LEARNING STABILIZER-TYPE CONSTRAINTS WITH ADAPTIVE PRODUCT MEASUREMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning structure in high-dimensional quantum systems from restricted measurements couples representation and acquisition. Under fresh-copy product-Pauli readout, we learn unknown stabilizer-type constraints from bitstrings and reuse their verified span as the adaptive policy state. This representation generates the coordinates for subsequent acquisition: a classical compiler forms a physical residual frame, packs compatible unresolved directions into the next product setting, and randomly completes the remaining axes. This representation-generating feedback loop uses no graph, code, generator list, or target-family prior; before any relation is verified, it is exactly uniform Random. Across 30–64-qubit chain, graph, and code-derived targets—including independently drawn graph and CSS cohorts and a mixed state with partial constraints—the same rule completes every fixed structured run and improves on Random across all tested physically organized suites. A layered theory identifies structural conditions under which different gains are possible: rare residual bottlenecks can change completion complexity; when Random is already polynomial, matching and joint capture yield repeated finite-budget gains; under an isotropic completion law, no stable directional advantage is available. Controlled interventions validate these mechanisms, and readout-noise experiments test verified feedback and targeted tail search.

1 INTRODUCTION

Learning quantum properties from limited measurements is an important statistical-learning problem spanning property prediction, adaptive tomography, and many-body experiments (Huang et al., 2020; Lange et al., 2023; Hu et al., 2025; Votto et al., 2026; Fischer et al., 2026). At scale, restricted local interfaces make measurement selection part of the learning problem. Under fresh-copy product-Pauli readout we therefore ask: *what representation should be learned from past outcomes to generate more informative future measurements?*

Learned representation and policy state. Our output is an initially unknown stabilizer-type relation system, rather than a state estimate, supplied-property estimator, or inference inside a supplied family model (Gross et al., 2010; Huang et al., 2020; Ouyang & Tomamichel, 2022; Huang et al., 2021; Hadfield, 2022). Relations are discovered jointly from outcome batches, independently verified, and stored in a span G_t. Thus G_t is both learned representation and policy state, generating the coordinates of the next action:

$$G_t \longrightarrow Q_{G_t} = G_t^{\perp}/G_t \longrightarrow \text{physical residual frame} \longrightarrow s_{t+1} \longrightarrow G_{t+1}.$$

We call this *representation-generating feedback*. Exact relations may be complete, as for a pure stabilizer target, or partial, as in mixed and non-stabilizer states.

Version space and physical action geometry. Pauli commutation defines the abstract residual space Q_G, but a product measurement acts on one physical realization. Algebraically equivalent quotient bases can change supports, local $X/Y/Z$ letters, conflicts, and co-exposed rank. The learned-row compiler therefore inherits a physical frame from verified rows; randQ and repR separately perturb quotient coordinates and physical lifts (Section 5.2).

Target-uninformed acquisition. The only inductive bias is stabilizer commutation and closure; generator identities, support patterns, graph/code labels, and target distributions are hidden. The policy is exactly uniform Random at $G = 0$; afterward it packs compatible unresolved directions and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

