Sketching, Noise Placement, and Empirical Leakage–Utility Comparisons in Private Ridge Regression

Yewei Xu [1]

Abstract

We study private ridge regression under different sketching and noise-placement choices, organizing mechanisms into three families: full-problem private computation (F), sketch-based randomization (P), and two-stage internal noisy sketches (T). Because regularization, sketch dimension, and noise parameters are strongly coupled, we compare mechanisms on a fixed ridge regression task calibrated via matched empirical leakage, measured by auditing AUC on neighboring datasets. In bounded synthetic experiments, the sketch-based P route is a strong baseline, while the T family moves substantially in the leakage–utility plane as the internal sketch dimension varies; tuned members become highly competitive in specific matched-leakage regions, supporting the interpretation of T as an interpolation family rather than a fixed mechanism. As a controlled extension, sparse-sign sketches preserve the qualitative tradeoff structure but underperform Gaussian sketches. Throughout, auditing AUC serves as a common empirical leakage coordinate, not as a formal replacement for worst-case DP accounting.

1. Introduction

Linear and ridge regression are basic primitives in statistical learning, and their differentially private variants are widely used as test cases for private numerical computation (Aguilera-Martínez & Berzal, 2026). In large-scale settings, computational cost matters as much as privacy and accuracy. Randomized sketching offers a natural way to reduce computation (Mahoney, 2011; Woodruff, 2014; Martinsson & Tropp, 2020), but in the private setting the sketch dimension, ridge regularization, stabilization, and noise scale all interact, and theoretical analyses often rely on conservative sufficient conditions that may not identify the best operating point in practice.

We organize this design space around three mechanism families. F mechanisms apply private computation directly to the full sufficient statistics (Kifer et al., 2012; Zhang et al., 2012). P mechanisms randomize a sketched representation and solve by post-processing; under additional assumptions they can be made private (Blocki et al., 2012; Kenthapadi et al., 2013). T mechanisms sketch internally, add noise to the sketched sufficient statistics, and release only the final estimator, making T a natural interpolation family between full-problem and sketch-based private computation.

We fix a bounded synthetic ridge regression task, scan the primary control knob of each mechanism, and calibrate comparisons using empirical privacy auditing (Jagielski et al., 2020; Nasr et al., 2023): for each parameter setting, we sample outputs on neighboring datasets, train a binary distinguisher, and use the resulting AUC as an empirical leakage score, then compare utility at matched leakage levels. Throughout, P and T are interpreted as empirically audited randomized mechanisms; auditing AUC is used as a common comparison coordinate, not as a formal DP certificate.

1.1. Contributions

Our main contributions are:

- We propose a simple F/P/T framework for studying noise placement and sketching choices in private ridge regression.
- We implement Gaussian full-statistics, sketch-based, and two-stage internal sketch mechanisms under a common bounded synthetic ridge regression setup.
- We introduce a matched empirical leakage protocol based on auditing AUC to compare mechanisms whose formal parameter choices are not directly aligned.
- We find that the T family meaningfully moves in the leakage–utility plane as the internal sketch dimension varies, supporting the interpolation view.
- We test sparse sign sketches as a controlled non-Gaussian extension and find that they preserve the qual-

[1]Department of Mathematics, University of Wisconsin-Madison, Madison, WI, USA. Correspondence to: Yewei Xu <xu464@wisc.edu>.

Preprint. May 1, 2026.

itative tradeoff structure but underperform Gaussian sketches in the current setup.

Roadmap. Section 2 reviews related work. Section 3 sets up the problem formally. Section 4 defines the F, P, and T mechanism families and discusses their formal privacy status (full pseudocode is deferred to Appendix A). Section 5 describes the experimental pipeline and reports the main Gaussian results, T-family interpolation, and V1 sanity checks. Section 6 briefly extends the analysis to sparse-sign sketches. Section 7 discusses insights and limitations, Section 8 outlines precise open questions, and Section 9 concludes. The appendices contain algorithm pseudocode, detailed experimental protocol, additional figures, and diagnostic plots.

2. Related Work

2.1. Private Linear and Ridge Regression

A straightforward approach to private linear regression is to perturb the final solution to the non-private regression problem (Kifer et al., 2012). Alternatively, one can perturb the sufficient statistics such as $X^\top X$ and $X^\top y$, followed by a ridge solve (Zhang et al., 2012). This line gives a natural full-problem baseline for our F family. More refined private regression methods use data-dependent regularization or adaptive mechanisms—such as the Staircase mechanism or adaptive clipping—to improve utility (Wang, 2018), but our first-round experiments focus on simple baselines to keep the comparison transparent.

2.2. Sketching and Private Sketching

Randomized sketches, including Gaussian and sparse sign embeddings, are standard tools for reducing the dimension of least-squares problems (Mahoney, 2011; Woodruff, 2014; Martinsson & Tropp, 2020). In private settings, sketching can also act as a randomized representation of the data, but its privacy and utility depend on spectral conditions, stabilization, and the sketch dimension (Blocki et al., 2012; Kenthapadi et al., 2013; Sheffet, 2019; Lev et al., 2026). Our P mechanism follows this sketch-based route, while our T mechanism uses the sketch only internally.

2.3. Privacy Auditing

Privacy auditing estimates how well an adversary can distinguish outputs produced from neighboring datasets (Jagielski et al., 2020; Muthu Selva Annamalai & De Cristofaro, 2024). We use auditing not as a replacement for formal privacy analysis, but as an empirical calibration tool: mechanisms are compared at matched auditing AUC rather than at nominally matched internal parameters. Recent work has shown that auditing can yield tight lower bounds on privacy loss when the auditor family is sufficiently expressive (Nasr et al., 2023).

3. Problem Setup and Preliminaries

3.1. Data and Ridge Reference

We assume a dataset $D = \{(x_i, y_i)\}_{i=1}^n$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. We consider a bounded setting where $\|x_i\|_2 \le 1$ and $|y_i| \le 1$. In matrix form, we have $X \in \mathbb{R}^{n\times d}$ and $y \in \mathbb{R}^n$.

We fix an outer ridge regression task defined as:

$$\hat{\beta}_{\text{ref}} = \arg\min_{\beta\in\mathbb{R}^d} \|X\beta - y\|_2^2 + \lambda_{\text{task}}\|\beta\|_2^2.$$

This has a closed-form solution:

$$\hat{\beta}_{\text{ref}} = (X^\top X + \lambda_{\text{task}} I)^{-1} X^\top y.$$

The task parameter λ_{task} defines the outer regression problem. Mechanism-specific stabilization terms may be used internally, but all estimators are evaluated against the same outer task.

3.2. Privacy Auditing Preliminaries

Neighboring datasets. We adopt the bounded **replace-one** model: D and D' are neighbors if they differ in a single row, with $\|x_i\|_2 \le 1$ and $|y_i| \le 1$. In our experiments the replacement row is a boundary instance: x'_j uniform on the unit sphere and $y'_j \in \{-1, +1\}$ uniformly.

Definition 3.1 ((ε, δ)-Differential Privacy)**.** Under replace-one, a randomized mechanism M satisfies (ε, δ)-DP if, for all neighboring $D \sim D'$ and measurable $\mathcal{B} \subseteq \mathbb{R}^d$,

$$\Pr[\, M(D) \in \mathcal{B}\,] \;\le\; e^\varepsilon \Pr[\, M(D') \in \mathcal{B}\,] + \delta.$$

Empirical auditing via ROC-AUC. Privacy limits how well an adversary can distinguish D from D' given only M's output. We operationalize this as a distinguishability game: sample $M(D)$ and $M(D')$ repeatedly, train a logistic-regression auditor to guess which dataset produced each output, and measure its success via the ROC-AUC. The auditor's scores are thresholded to trace a ROC curve; the area under this curve, $\text{AUC}_{\text{raw}} \in [0, 1]$, summarizes discriminative power (Bradley, 1997). We fold to $\text{AUC}^* = \max\{\text{AUC}_{\text{raw}}, 1 - \text{AUC}_{\text{raw}}\} \in [0.5, 1]$, with larger values indicating higher empirical leakage.

We use AUC^* as a common comparison coordinate: because F, P, and T have different natural parameters, comparing them at the same AUC^* places them on equal empirical footing. We do *not* convert AUC to formal (ε, δ) bounds; the folded AUC is purely a scalar distinguishability score for the tested neighbor pair and auditor class.

Table 1. Core notation used throughout the paper.

Symbol	Meaning
$D = \{(x_i, y_i)\}_{i=1}^n$	Dataset with $\|x_i\|_2 \le 1$, $\|y_i\| \le 1$
λ_{task}	Outer ridge-regression task parameter
λ_{stab} (family-specific)	Internal stabilization for the ridge solve
r_P, r_T	Sketch dimensions for P and T
c_F, c_T	Noise multipliers for F and T
$(A, b) = (X^\top X, X^\top y)$	Full sufficient statistics
(A_r, b_r)	Sketched sufficient statistics
(X_λ, y_λ)	Ridge-augmented least-squares instance

3.3. Utility Metrics

We evaluate utility through test MSE and parameter error $\text{Err}_\beta(\hat{\beta}) = \|\hat{\beta} - \hat{\beta}_{\text{ref}}\|_2$, where $\hat{\beta}_{\text{ref}}$ is the non-private ridge reference defined above. Because mechanisms are randomized, we also report stability summaries over repeated runs: the median and the 90% quantile (q90). The q90 tells us how bad the error can be in the worst 10% of runs. Means and standard deviations are stored in the experiment outputs.

3.4. Sketching Preliminaries

The ridge objective can be rewritten as a standard least squares problem on an augmented instance $X_\lambda = \begin{bmatrix} X \\ \sqrt{\lambda_{\text{task}}} I_d \end{bmatrix}$ and $y_\lambda = \begin{bmatrix} y \\ 0 \end{bmatrix}$. A Gaussian sketch applies a matrix $S \in \mathbb{R}^{r \times m}$ (where $m = n + d$) with entries $S_{ij} \sim \mathcal{N}(0, 1/r)$ (Blocki et al., 2012; Kenthapadi et al., 2013; Sheffet, 2019). The sketched instance is $\widetilde{X} = SX_\lambda$ and $\widetilde{y} = Sy_\lambda$, yielding sketched sufficient statistics $A_r = \widetilde{X}^\top \widetilde{X}$ and $b_r = \widetilde{X}^\top \widetilde{y}$.

Similarly, we also consider a sparse sign sketch with sparsity parameter s. The entries are independently chosen to be $+\sqrt{s/r}$ with probability $1/(2s)$, $-\sqrt{s/r}$ with probability $1/(2s)$, and 0 otherwise. This matches the second-moment scale of the Gaussian sketch (Achlioptas, 2003). Note that this is an i.i.d. sparse sign sketch, not a CountSketch-style embedding.

4. Mechanism Families

We organize mechanisms into three families. Detailed pseudocode for all variants is given in Appendix A; here we summarize the released object, main tuning knob, and formal privacy status. Throughout, λ_{task} denotes the outer task ridge parameter, λ_{stab} a mechanism-internal numerical stabilization term, r the sketch dimension, c the noise multiplier, S the sketch matrix, (A, b) the full sufficient statistics, and (A_r, b_r) the sketched sufficient statistics. In all experiments, Gaussian noise scales are divided by a fixed factor $q = 8$, so the reported multiplier c is an empirical tuning knob rather than a formal privacy parameter.

4.1. F: Full-Problem Private Computation

F-v0 (Algorithm 1) computes the full sufficient statistics $(A, b) = (X^\top X, X^\top y)$, adds calibrated Gaussian noise to both with scale controlled by a single multiplier c_F, projects the noisy Gramian to the PSD cone, and solves the resulting noisy ridge problem. F-v1 (Algorithm 2) is an alternative baseline that perturbs the final ridge estimator rather than the sufficient statistics, using a heuristic output-noise scale $\sigma_{\beta,F}^{\text{base}} = 2/\lambda_{\text{task}}$ (under unit clipping bounds). Both F variants serve as natural full-problem reference points.

4.2. P: Sketch-based Route

P (Algorithm 3) forms the ridge-lifted instance (X_λ, y_λ), draws a sketch matrix $S \in \mathbb{R}^{r_P \times (n+d)}$, and computes sketched sufficient statistics $A_r = (SX_\lambda)^\top (SX_\lambda)$ and $b_r = (SX_\lambda)^\top (Sy_\lambda)$. The released estimator is obtained by solving the sketched ridge problem; no additional noise is added beyond the randomness of the sketch itself. The primary control knob is the sketch dimension r_P. We treat P as a sketch-based randomized mechanism and do not claim that the sketch route alone gives a formal DP guarantee in full generality.

