP guarantees to rigorous bounds on both disclosure risk and specification-side amplification risk, showing that the additional inference advantage attributable to the specification is controlled by the corresponding DP parameters. Finally, we discuss how privacy and identification utility vary with the PAVE generation settings. Our analysis offers useful building blocks for privacy-aware learnware specification design and contributes to a DP-based understanding of dataset leakage through parameter-variation representations.

Ethical Statement

This work studies privacy risks and privacy-preserving guarantees for PAVE specifications in learnware systems. It does not collect new human-subject data or conduct experiments on private user data. The proposed analysis is intended to support privacy-aware deployment by clarifying when compact PAVE releases admit formal differential privacy guarantees and by providing DP-S-PAVE for settings where the structural conditions are not enforced. We also emphasize that practical deployments should verify the required assumptions or use the certified DP variant before releasing specifications.

Acknowledgments

This work was financially supported by FIDBP of MoE (JYB2025XDXM118) and 111 Center (B26023). Tan was supported by the Postdoctoral Fellowship Program of CPSF (GZB20250396) and Jiangsu Funding Program for Excellent Postdoctoral Talent (2025ZB482).

References

[Abadi *et al.*, 2016] Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[Blocki *et al.*, 2012] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The Johnson-Lindenstrauss transform itself preserves differential privacy. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*, pages 410–419, 2012.

[Dwork, 2006] Cynthia Dwork. Differential privacy. In *Proceedings of 33rd International Colloquium on Automata, Languages, and Programming*, pages 1–12, 2006.

[Fredrikson *et al.*, 2014] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: an end-to-end case study of personalized warfarin dosing. In *Proceedings of the 23rd USENIX Security Symposium*, pages 17–32, 2014.

[Fredrikson *et al.*, 2015] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[Geiping *et al.*, 2020] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients: How easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems 33*, pages 16937–16947, 2020.

[Hu *et al.*, 2026] Yaxi Hu, Johanna Düngler, Bernhard Schölkopf, and Amartya Sanyal. LoRA and privacy: When random projections help (and when they don't), 2026. arXiv preprint arXiv:2601.21719.

[Lei *et al.*, 2024] Hao-Yi Lei, Zhi-Hao Tan, and Zhi-Hua Zhou. On the ability of developers' training data preservation of learnware. In *Advances in Neural Information Processing Systems 37*, pages 36471–36513, 2024.

[Lev *et al.*, 2025] Omri Lev, Vishwak Srinivasan, Moshe Shenfeld, Katrina Ligett, Ayush Sekhari, and Ashia C. Wilson. The Gaussian mixing mechanism: Rényi differential privacy via Gaussian sketches, 2025. arXiv preprint arXiv:2505.24603.

[Liu *et al.*, 2024] Jian-Dong Liu, Zhi-Hao Tan, and Zhi-Hua Zhou. Towards making learnware specification and market evolvable. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, pages 13909–13917, 2024.

[Liu *et al.*, 2026] Jian-Dong Liu, Zi-Chen Zhao, Hao Sun, Lin-Xing Wu, Huan Zhang, Pengyuan Wang, Ming Zhao, Xinyu Chu, Shu Yan, Yongbei Zhu, Weijun Zhong, Zhi-Hao Tan, Jing Shang, Yang Yu, and Zhi-Hua Zhou. Constructive specification for plug-and-play learnware agents. In *The ICLR Workshop on Lifelong Agents: Learning, Aligning, Evolving*, 2026.

[Mironov, 2017] Ilya Mironov. Rényi differential privacy. In *Proceedings of the IEEE 30th Computer Security Foundations Symposium*, pages 263–275, 2017.

[Salem *et al.*, 2019] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, 2019.

[Shi *et al.*, 2026] Hao-Yu Shi, Zhi-Hao Tan, Zi-Chen Zhao, Yang Yu, and Zhi-Hua Zhou. A study on PAVE specification for learnware. In *International Conference on Learning Representations*, 2026.

[Shokri *et al.*, 2017] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of the 38th IEEE Symposium on Security and Privacy*, pages 3–18, 2017.

[Tan *et al.*, 2024a] Peng Tan, Hai-Tian Liu, Zhi-Hao Tan, and Zhi-Hua Zhou. Handling learnwares from heterogeneous feature spaces with explicit label exploitation. In *Advances in Neural Information Processing Systems 37*, pages 12767–12795, 2024.

[Tan *et al.*, 2024b] Peng Tan, Zhi-Hao Tan, Yuan Jiang, and Zhi-Hua Zhou. Towards enabling learnware to handle heterogeneous feature spaces. *Machine Learning*, 113(4):1839–1860, 2024.

[Tan *et al.*, 2024c] Zhi-Hao Tan, Jian-Dong Liu, Xiao-Dong Bi, Peng Tan, Qin-Cheng Zheng, Hai-Tian Liu, Yi Xie, Xiao-Chuan Zou, Yang Yu, and Zhi-Hua Zhou. Beimingwu: A learnware dock system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5773–5782, 2024.

[Tan *et al.*, 2025] Zhi-Hao Tan, Zi-Chen Zhao, Hao-Yu Shi, Xin-Yu Zhang, Peng Tan, Yang Yu, and Zhi-

Hua Zhou. Learnware of language models: Specialized small language models can do big. *arXiv preprint arXiv:2505.13425*, 2025.

[Wang *et al.*, 2019] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled Rényi differential privacy and analytical moments accountant. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235, 2019.

[Wu *et al.*, 2023] Xi-Zhu Wu, Wenkai Xu, Song Liu, and Zhi-Hua Zhou. Model reuse with reduced kernel mean embedding specification. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):699–710, 2023.

[Yeom *et al.*, 2018] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *Proceedings of 31st IEEE Computer Security Foundations Symposium*, pages 268–282, 2018.

[Zhang *et al.*, 2020] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 250–258, 2020.

[Zhou and Tan, 2024] Zhi-Hua Zhou and Zhi-Hao Tan. Learnware: Small models do big. *Science China Information Sciences*, 67(1):112102, 2024.

[Zhou, 2016] Zhi-Hua Zhou. Learnware: on the future of machine learning. *Frontiers of Computer Science*, 10(4):589–590, 2016.

[Zhu *et al.*, 2019] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems 32*, pages 14774–14784, 2019.