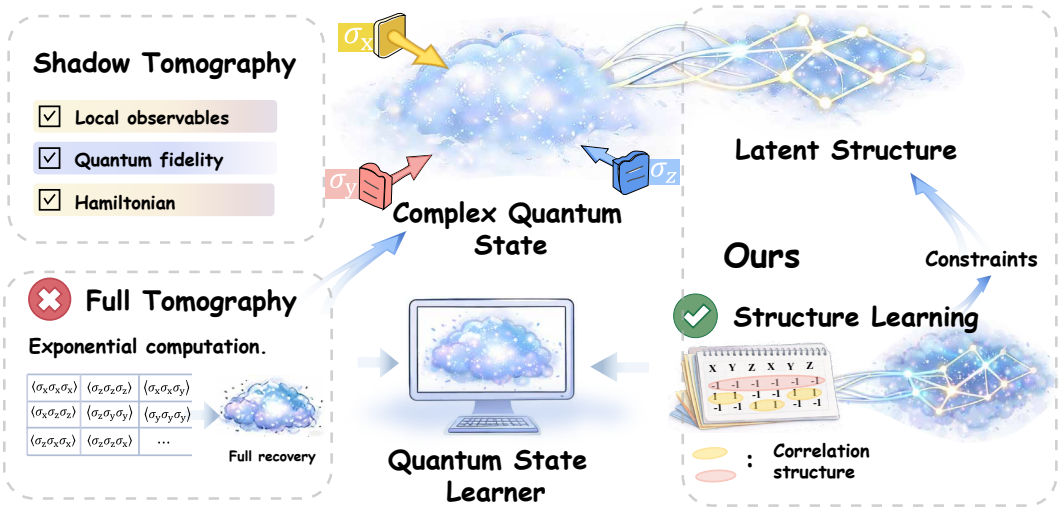


Figure 1: **One product-measurement interface, different learning objects.** Tomography reconstructs a state and shadows estimate queried properties. Constraint learning discovers unknown Pauli relations, stores their verified span, and compiles that span into coordinates for subsequent measurements and candidate search.

Table 1: Sequential learning interface under product-Pauli access.

Quantum object	Learning role	Operational meaning
Product setting	query / action	one local axis per qubit
Outcome batch	observations	repeated classical bitstrings
Verified span G	learned state	all accepted rules and their implications
Residual space Q_G	version space	legal relations not yet resolved
New independent rank	progress	newly resolved degrees of freedom

uniformly fills remaining axes. Figure 1 summarizes the two coupled interfaces: data are compressed into verified constraints, then compiled back into the next query.

All feedback computation is classical; under noise, only relations admitted on fresh data may change later actions.

Contributions. **(1) Representation-generating feedback.** Verified relations form both the learned structure and the persistent acquisition state. **(2) A target-uninformed physical residual compiler.** A polynomial-time learned-row compiler realizes $(G_t^{\perp} \cap V_A)/G_t$ in physical coordinates, packs compatible directions, and reverts exactly to Random at $G = 0$; coordinate, lift, order, and packing interventions isolate its operative choices. **(3) Mechanism theory and cross-structure evidence.** Analysis separates rare bottlenecks, local matching, joint capture, and isotropic regimes, with tests across fixed and randomized graph/check targets, partial mixed-state relations, and readout noise.

2 Problem Formulation: Learning Stabilizer-Type Constraints

We consider sequential structure learning for an unknown n-qubit state under repeated fresh-copy product-Pauli measurements. The target is its exact stabilizer-type relation space; each action chooses an X, Y, or Z readout axis on every qubit, and repeated execution returns a batch of classical bitstrings. Table 1 summarizes the learning interface.

The useful target need not be a reconstructed state or a preselected observable. For example, bits may vary individually while their XOR is fixed. Binary elimination can discover such a rule from one batch, and fresh data can verify it. The learner therefore turns outcomes into reusable constraints.

Only independent verified relations must be maintained; all their implications follow algebraically. Let $S(\rho)$ be the signed exact Pauli relations of the unknown state. They define a certified common eigenspace without reconstructing the state inside it and may have any rank $r_S \leq n$; a generic state

may have none. We learn this stabilizer-type relation space. Pure stabilizer states give the complete case $r_S = n$ and hence a full-recovery benchmark, while partial relations remain valid learned structure.

What remains after learning some constraints? Before choosing a new measurement, the learner can remove two kinds of redundancy: directions inconsistent with verified relations, and directions that differ only by information already known. In the binary Pauli representation, compatibility with the known relations means commutation. Let $G^\perp$ denote the commuting directions. Identifying directions that differ by an element of G produces the unresolved space—the analogue of a version space after removing already solved distinctions—

$$Q_G = G^\perp/G, \qquad d = n - \mathrm{rank}(G), \qquad |Q_G| = 4^d. \tag{1}$$

Let $S = S(\rho)$ denote the target's exact stabilizer-type relation space. Every unseen true relation commutes with G, so it lies in this unresolved space. Equation 1 is an exact feasible set: it removes inconsistent or already resolved directions. A posterior weighting over the remaining legal directions would require additional statistical state. The implementation represents it with a linear-size coordinate basis rather than enumerating 4^d classes.

The unknown remaining relations $L_G = S/G$ have dimension $\ell = r_S - \mathrm{rank}(G)$; in the complete pure-stabilizer case, $\ell = d$. Let M_s be the Pauli span generated by the local axes of product setting s, and let π_G denote projection into Q_G. The measurable residual slice is $E_G(s) = \pi_G(M_s \cap G^\perp)$. The learning reward is the number of independent true relations inside that slice:

$$\Delta_G(s) = \dim(E_G(s) \cap L_G). \tag{2}$$

Under complete extraction, every independent true relation measurable in the queried setting is admitted, so this intersection becomes the update; a restricted extractor may recover less. Feasibility alone does not rank legal queries: exact lookahead requires a posterior over legal completions. Under the symmetry-preserving maximum-entropy completion law, every product setting has the same one-step gain distribution (Proposition 5). Compatible packing is also NP-hard; we instead use verified physical structure while retaining random exploration.