4.3. T: Two-stage Internal Noisy Sketch

T-v0 (Algorithm 4) first sketches the ridge-lifted instance to dimension r_T, then adds Gaussian noise to the sketched sufficient statistics and solves the noisy sketched ridge problem. The noise base scales shrink with r_T: $\sigma_{A,T}^{\text{base}} = \sigma_{A,F}^{\text{base}}/\sqrt{r_T/d}$ and similarly for $\sigma_{b,T}^{\text{base}}$. T-v1 (Algorithm 5) preserves the two-stage structure but replaces the noisy solve with a clean solve followed by output perturbation, using a heuristic base scale $\sigma_{\beta,T}^{\text{base}} = 2/(\lambda_{\text{task}}\sqrt{r_T/d})$. Both T variants have two control knobs—noise multiplier c_T and internal sketch dimension r_T—making T a natural candidate interpolation family between full-problem private computation and sketch-based computation.

4.4. Formal Privacy Status of Each Mechanism

We briefly summarize what formal DP guarantees each family admits under the bounded replace-one model.

4.4.1. What is Formally Certifiable?

Some of the mechanisms we study can be turned into formally certified Gaussian mechanisms if the noise scales are calibrated to valid global sensitivity bounds. For example, in the bounded replace-one setting, the full sufficient statistics satisfy

$$\Delta_A = \|xx^\top - x'x'^\top\|_F \le \|x\|_2^2 + \|x'\|_2^2 \le 2,$$
$$\Delta_b = \|xy - x'y'\|_2 \le |y|\|x\|_2 + |y'|\|x'\|_2 \le 2.$$

Thus, F-v0 can be viewed as a Gaussian mechanism applied to the vectorized sufficient statistics, followed by post-processing. A fully certified implementation would calibrate a joint Gaussian mechanism to the concatenated query (A, b), whose overall ℓ_2 sensitivity is at most $\sqrt{\Delta_A^2 + \Delta_b^2} \leq 2\sqrt{2}$ under these bounds, or account for the two noisy releases by composition. The PSD projection and the final ridge solve are post-processing steps and therefore do not increase privacy loss.

4.4.2. Formal Status of F-v1

F-v1 is an output perturbation mechanism. It can also be given a formal DP guarantee if the Gaussian output noise is calibrated to a valid sensitivity bound for the ridge estimator under the chosen bounded data model and ridge parameter (Zhang et al., 2012). In this work, we use F-v1 primarily as an empirically calibrated full-problem baseline rather than as a fully certified mechanism.

4.4.3. Formal Status of P and T

For P and T, the formal privacy status is more subtle. The random sketch alone is not automatically a distribution-free DP mechanism in the same way as a calibrated Gaussian mechanism on a bounded-sensitivity query (Dwork & Roth, 2014). Formal privacy statements for sketch-based releases generally require additional assumptions, stabilization, or careful accounting of the released object (Blocki et al., 2012; Kenthapadi et al., 2013; Sheffet, 2019). In our experiments, we therefore treat P and T as randomized mechanisms and evaluate their induced leakage empirically.

For T in particular, the sketch is internal and only the final estimator is released. This differs from releasing the sketch itself, but a complete formal analysis would still require bounding the sensitivity of the final randomized map from datasets to outputs. We leave this as future work.

5. Experiments

5.1. Experimental Pipeline

Our pipeline has six stages: (i) generate a bounded synthetic dataset ($n_{\text{train}} = 2000$, $n_{\text{test}} = 5000$, $d = 200$, condition number $\kappa = 50$, β_{true} unit-norm, labels with $\sigma = 0.1$ noise, all clipped to unit norm); (ii) construct a neighboring dataset D' via boundary replace-one; (iii) compute the non-private ridge reference $\hat{\beta}_{\text{ref}}$; (iv) run each mechanism 300 times on both D and D' with independent seeds; (v) train a logistic-regression auditor on standardized output vectors and compute folded AUC*; and (vi) select the closest parameter setting to each target AUC$^* \in \{0.55, 0.60, 0.65, 0.70, 0.75, 0.85\}$ (tolerance 0.02) and compare utility at matched leakage.

For sketch-based mechanisms, r is chosen in $[d, n_{\text{train}}]$ so that A_r remains positive definite while keeping the sketch dimension below the full sample size. P scans $r_P \in [220, 1500]$; T uses $r_T/d \in [2, 4]$. Exhaustive parameter grids and code-level details (data generation, neighbor construction, auditor implementation) are given in Appendix B.

We emphasize that matched AUC* is an empirical comparison coordinate: a higher AUC means the auditor better distinguishes $M(D)$ from $M(D')$ for this neighbor pair and auditor class. We do not convert AUC to formal (ε, δ) bounds; rather, matching mechanisms at the same AUC places them on equal empirical footing before comparing utility.

5.2. Main Gaussian Comparison

Table 3 reports the matched-leakage comparison at four representative targets. The three families occupy clearly distinct regions. Among the fixed mechanisms, P is the strongest baseline across the main matched-leakage targets: at AUC$^* \approx 0.55$, P achieves MSE 0.031 with $r_P = 280$, roughly $17\times$ better than F-v0 (MSE 0.535, $c_F = 6.0$); at higher leakage the gap narrows but P retains a clear edge. The fixed T-v0 baseline ($r_T/d = 2$) consistently sits between F-v0 and P, confirming the two-stage pipeline's intermediate position. However, the selected T-family rows at $r_T/d = 3, 4$ reveal that this intermediate position is tunable: at AUC$^* \approx 0.60$, T-v0 with $r_T/d = 3$ (MSE 0.017, $c_T = 0.9$) actually outperforms P (MSE 0.027), while at AUC$^* \approx 0.55$, $r_T/d = 4$ degrades to MSE 0.123, much worse than the $r_T/d = 2$ baseline. This non-monotone behavior motivates treating T as a family whose members must be selected per leakage target.

Runtime differences are modest in this small dense-sketch regime: F-v0 is fastest ($\sim$ 0.01s), P and T are comparable ($\sim$ 0.02–0.05s). True compute gains require larger n or structured sketches.

5.3. T-v0-family Interpolation

Varying r_T/d moves T-v0 substantially in the leakage–utility plane (Figure 2, Table 3). At AUC$^* \approx 0.55$, raising r_T/d from 2 to 3 cuts MSE from 0.037 to 0.025, but further increasing to $r_T/d = 4$ degrades MSE to 0.123. At AUC$^* \approx 0.60$, $r_T/d = 3$ achieves MSE 0.017, surpassing P (0.027) and F-v0 (0.061)—the strongest evidence that tuned T-family members can beat the sketch-based baseline. At AUC$^* \approx 0.65$, the optimal r_T/d is again 3 (MSE 0.016), competitive with P (0.019). The $r_T/d = 4$ setting consistently underperforms $r_T/d = 3$ across all leakage targets, suggesting that $r_T/d \approx 3$ is the sweet spot in this regime. The fact that the same mechanism family spans a wide utility range in the low-to-moderate leakage regime—

Table 2. Summary of mechanism families and variants. All mechanisms release a ridge estimator $\hat{\beta} \in \mathbb{R}^d$. The formal-status column reflects what is certifiable under the bounded replace-one model without additional assumptions.

Mechanism	Main knob	Sketching	Noise placement	Formal status
F-v0	Noise multiplier c_F	No	On suff. stats (A, b)	Gaussian mechanism on vectorized stats + post-processing: PSD projection and ridge solve
F-v1	Noise multiplier c_F	No	On output $\hat{\beta}$	Output perturbation; certifiable if a valid sensitivity bound for the ridge estimator is available
P	Sketch dimension r_P	Yes	Sketch randomness only	Not automatically DP in full generality without extra structural assumptions
T-v0	Noise multiplier c_T at fixed r_T	Yes	On sketched suff. stats (A_r, b_r)	Requires bounding the sensitivity of the full randomized map from dataset to output
T-v1	Noise multiplier c_T at fixed r_T	Yes	On output $\hat{\beta}$	Requires bounding the sensitivity of the full randomized map from dataset to output

Table 3. Gaussian V0 matched-leakage comparison, including selected T-family configurations ($\lambda_{\text{task}} = 1.0$). For T-v0, $r_T/d = 2$ is the fixed baseline, while $r_T/d = 3, 4$ rows illustrate the effect of varying the internal sketch dimension. For each target AUC* we select the closest available parameter setting (tolerance 0.02). Utility and runtime are computed for outputs sampled from $M(D)$; MSE is evaluated on the held-out test set.

Target AUC*	Mechanism	Matched AUC*	Parameter	MSE median	MSE q90	Param. error med.	Runtime (s)
0.55	F-v0	0.556	$c = 6.0$	0.5348	0.6606	13.539	0.0102
	P	0.540	$r = 280$	0.0306	0.0359	3.125	0.0180
	T-v0 ($r_T/d = 2$)	0.550	$c = 1.8$	0.0368	0.0447	3.475	0.0305
	T-v0 ($r_T/d = 3$)	0.549	$c = 1.8$	0.0249	0.0291	2.711	0.0434
	T-v0 ($r_T/d = 4$)	0.536	$c = 6.0$	0.1229	0.1567	6.863	0.0506
0.60	F-v0	0.602	$c = 2.5$	0.0608	0.0754	5.184	0.0103
	P	0.582	$r = 300$	0.0268	0.0305	2.821	0.0202
	T-v0 ($r_T/d = 2$)	0.597	$c = 1.2$	0.0266	0.0310	2.720	0.0311
	T-v0 ($r_T/d = 3$)	0.602	$c = 0.9$	0.0170	0.0185	1.810	0.0400
	T-v0 ($r_T/d = 4$)	0.607	$c = 3.0$	0.0343	0.0403	3.523	0.0498
0.65	F-v0	0.644	$c = 1.8$	0.0311	0.0373	3.575	0.0085
	P	0.654	$r = 400$	0.0185	0.0204	1.975	0.0236
	T-v0 ($r_T/d = 2$)	0.656	$c = 0.9$	0.0220	0.0251	2.289	0.0300
	T-v0 ($r_T/d = 3$)	0.653	$c = 0.6$	0.0156	0.0166	1.571	0.0400
0.70	F-v0	0.715	$c = 1.4$	0.0209	0.0238	2.679	0.0075
	P	0.699	$r = 500$	0.0161	0.0171	1.621	0.0306
	T-v0 ($r_T/d = 2$)	0.690	$c = 0.6$	0.0193	0.0216	1.978	0.0304

from MSE 0.123 to 0.025 around $\text{AUC}^* \approx 0.55$, down to MSE 0.017 around $\text{AUC}^* \approx 0.60$—purely by varying the internal sketch dimension confirms that T-v0 is a tunable interpolation family, not a single fixed compromise.

5.4. V1-family Sanity Check

We also evaluated V1 variants that perturb the output rather than adding noise to sufficient statistics. These experiments are diagnostic rather than central: output-perturbation-style variants can be competitive in some regimes, but the heuristic sensitivity scaling becomes unstable when λ_{task} is small. The qualitative F/P/T ordering is preserved, and T-v1 exhibits interpolation behavior similar to T-v0 as r_T/d varies. We therefore keep the Gaussian V0 mechanisms as the main experimental evidence and report V1 plots (Gaussian and sparse-sign) in the appendix.

6. Extensions to Non-Gaussian Sketching

6.1. Sparse-sign Sketching

As a controlled non-Gaussian extension, we replace the Gaussian sketch with an i.i.d. sparse-sign sketch (density $1/s$) (Achlioptas, 2003), which preserves the same second-moment scaling but has a different entry distribution and sparsity structure.

6.2. Sparse P and T-v0 Mechanisms

We reran the matched-leakage protocol with sparse-sign sketches at both $\lambda_{\text{task}} = 1.0$ and $\lambda_{\text{task}} = 0.1$ (Figures 3–

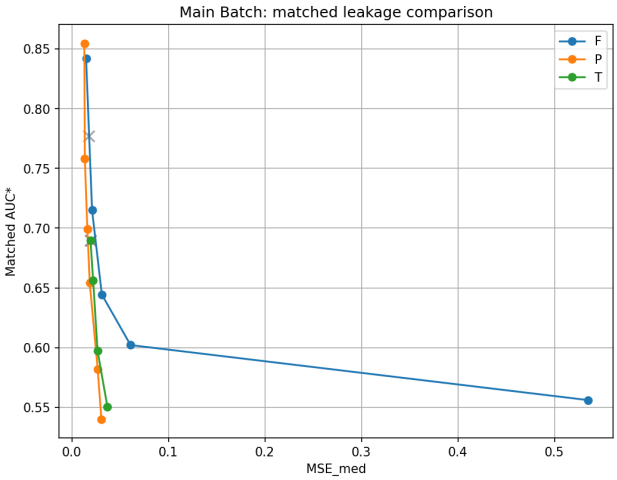

Figure 1. V0 Gaussian main comparison at matched empirical leakage ($\lambda_{\text{task}} = 1.0$). Each point selects the closest available parameter setting for a target auditing AUC. F-v0 degrades sharply in the low-leakage region, P is strong across a wide range, and T-v0 with $r_T/d = 2$ forms a middle region but is sensitive to the fixed internal sketch dimension.

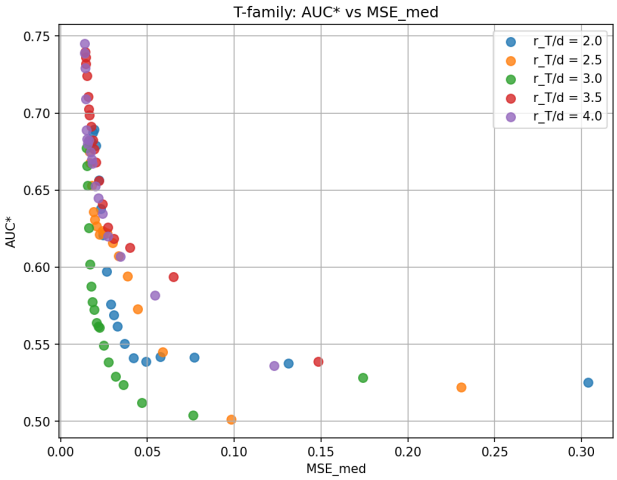

Figure 2. V0 T-family interpolation under Gaussian sketches at $\lambda_{\text{task}} = 1.0$. Varying r_T/d moves T-v0 substantially in the leakage–utility plane. The effect is not monotone at every matched leakage level, but tuned intermediate sketch dimensions can be highly competitive.

4). The qualitative behavior is consistent with the Gaussian experiments: P remains a strong sketch-randomization baseline, and varying r_T/d again moves the T family across the leakage–utility plane. Sparse T-v0 with $r_T/d = 2$ cannot reach the lowest leakage targets ($\text{AUC}^* \leq 0.60$) even at maximum noise, but increasing r_T/d partially recovers utility, and $r_T/d \approx 3$ again emerges as the most competitive operating point. Quantitatively, the sparse-sign variants are generally weaker and less stable than their Gaussian counterparts in the current dense implementation; they do not overturn the Gaussian V0 comparison, which remains our main evidence. Sparse V1 output-perturbation variants show similar diagnostic behavior and are reported in the appendix.

7. Discussion

7.1. Insights into the T Mechanism Family

The clearest empirical takeaway is that T is not a single operating point but a tunable family. Varying the internal sketch dimension lets T navigate the leakage–utility plane, interpolating between full noisy statistics (small r_T, high noise needed) and purely sketch-based methods (large r_T, low noise). At $r_T/d \approx 3$, T-v0 becomes competitive with—and across multiple matched-leakage targets ($\text{AUC}^* \in \{0.55, 0.60, 0.65\}$) surpasses—the sketch-based P baseline, demonstrating that the two-stage design is not merely an intermediate compromise but a genuinely useful degree of freedom.

7.2. Limitations

Our evaluation is confined to a bounded synthetic setting with fixed dimensions and a single boundary replace-one neighbor construction. The auditing AUC is an empirical lower bound for the chosen neighbor pair and auditor class, not a universal privacy certificate. The F family uses basic sufficient-statistics perturbation; stronger adaptive baselines (e.g., SSP-type mechanisms) could sharpen the comparison. Finally, the dense-sketch implementation at $d = 200$ does not demonstrate practical compute savings, which would require larger-scale or structured-sketch experiments.

8. Future Directions

We highlight three precise open questions that follow from this work.

Formal privacy of the T family. Can the two-stage T family be certified under replace-one DP or Rényi DP with bounds depending explicitly on the sketch dimension, spectral conditioning, and stable rank? A formal analysis must bound the sensitivity of the full randomized map $D \mapsto \hat{\beta}_T$, accounting for both the sketch randomness (which induces a distribution over intermediate representations) and the final Gaussian noise.

Robustness of empirical leakage curves. Do the T-family interpolation curves persist under neighbor constructions beyond the boundary replace-one used here—e.g., random within-ball replacements, replacements along spectral directions of $X^\top X$, or add/remove neighbors? Robustness across neighbor models would strengthen the empirical case

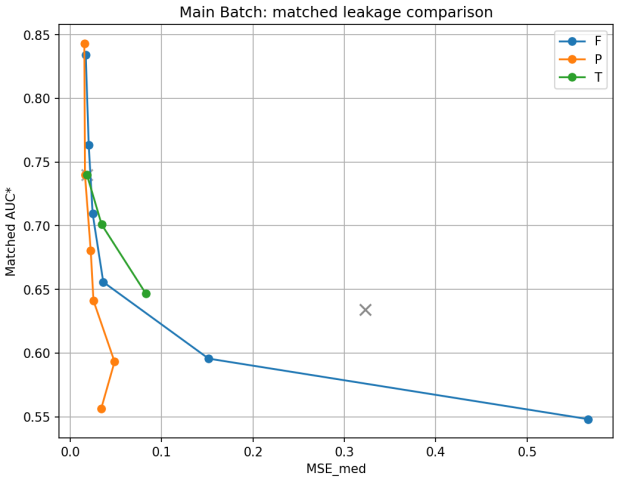

Figure 3. V0 sparse-sign main comparison at $\lambda_{\text{task}} = 0.1$. The qualitative structure resembles the Gaussian counterpart, though sparse-sign P is weaker in this regime.

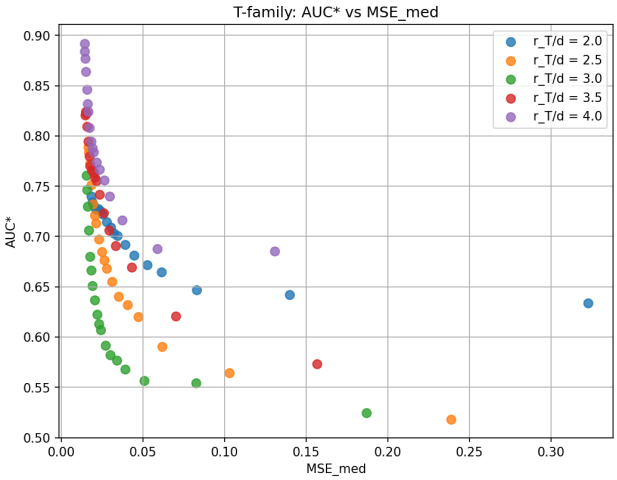

Figure 4. V0 T-family interpolation under sparse-sign sketches at $\lambda_{\text{task}} = 0.1$. Varying r_T/d preserves the interpolation behavior seen under Gaussian sketches.

for tuning r_T/d as a leakage–utility design parameter.

Compute-aware sketch design. Can sparse or structured sketches (e.g., CountSketch, Subsampled Randomized Hadamard Transform) preserve the same qualitative leakage–utility tradeoff while delivering actual runtime and memory gains at scale ($n \gg 10^4$, $d \gg 200$)? The current dense-sketch implementation is a proof of concept; genuine compute savings require moving beyond small dense matrices.

9. Conclusion

We studied private ridge regression through the lens of sketching and noise placement, organizing mechanisms into F, P, and T families and comparing them at matched empirical leakage via auditing AUC. The sketch-based P route is a strong baseline, while the two-stage T mechanism acts as a genuine interpolation family: varying r_T/d moves it across the leakage–utility plane, and at $r_T/d \approx 3$ it can surpass P in specific matched-leakage regions. Sparse-sign sketches preserve the qualitative tradeoff structure but underperform Gaussian sketches. Taken together, the experiments demonstrate that noise placement and sketch dimension should be studied jointly, and that auditing AUC provides a practical, principled coordinate for comparing mechanisms with otherwise incommensurable parameterizations.

References

Achlioptas, D. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003. ISSN 0022-0000. doi: 10.1016/S0022-0000(03)00025-4. Special Issue on PODS 2001.

Aguilera-Martínez, F. and Berzal, F. Differential privacy in machine learning: A survey from symbolic ai to llms, 2026. URL https://arxiv.org/abs/2506.11687.

Blocki, J., Blum, A., Datta, A., and Sheffet, O. The johnson-lindenstrauss transform itself preserves differential privacy. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pp. 410–419, 2012. doi: 10.1109/FOCS.2012.67.

Bradley, A. P. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. ISSN 0031-3203. doi: https://doi.org/10.1016/S0031-3203(96)00142-2.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014. ISSN 1551-305X. doi: 10.1561/0400000042.

Jagielski, M., Ullman, J., and Oprea, A. Auditing differentially private machine learning: How private is private sgd? In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 22205–22216. Curran Associates, Inc., 2020.

Kenthapadi, K., Korolova, A., Mironov, I., and Mishra, N. Privacy via the johnson-lindenstrauss transform. *Journal of Privacy and Confidentiality*, 5(1), Aug. 2013. doi: 10.29012/jpc.v5i1.625.

Kifer, D., Smith, A., and Thakurta, A. Private convex empirical risk minimization and high-dimensional regression.

In Mannor, S., Srebro, N., and Williamson, R. C. (eds.), *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pp. 25.1–25.40, Edinburgh, Scotland, 25–27 Jun 2012. PMLR.

Lev, O., Srinivasan, V., Shenfeld, M., Ligett, K., Sekhari, A., and Wilson, A. C. The gaussian mixing mechanism: Renyi differential privacy via gaussian sketches. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=IjqTJELKU1.

Mahoney, M. W. Randomized algorithms for matrices and data. *Found. Trends Mach. Learn.*, 3(2):123–224, February 2011. ISSN 1935-8237. doi: 10.1561/2200000035.

Martinsson, P.-G. and Tropp, J. A. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29:403–572, 2020. doi: 10.1017/S0962492920000021.

Muthu Selva Annamalai, M. S. and De Cristofaro, E. Nearly tight black-box auditing of differentially private machine learning. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 131482–131502. Curran Associates, Inc., 2024. doi: 10.52202/079017-4179.

Nasr, M., Hayes, J., Steinke, T., Balle, B., Tramèr, F., Jagielski, M., Carlini, N., and Terzis, A. Tight auditing of differentially private machine learning. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 1631–1648, Anaheim, CA, August 2023. USENIX Association. ISBN 978-1-939133-37-3. URL https://www.usenix.org/conference/usenixsecurity23/presentation/nasr.

Sheffet, O. Old techniques in differentially private linear regression. In Garivier, A. and Kale, S. (eds.), *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, volume 98 of *Proceedings of Machine Learning Research*, pp. 789–827. PMLR, 22–24 Mar 2019.

Wang, Y.-X. Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain. In *Uncertainty in Artificial Intelligence (UAI-18)*, 2018.

Woodruff, D. P. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1–2):1–157, October 2014. ISSN 1551-305X. doi: 10.1561/0400000060.

Zhang, J., Zhang, Z., Xiao, X., Yang, Y., and Winslett, M. Functional mechanism: regression analysis under differential privacy. *Proc. VLDB Endow.*, 5(11):1364–1375, July 2012. ISSN 2150-8097. doi: 10.14778/2350229.2350253.

A. Algorithm Pseudocode

Algorithms 1–5 provide the complete pseudocode for all five mechanism variants discussed in Section 4. In all algorithms, noise scales are divided by $q = 8$ (see Section 4).

Algorithm 1 F-v0: noisy full sufficient statistics

Input: training data (X, y), task ridge parameter λ_{task}, stabilization parameter $\lambda_{F,\text{stab}}$, noise multiplier c_F.
Compute full sufficient statistics $A = X^\top X$, $b = X^\top y$.
Set noise scales $\sigma_A = c_F \sigma_{A,F}^{\text{base}}/q$ and $\sigma_b = c_F \sigma_{b,F}^{\text{base}}/q$.
Draw symmetric matrix noise $N_A = (G + G^\top)/2$ with $G_{ij} \sim \mathcal{N}(0, \sigma_A^2)$, and vector noise $n_b \sim \mathcal{N}(0, \sigma_b^2 I_d)$.
Form $\bar{A}_F = A + \lambda_{\text{task}} I + N_A$ and $\bar{b}_F = b + n_b$.
Project $\bar{A}_F$ to the PSD cone with eigenvalue floor ρ_F: $\widetilde{A}_F = \Pi_{\succeq \rho_F I}(\bar{A}_F)$.
Release $\hat{\beta}_{F,\text{v0}} = (\widetilde{A}_F + \lambda_{F,\text{stab}} I)^{-1} \bar{b}_F$.