What is the policy allowed to trust? Only admitted relations affect the next action. Conditional on their correctness, residualization and query compilation are exact; finite samples and noise determine which updates are admitted. This separates statistical validation from acquisition (Appendices E and G).

3 Method: Representation-Generating Feedback

3.1 Learned-row residual compiler

The residual quotient specifies which directions remain unresolved, while hardware requires one concrete local axis per qubit. The production compiler therefore fixes a physical realization before packing. It computes active-site incidence from all maintained verified physical rows (local rows followed by global rows), sorts sites by decreasing incidence with physical-index tie breaking, then scans the rows in maintained order and retains the first span-increasing rows without rebasing. Rank-aware tableau completion yields logical X/Z outputs that are embedded back as the proposal frame; production applies no random quotient rebasing or within-coset lift randomization. It then greedily combines locally compatible proposals and fills unassigned axes uniformly at random.

Random proposal order and completion preserve exploration. We use a random-order greedy scan rather than NP-hard maximum compatible packing (Appendix A.2). At $G = 0$ the rule is exactly uniform Random; for m proposals on active support A, the scan costs $O(m|A|)$ axis comparisons. No instance-level label is supplied, and all computation beyond quantum sampling is classical.

Algorithm 1 Production loop with deterministic learned-row compilation

Initialize the verified relation span $G_0 \leftarrow \{0\}$
for $t = 0, 1, \ldots$ until recovery or the stopping rule **do**
  $A \leftarrow \mathrm{supp}(G_t)$, $m \leftarrow |A|$; let R_t be the maintained ordered verified physical rows
  Set $d_i \leftarrow |\{g \in R_t : g \text{ acts nontrivially on } i\}|$, let π sort A by $(-d_i, i)$, and retain the first span-increasing rows of R_t as $g_1, \ldots, g_r$
  if $m = 0$ or $r = m$ **then**
    $P_t \leftarrow \varnothing$
  else
    $T \leftarrow$ StabilizerComplete$(g_1|_\pi, \ldots, g_r|_\pi)$ with underconstrained completion enabled
    $P_t \leftarrow \{\mathrm{Embed}_\pi(T.X_j), \mathrm{Embed}_\pi(T.Z_j) : j = r, \ldots, m-1\}$
  end if
  if $P_t = \varnothing$ **then**
    Draw s_t uniformly from $\{X, Y, Z\}^n$
  else
    Uniformly permute P_t; greedily retain every proposal whose nonidentity letters agree with the partial setting
    Fill every unassigned axis independently and uniformly from $\{X, Y, Z\}$ to obtain s_t
  end if
  Collect a discovery batch; extract measurable candidates; freeze them and confirm parity means/signs on fresh outcomes
  $G_{t+1} \leftarrow \mathrm{span}(G_t \cup \{\text{independent accepted candidates}\})$
end for
return the learned relations and signs

Production pins `stim 1.16.0` (release commit `e2fc1ec`) with redundant rows disabled and underconstrained completion. Row and site orders are algorithmic; Appendix A.2 gives the source-level rule used by the GHZ and Ring certificates. The returned $2(m-r)$ logical rows are used exactly as emitted.

The embedded rows represent $(G_t^\perp \cap V_A)/G_t$. For $G_t = \langle Z_1 Z_2 \rangle$, one valid pair is $Z_1, X_1 X_2$; random packing chooses which conflicting pattern to exploit. If the active quotient produces no proposal, the production loop returns to a fully uniform product setting instead of reshaping the unexplored support with a full-space fallback. Algorithm 1 is the acquisition rule used by the headline experiments and by the production theory below. Appendices A and G give extraction, validation, and the optional noisy residual-tail protocol.

4 Theory: When Does Constraint Reuse Help?

The central active-learning question is when a learned representation improves future acquisition. Algorithm 1 fixes axes inherited from verified relations and leaves the others random, yielding three regimes: stepped spectra with rare residual bottlenecks can change completion scale, structured residuals can improve finite-budget exposure, and isotropic uncertainty offers no stable direction. We analyze noiseless complete extraction, so every exposed true relation is admitted and $G \subseteq S$. Let $g_\pi(G) = \mathbb{E}_\pi[\Delta_G(s) \mid G]$ denote expected newly exposed rank; restricted extraction replaces exposed gain by admitted gain.