Algorithm 2 F-v1: output perturbation

Input: training data (X, y), task ridge parameter λ_{task}, noise multiplier c_F.
Compute the non-private ridge solution $\hat{\beta}_{\text{ridge}} = (X^\top X + \lambda_{\text{task}} I)^{-1} X^\top y$.
Set output noise scale $\sigma_\beta = c_F \sigma_{\beta,F}^{\text{base}}/q$.
Draw $z_F \sim \mathcal{N}(0, \sigma_\beta^2 I_d)$.
Release $\hat{\beta}_{F,\text{v1}} = \hat{\beta}_{\text{ridge}} + z_F$.

Algorithm 3 P: sketch-based ridge solve

Input: training data (X, y), task ridge parameter λ_{task}, sketch dimension r_P, stabilization parameter $\lambda_{P,\text{stab}}$, sketch distribution.
Form the ridge-lifted instance $X_\lambda = [X^\top, \sqrt{\lambda_{\text{task}}} I_d]^\top$ and $y_\lambda = [y^\top, 0]^\top$.
Draw a sketch matrix $S \in \mathbb{R}^{r_P \times (n+d)}$. For Gaussian sketches, $S_{ij} \sim \mathcal{N}(0, 1/r_P)$; for sparse-sign sketches, $S_{ij} \in \{0, \pm\sqrt{s/r_P}\}$.
Compute $\widetilde{X} = S X_\lambda$ and $\widetilde{y} = S y_\lambda$.
Form sketched sufficient statistics $A_r = \widetilde{X}^\top \widetilde{X}$ and $b_r = \widetilde{X}^\top \widetilde{y}$.
Release $\hat{\beta}_P = (A_r + \lambda_{P,\text{stab}} I)^{-1} b_r$.

Algorithm 4 T-v0: internal sketch followed by noisy sketched solve

Input: training data (X, y), task ridge parameter λ_{task}, internal sketch dimension r_T, noise multiplier c_T, stabilization parameter $\lambda_{T,\text{stab}}$.
Form the ridge-lifted instance (X_λ, y_λ).
Draw an internal sketch matrix $S \in \mathbb{R}^{r_T \times (n+d)}$.
Compute $\widetilde{X} = S X_\lambda$ and $\widetilde{y} = S y_\lambda$.
Form sketched sufficient statistics $A_r = \widetilde{X}^\top \widetilde{X}$ and $b_r = \widetilde{X}^\top \widetilde{y}$.
Set noise scales $\sigma_{A,T} = c_T \sigma_{A,T}^{\text{base}}/q$ and $\sigma_{b,T} = c_T \sigma_{b,T}^{\text{base}}/q$.
Add Gaussian noise: $\bar{A}_T = A_r + N_A$ and $\bar{b}_T = b_r + n_b$.
Project $\bar{A}_T$ to the PSD cone with floor ρ_T: $\widetilde{A}_T = \Pi_{\succeq \rho_T I}(\bar{A}_T)$.
Release $\hat{\beta}_{T,\text{v0}} = (\widetilde{A}_T + \lambda_{T,\text{stab}} I)^{-1} \bar{b}_T$.

Algorithm 5 T-v1: internal sketch with output perturbation

Input: training data (X, y), task ridge parameter λ_{task}, internal sketch dimension r_T, output noise multiplier c_T, stabilization parameter $\lambda_{T,\text{stab}}$, sketch distribution.
Form the ridge-lifted instance (X_λ, y_λ).
Draw an internal sketch matrix $S \in \mathbb{R}^{r_T \times (n+d)}$. For Gaussian sketches, $S_{ij} \sim \mathcal{N}(0, 1/r_T)$; for sparse-sign sketches, $S_{ij} \in \{0, \pm\sqrt{s/r_T}\}$.
Compute the sketched instance $\widetilde{X} = SX_\lambda$ and $\widetilde{y} = Sy_\lambda$.
Solve the sketched ridge problem $\hat{\beta}_{\text{sketch}} = (\widetilde{X}^\top \widetilde{X} + \lambda_{T,\text{stab}} I)^{-1} \widetilde{X}^\top \widetilde{y}$.
Draw output noise $z_T \sim \mathcal{N}(0, \sigma_{\beta,T}^2 I_d)$ with $\sigma_{\beta,T} = c_T \sigma_{\beta,T}^{\text{base}}/q$.
Release $\hat{\beta}_{T,\text{v1}} = \hat{\beta}_{\text{sketch}} + z_T$.

B. Detailed Experimental Protocol

This appendix provides the code-level details that were omitted from the main-text description of the experimental pipeline (Section 5.1).

Code-level data generation. The function `generate_synthetic_ridge_data` draws a true parameter $\beta_{\text{true}} \sim \mathcal{N}(0, I_d)$, normalizes it to unit length, and scales by $\text{signal\_scale} = 1$. The feature covariance Σ has eigenvalues spaced geometrically from 1 to $\kappa_{\text{feature}} = 50$ (`np.geomspace(50, 1, d)`), rotated by a random orthogonal matrix Q from the QR decomposition of a $d \times d$ standard Gaussian matrix. Raw features $Z \sim \mathcal{N}(0, I_d)$ of size $(n_{\text{train}} + n_{\text{test}}) \times d$ are transformed by $\Sigma^{1/2} = Q\text{diag}(\sqrt{\lambda})Q^\top$, then row-wise clipped: any row with $\|x_i\|_2 > 1$ is rescaled to unit norm. Labels are $y_i = x_i^\top \beta_{\text{true}} + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, 0.1^2)$, clipped to $|y_i| \le 1$.

Neighbor generation. D' is built by `make_neighbor_dataset`, which copies the training arrays and replaces row $j = 0$ with a boundary example. The boundary feature x'_j is drawn from a standard Gaussian, normalized to unit length, and scaled by $\text{clip\_x} = 1$. The boundary label is $y'_j \in \{-1, +1\}$ uniformly at random. This is the replace-one perturbation of Section 3.2: the dataset size stays n_{train} and exactly one row changes. The neighbor seed is fixed (1) so that D' is deterministic once D is fixed, ensuring reproducibility.

Auditor implementation. The 600 released vectors form a binary classification dataset (label 0 for D, 1 for D'), stacked into a $600 \times d$ feature matrix. The data is split 70/30 stratified (420 train, 180 test) using scikit-learn's `train_test_split` with a fixed random state. Coordinates are standardized to unit variance based on the training split only: $\mu_j = \frac{1}{n_{\text{train}}} \sum_i x_{ij}$, $\sigma_j = \max\{\text{std}_j, 10^{-12}\}$, with both splits transformed by $(x - \mu)/\sigma$. The auditor is an L2-regularized `LogisticRegression` classifier (default regularization, $\text{max\_iter} = 1000$) whose sole input is $\hat{\beta} \in \mathbb{R}^d$; it has no access to the original data or mechanism randomness. On the test split, the auditor outputs probability scores $p_i \in [0, 1]$, which are fed to `roc_auc_score` to compute AUC_{raw}. We fold to $\text{AUC}^* = \max\{\text{AUC}_{\text{raw}}, 1 - \text{AUC}_{\text{raw}}\}$ and record $\text{Adv} = 2|\text{AUC}_{\text{raw}} - 0.5|$.

Utility metrics. For each of the 300 runs on D, test MSE is $\frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (x_i^\top \hat{\beta} - y_i)^2$ and parameter error is $\|\hat{\beta} - \beta_{\text{ref}}\|_2$ where β_{ref} is the non-private ridge solution. We report median and 90% quantile (q90); means and standard deviations are stored in the experiment outputs.

Exhaustive parameter grids. *V0 main batch at $\lambda_{\text{task}} = 1.0$ (Gaussian).* F-v0: $c_F \in \{0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.25, 2.5, 2.75, 3.0, 3.5, 4.0, 5.0, 6.0, 8.0\}$ (16 values). P: $r_P \in \{220, 250, 280, 300, 330, 360, 400, 450, 500, 560, 620, 700, 800, 950, 1100, 1300, 1500\}$ (17 values). T-v0: $c_T \in \{0.4, 0.5, 0.6, 0.75, 0.9, 1.0, 1.1, 1.25, 1.4, 1.5, 1.6, 1.8, 2.0, 2.25, 2.5, 3.0, 4.0, 6.0\}$ (18 values) at fixed $r_T/d = 2$.

V0 T-family interpolation at $\lambda_{\text{task}} = 1.0$ (Gaussian). $r_T/d \in \{2, 2.5, 3, 3.5, 4\}$ (i.e. $r_T \in \{400, 500, 600, 700, 800\}$ when $d = 200$), with the same noise grid as above.

V1 main batch at $\lambda_{\text{task}} = 0.1$ (Gaussian). F-v1: $c_F \in \{0.01, 0.02, 0.03, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 1.0, 1.2\}$ (14 values). P: $r_P \in \{220, 250, 280, 300, 330, 360, 400, 450, 500, 560, 620, 700, 800, 950, 1200, 1400, 1600, 1800, 2000\}$ (19 values). T-v1: $c_T \in \{0.1, 0.2, 0.5, 0.8, 1.0, 1.2, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 6.0\}$ (15 values) at fixed $r_T/d = 6$.

V1 T-family interpolation at $\lambda_{\text{task}} = 0.1$ (Gaussian). $r_T/d \in \{6, 7, 8, 9, 10\}$ (i.e. $r_T \in \{1200, 1400, 1600, 1800, 2000\}$), with the same noise grid as above.

The sparse-sign experiments use identical grid structures with the sketch family switched from Gaussian to i.i.d. sparse-sign with density $1/3$.

C. Additional Main-Batch and T-Family Comparisons

C.1. V0 Mechanisms at $\lambda_{\text{task}} = 0.1$ (Gaussian)

Figure 5 and Figure 6 show the Gaussian main comparison and T-family interpolation for the V0 mechanisms under $\lambda_{\text{task}} = 0.1$.

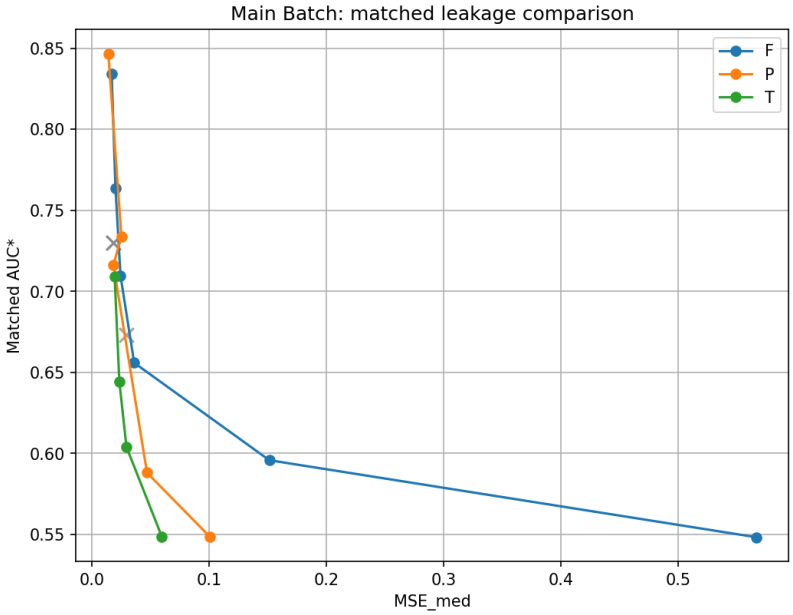

Figure 5. V0 Gaussian main comparison at $\lambda_{\text{task}} = 0.1$.

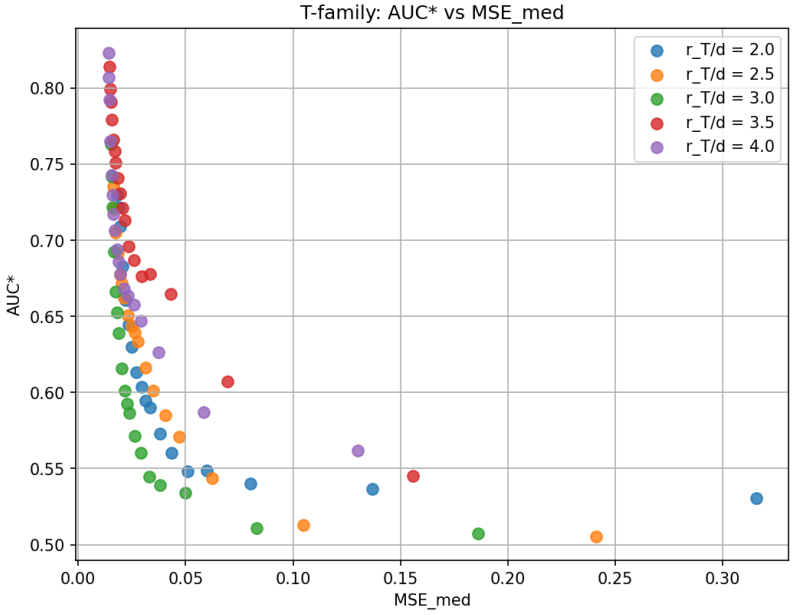

Figure 6. V0 T-family interpolation under Gaussian sketches at $\lambda_{\text{task}} = 0.1$.

C.2. V0 Mechanisms at $\lambda_{\text{task}} = 1.0$ (Sparse)

Figure 7 and Figure 8 show the sparse-sign main comparison and T-family interpolation for the V0 mechanisms under $\lambda_{\text{task}} = 1.0$.

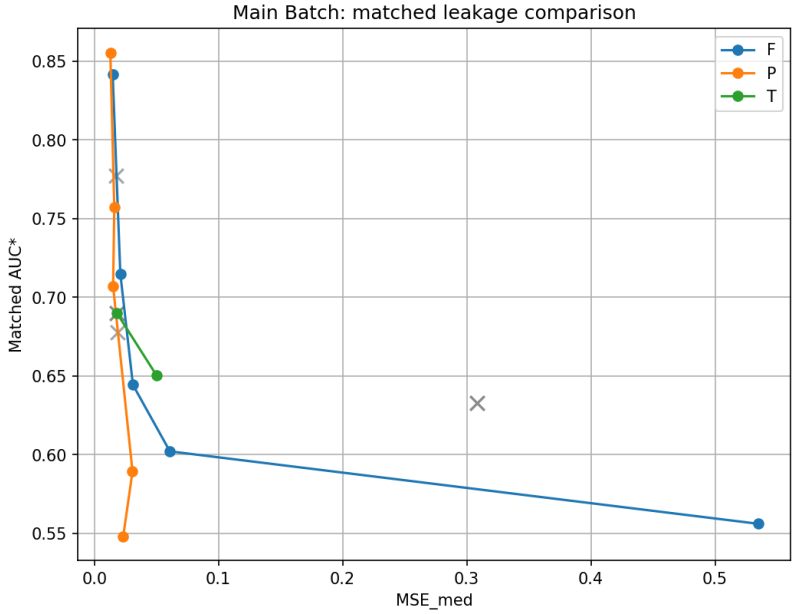

Figure 7. V0 sparse-sign main comparison at $\lambda_{\text{task}} = 1.0$.

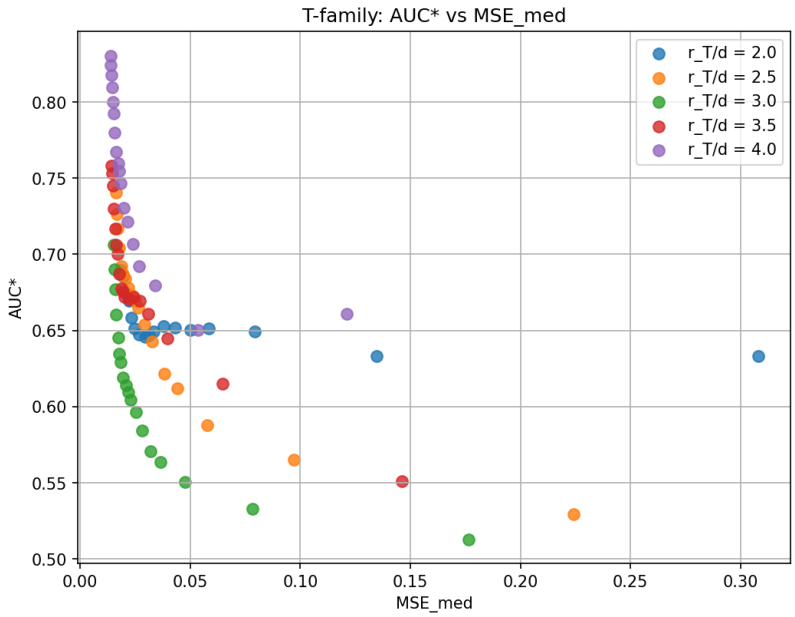

Figure 8. V0 T-family interpolation under sparse-sign sketches at $\lambda_{\text{task}} = 1.0$.

C.3. V1 Gaussian Mechanisms at $\lambda_{\text{task}} = 0.1$

Figure 9 and Figure 10 show the Gaussian main comparison and T-family interpolation for the V1 mechanisms under $\lambda_{\text{task}} = 0.1$.

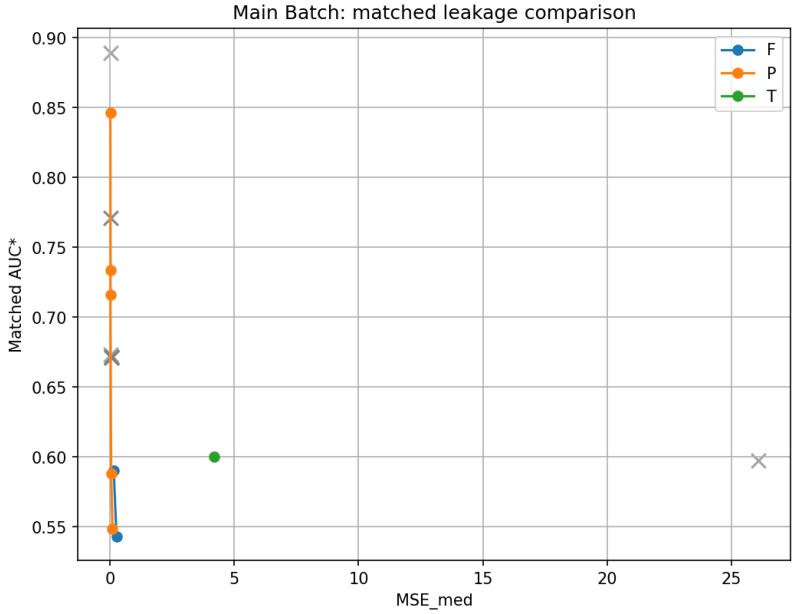

Figure 9. V1 Gaussian main comparison at $\lambda_{\text{task}} = 0.1$.

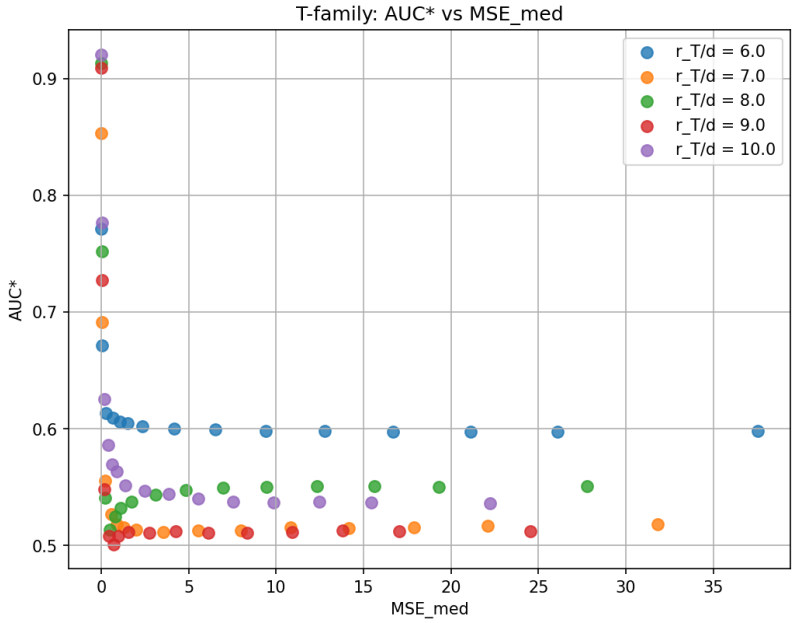

Figure 10. V1 T-family interpolation under Gaussian sketches at $\lambda_{\text{task}} = 0.1$.

C.4. V1 Gaussian Mechanisms at $\lambda_{\text{task}} = 1.0$

Figure 11 and Figure 12 show the Gaussian main comparison and T-family interpolation for the V1 mechanisms under $\lambda_{\text{task}} = 1.0$.

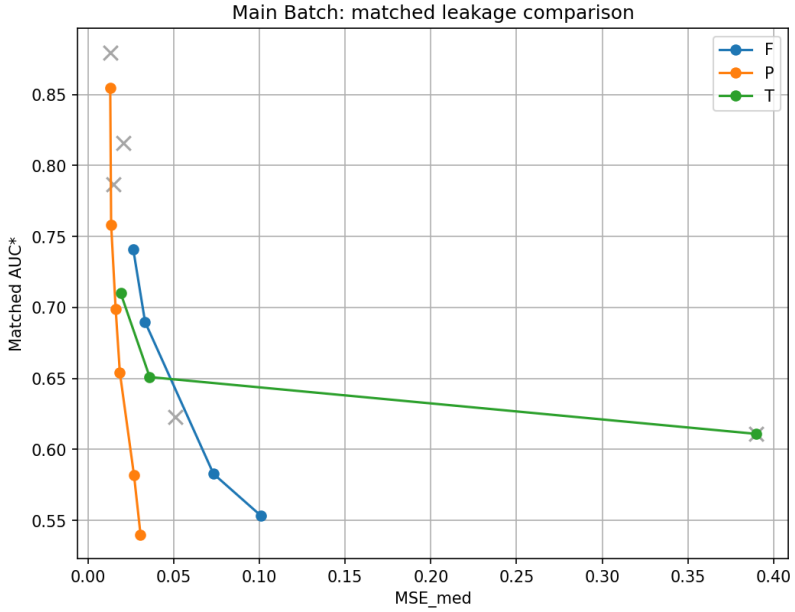

Figure 11. V1 Gaussian main comparison at $\lambda_{\text{task}} = 1.0$.

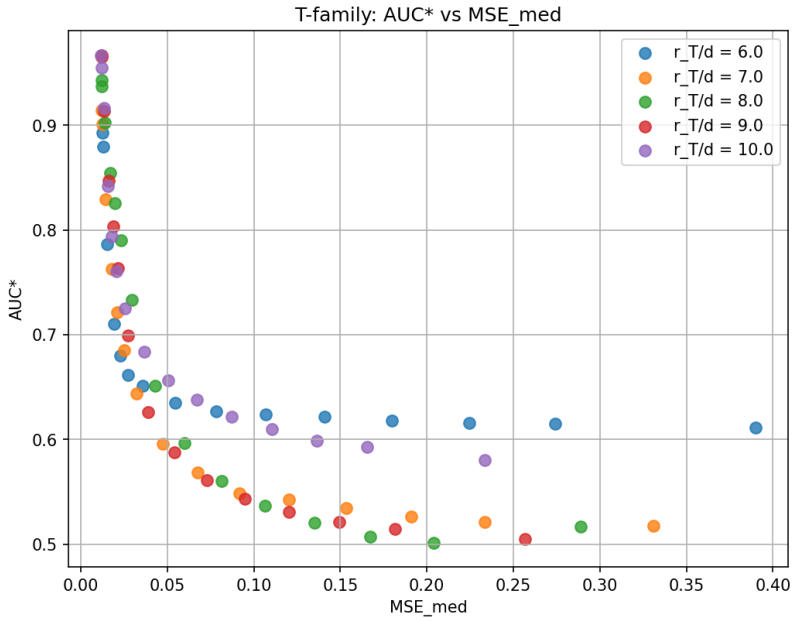

Figure 12. V1 T-family interpolation under Gaussian sketches at $\lambda_{\text{task}} = 1.0$.

C.5. V1 Sparse Mechanisms at $\lambda_{\text{task}} = 1.0$

Figure 13 and Figure 14 report the sparse-sign main comparison and T-family interpolation for the V1 mechanisms under $\lambda_{\text{task}} = 1.0$.

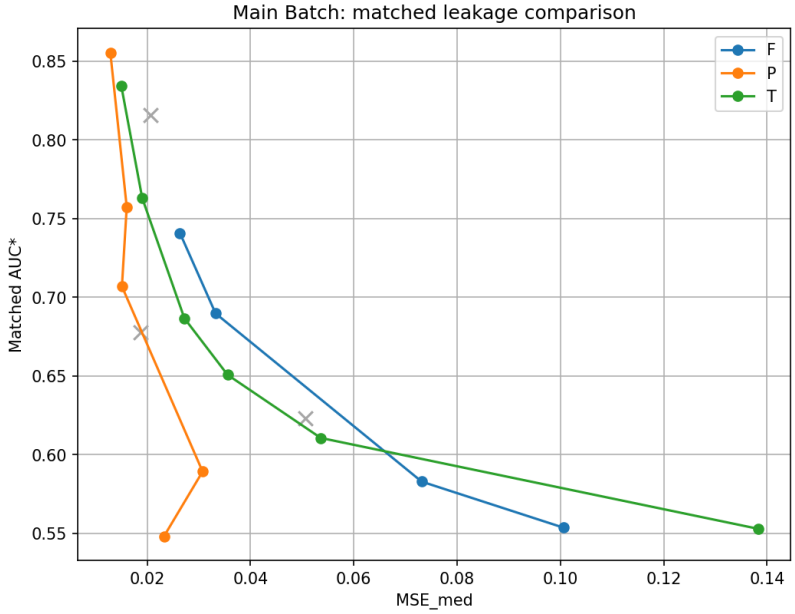

Figure 13. V1 sparse-sign main comparison at $\lambda_{\text{task}} = 1.0$.

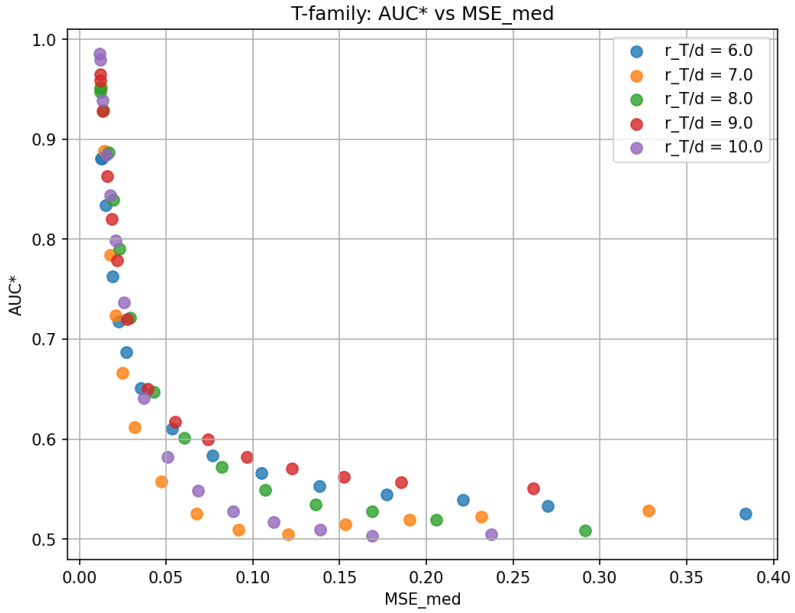

Figure 14. V1 T-family interpolation under sparse-sign sketches at $\lambda_{\text{task}} = 1.0$.

C.6. V1 Sparse Mechanisms at $\lambda_{\text{task}} = 0.1$

Figure 15 and Figure 16 report the sparse-sign main comparison and T-family interpolation for the V1 mechanisms under $\lambda_{\text{task}} = 0.1$.

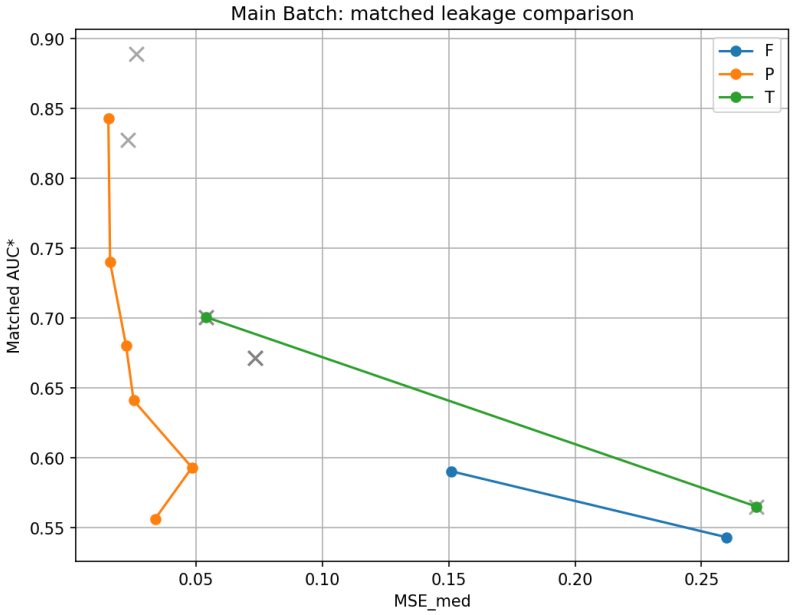

Figure 15. V1 sparse-sign main comparison at $\lambda_{\text{task}} = 0.1$.

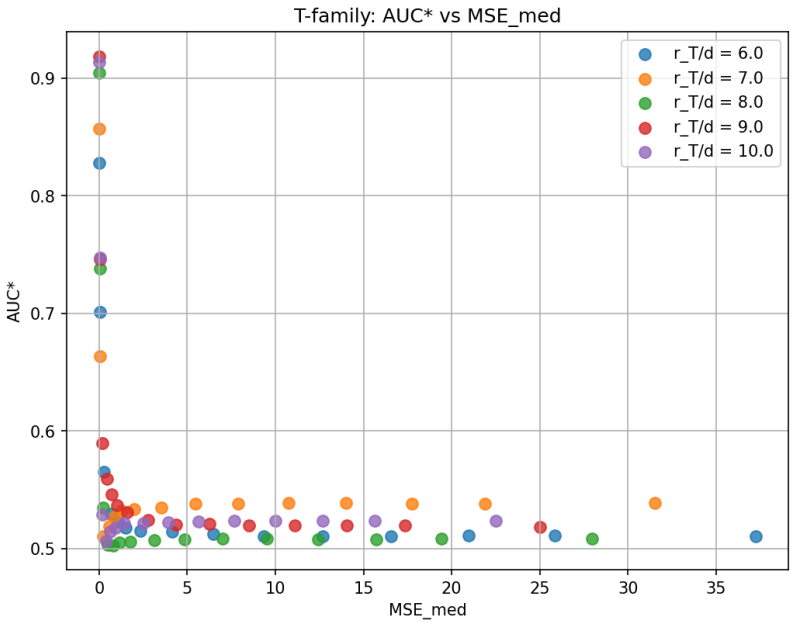

Figure 16. V1 T-family interpolation under sparse-sign sketches at $\lambda_{\text{task}} = 0.1$.

D. Diagnostics by Noise Multiplier

In exploring the T-family, tracking empirical leakage against the noise multiplier provides further clarity regarding the mechanism's sensitivity. As depicted in the diagnostic plots, increasing the stage-two noise multiplier fundamentally limits AUC and concurrently raises regression error. Figures 17–24 report the full set of diagnostic metrics across all mechanism families, sketch types, and task ridge parameters.

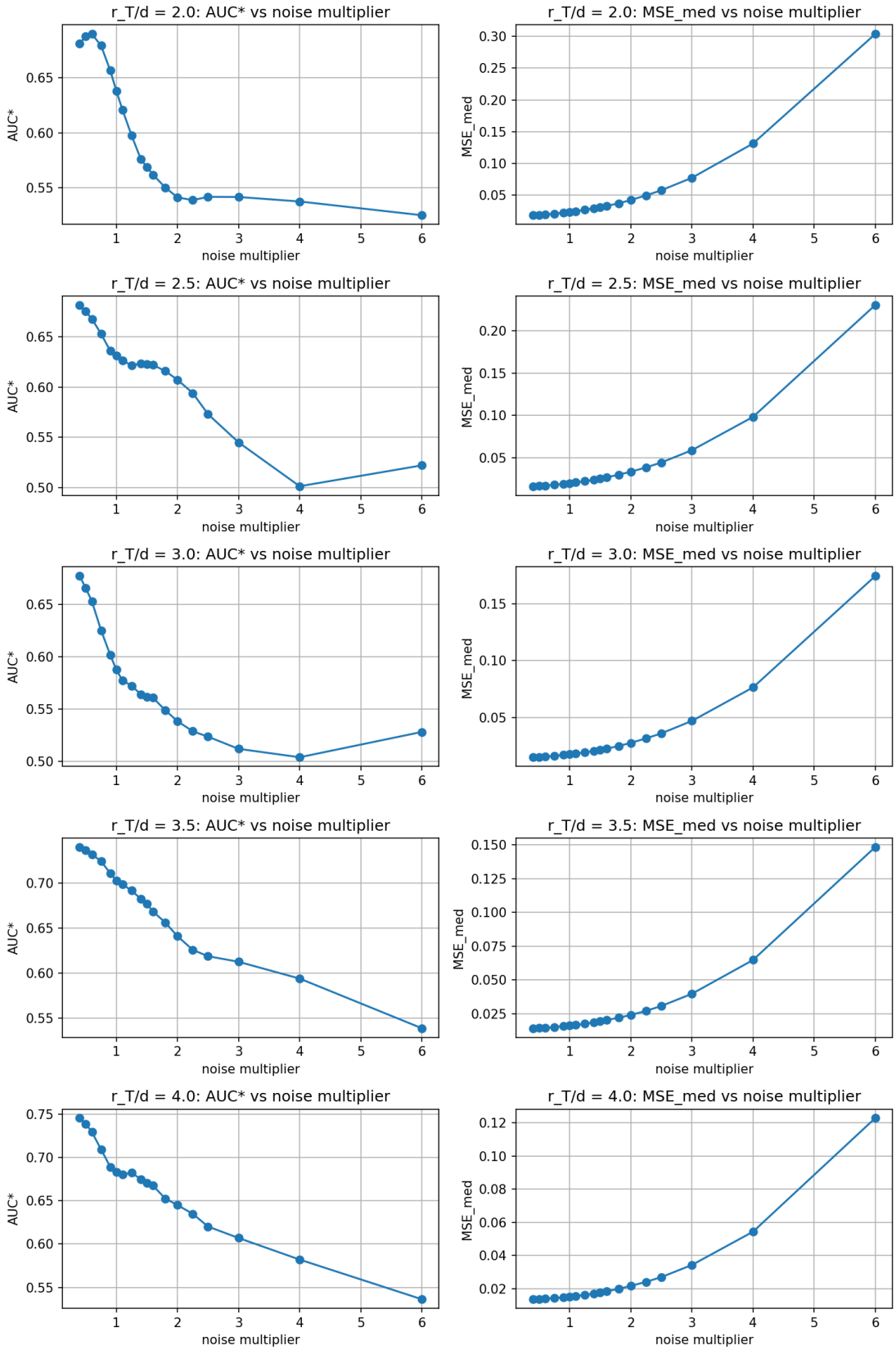

Figure 17. Diagnostic view of the V0 T-family under Gaussian sketching at $\lambda_{\text{task}} = 1.0$.

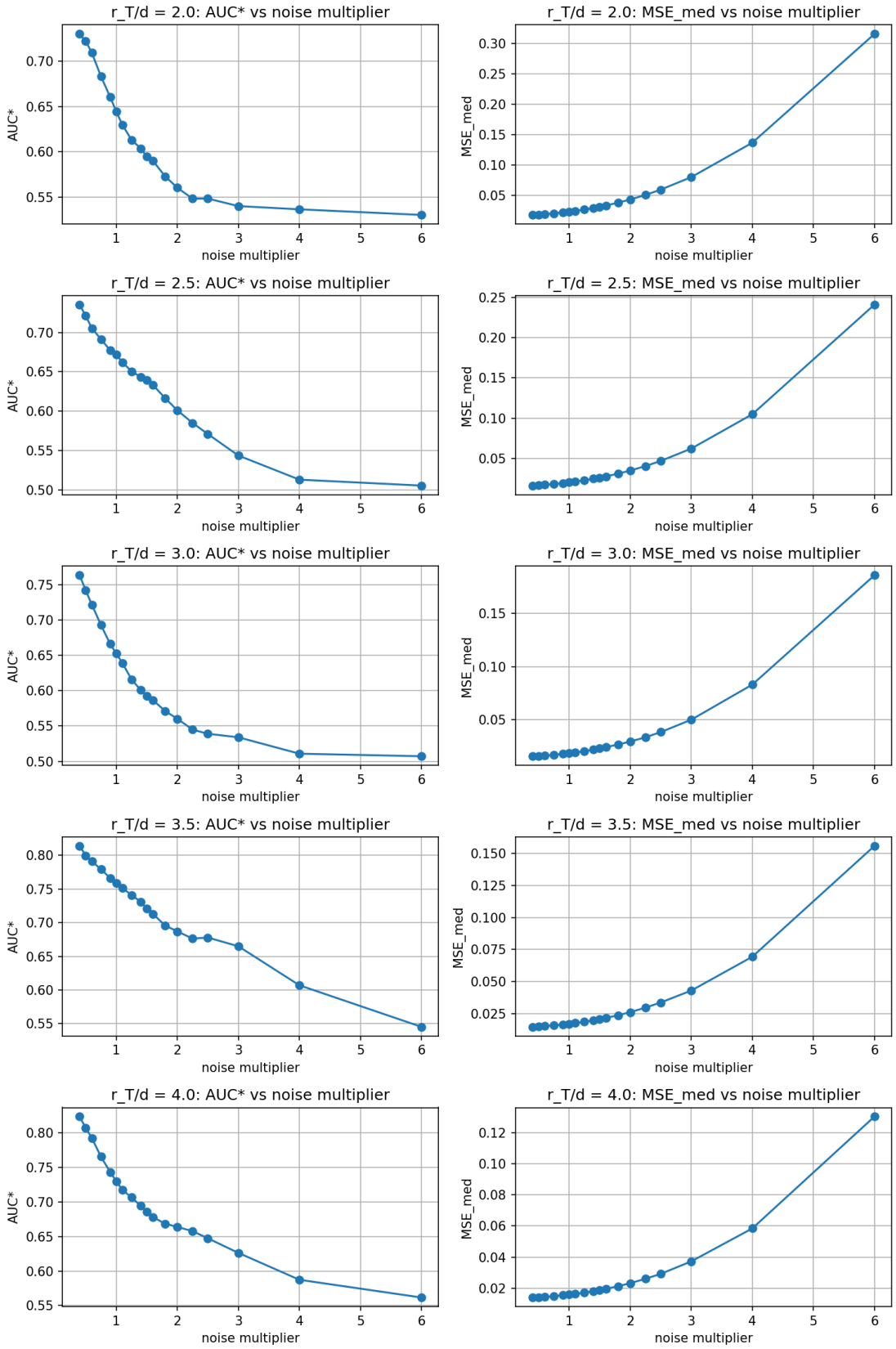

Figure 18. Diagnostic view of the V0 T-family under Gaussian sketching at $\lambda_{\text{task}} = 0.1$.

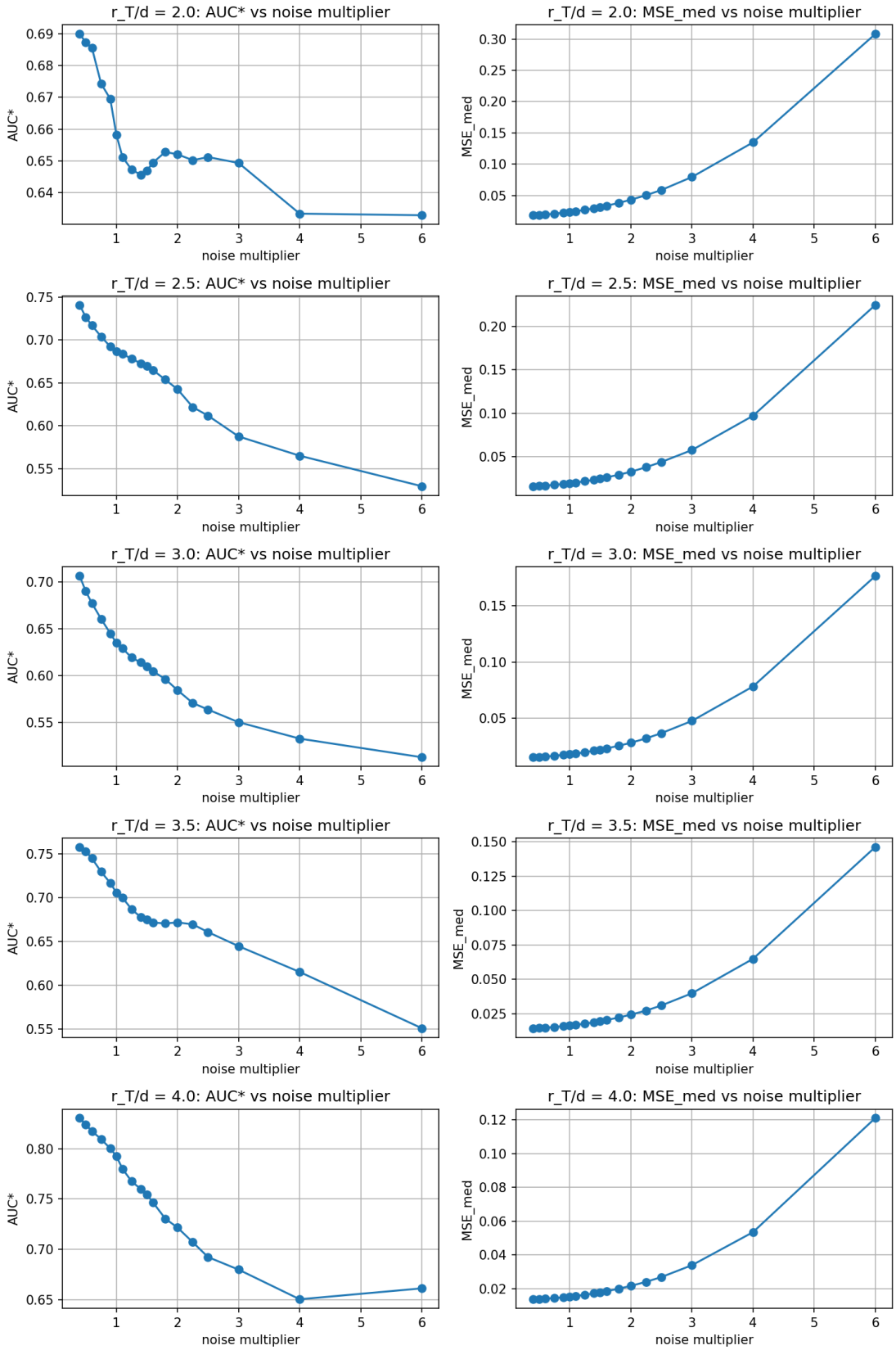

Figure 19. Diagnostic view of the V0 T-family under sparse-sign sketching at $\lambda_{\text{task}} = 1.0$.

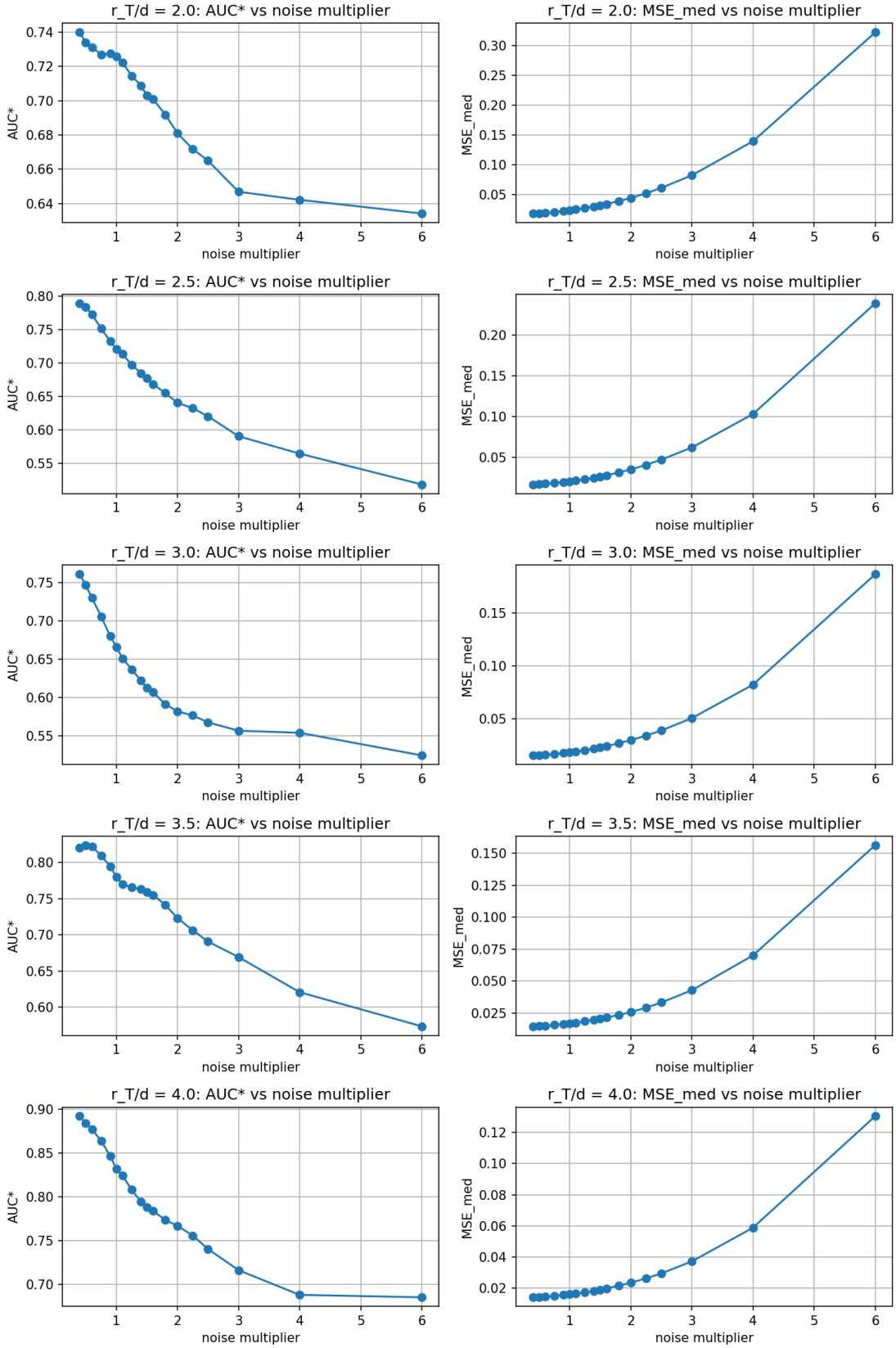

Figure 20. Diagnostic view of the V0 T-family under sparse-sign sketching at $\lambda_{\text{task}} = 0.1$.

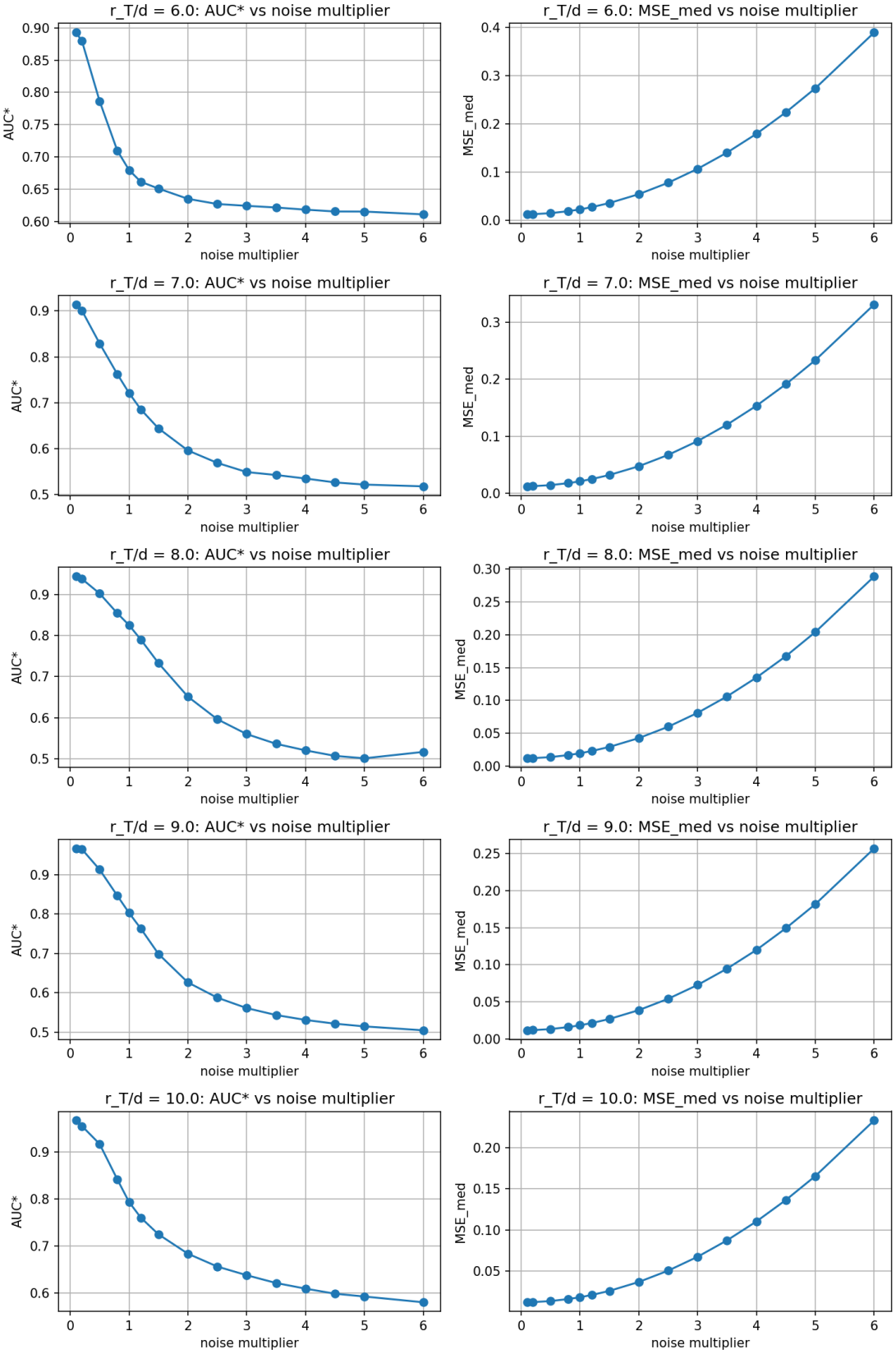

Figure 21. Diagnostic view of the V1 T-family under Gaussian sketching at $\lambda_{\text{task}} = 1.0$.

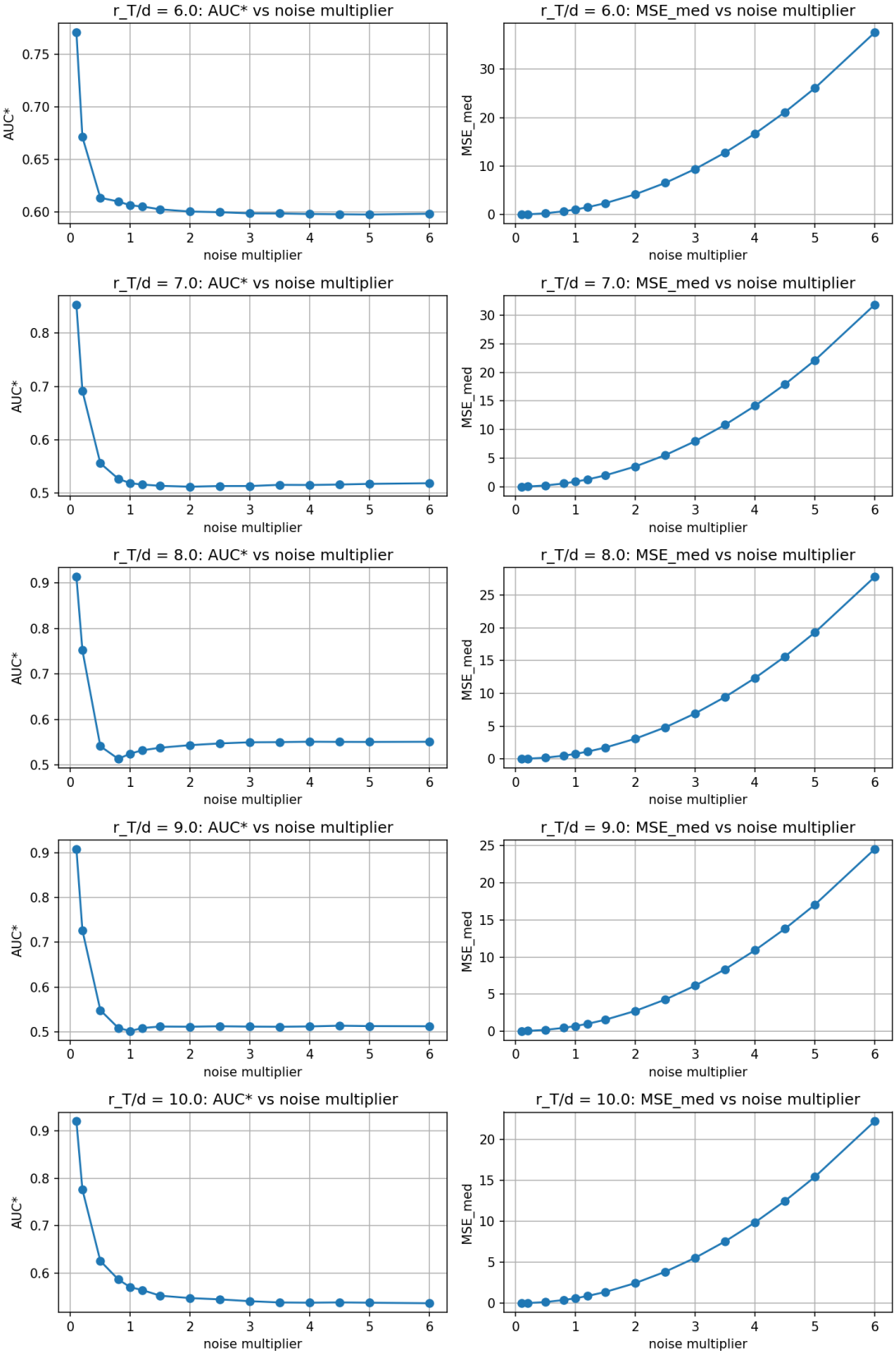

Figure 22. Diagnostic view of the V1 T-family under Gaussian sketching at $\lambda_{\text{task}} = 0.1$.

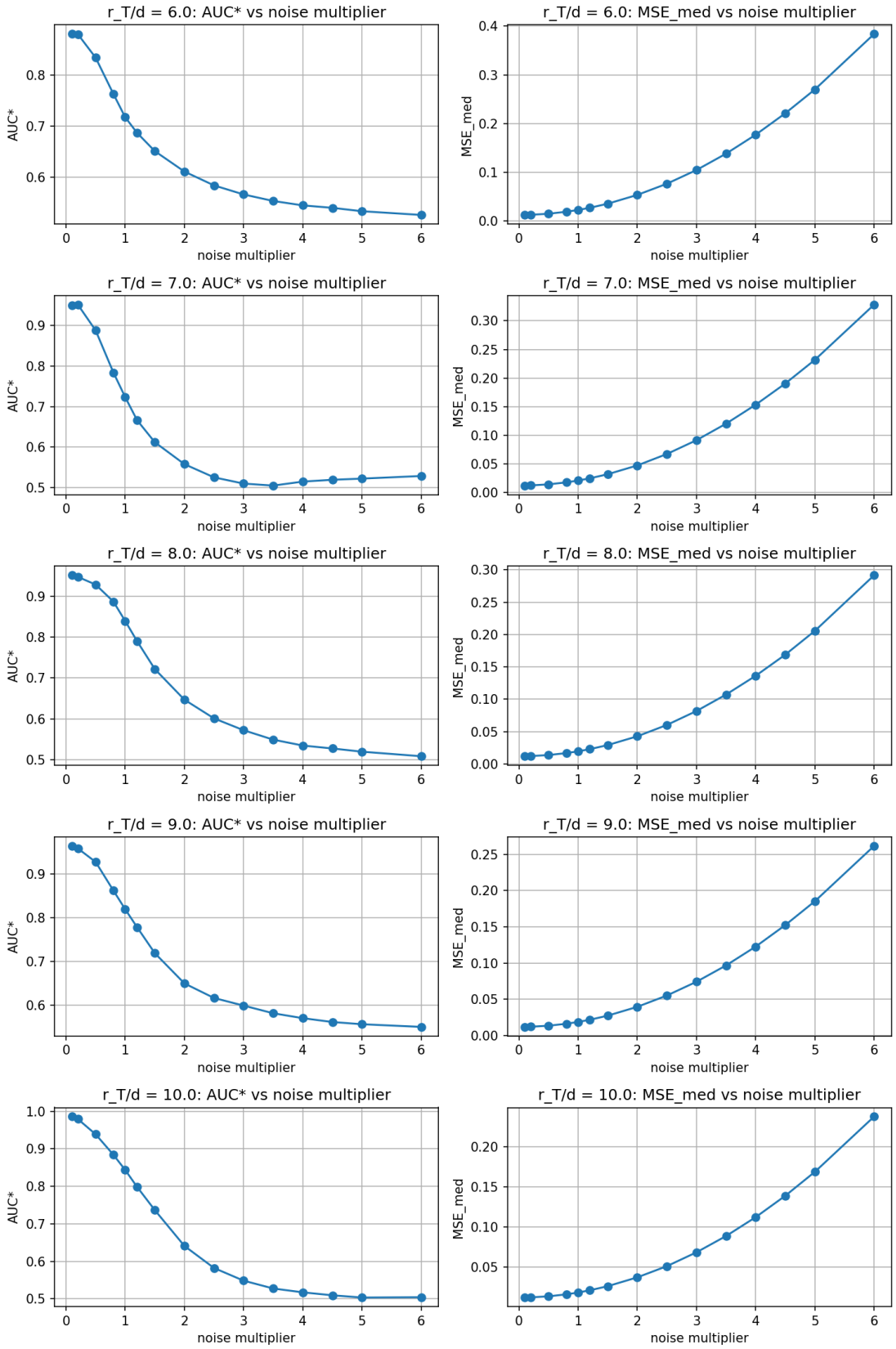

Figure 23. Diagnostic view of the V1 T-family under sparse-sign sketching at $\lambda_{\text{task}} = 1.0$.

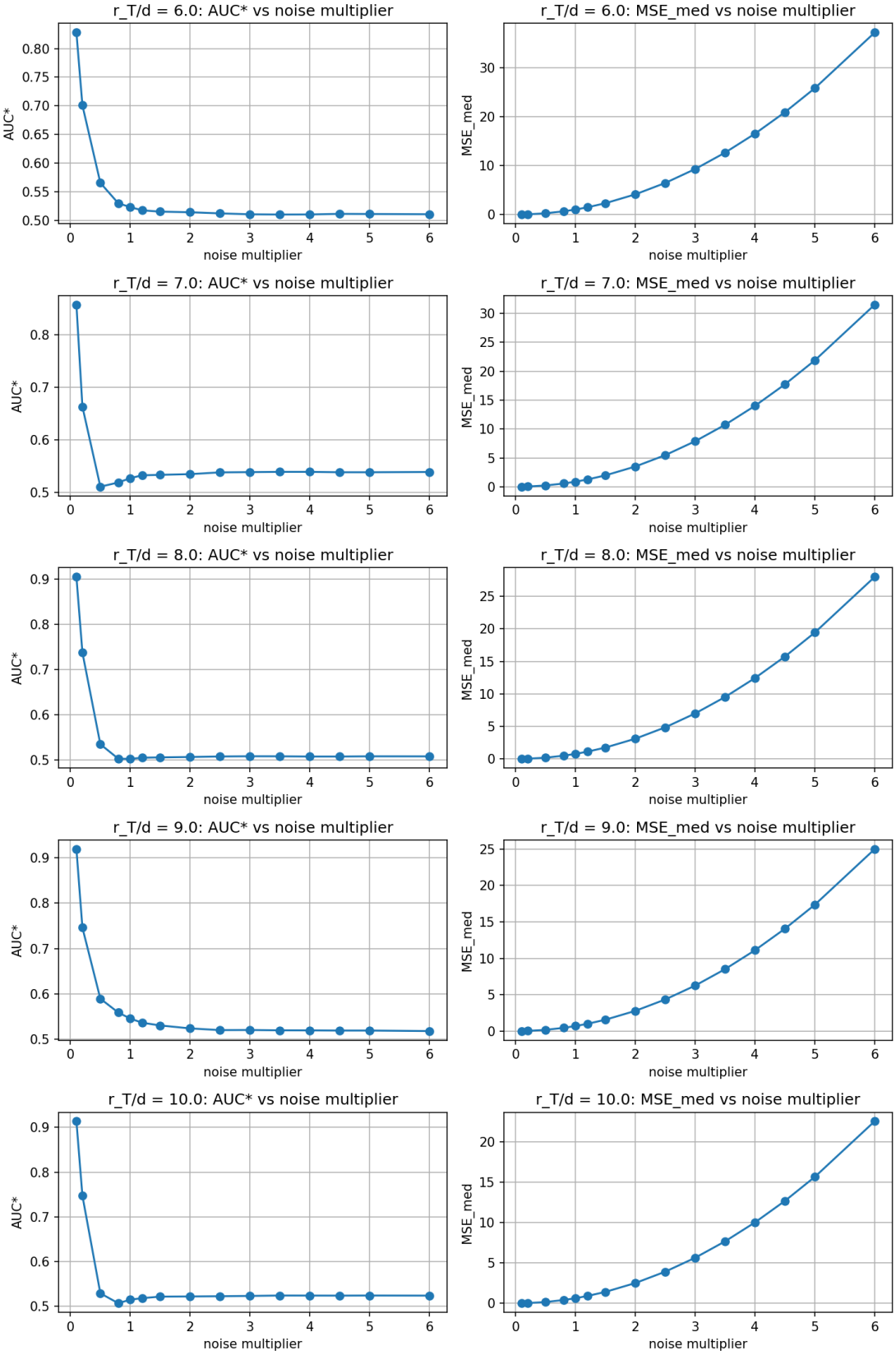

Figure 24. Diagnostic view of the V1 T-family under sparse-sign sketching at $\lambda_{\text{task}} = 0.1$.