4.1 A common progress measure and two reuse mechanisms

Fix a compatible subset C of the proposals actually constructed by the algorithm. Let ρ be the probability that the greedy scan retains all of C. Conditional on retention, their axes are fixed. If a representative of an unknown target class is a product of these proposals followed by r letters outside the active support, those letters remain independent uniform choices. Therefore

$$\Pr_A(\text{target class exposed} \mid G) \geq \rho\, 3^{-r}. \tag{3}$$

Multiplying by a known relation does not change which new information is learned. Thus the relevant cost is the number of remaining random matches, not the weight of a chosen target generator. For example, fixing two letters of a three-site pattern leaves one random match: the success probability is at least $\rho/3$, compared with $1/27$ for Random to match that particular pattern. Appendix B bounds ρ directly from the proposals that conflict with C. The proposal rule uses only G.

Packing can also help through products. If the span of C contains ν independent target classes modulo G, retaining C exposes all of them, so

$$g_A(G) \geq \rho\nu. \tag{4}$$

For instance, compatible proposals X_1X_2 and X_2X_3 expose their product X_1X_3, which may be a true relation even when neither proposal is. This explains why counting retained proposals alone

misses the benefit. Equations 3–4 are the two consequences of the same algorithmic event: retaining a useful compatible subset. They require neither independent retention events nor a basis of hidden target generators.

Interpretation. Matching compression reduces remaining random axis matches; joint capture exposes more independent relations per query, while random completion preserves exploration. Fixing axes can also displace useful random completion, so net gain requires captured rank to exceed that loss; frozen-G replay measures the resulting $g_A(G) - g_R(G)$ (Appendix B).

4.2 Stepped spectra: when bottlenecks change complexity

A completion-scale separation arises when acquisition has a stepped exposure spectrum: an accessible structural skeleton can be accumulated at polynomial cost, but completion requires crossing a residual coset whose total exposure mass under uniform Random is exponentially small. Let $E_n < S_n$ be a codimension-one skeleton, $\mathcal{C}_n = h_n + E_n$ its remaining coset, and $d_E(G) = \dim E_n - \dim(E_n \cap G)$. Let $m(G)$ denote the number of independent missing skeleton classes with compatible cores in the production proposal family at a production-reachable correct G. Suppose that at every such G with $d_E(G) > 0$,

$$m(G) \geq c_* d_E(G),$$

that each such core is retained with probability at least ρ_*, and that its representative needs at most r_* additional random axis matches. At $G = E_n$, suppose one representative of $\mathcal{C}_n$ has a core retained with probability at least ρ_B and needs at most r_B random matches. Finally define the Random exposure mass of the necessary terminal coset by

$$B_n = \sum_{P \in \mathcal{C}_n} 3^{-\operatorname{wt}(P)}.$$

These are certificates of the physical proposal frame, not target information supplied to the learner.

Theorem 1 (Structural spectral-bottleneck separation)**.** *Under noiseless complete extraction and the certificates above,*

$$\mathbb{E}T_{\text{prod}} \leq \frac{1 + \log d_E(0)}{c_* \rho_* 3^{-r_*}} + (\rho_B 3^{-r_B})^{-1}, \qquad \mathbb{E}T_{\text{Random}} \geq B_n^{-1}.$$

Hence inverse-polynomial production certificates together with $B_n = \exp[-\Omega(n)]$ give a polynomial–exponential separation in expected settings. Appendix C.3 proves the more general admission-robust form.

GHZ instantiation. For GHZ_n, the easy skeleton is $E = \langle Z_i Z_{i+1} \rangle$, while the remaining coset consists of full-support X/Y strings. The learned-row compiler contracts the learned Z-blocks geometrically, so the skeleton certificates in Theorem 1 are polynomial. At the terminal skeleton, production selects the block-wide target direction with probability $1/2$, whereas uniform Random has total terminal exposure mass

$$p_A = \frac{1}{2}, \qquad B_n = p_R = \frac{2^{n-1}}{3^n}. \tag{5}$$

Equation 5 gives the terminal hazard gap. The opposite branch, where the global class arrives first, also has polynomial drift. Appendix C.3 gives the exact learned-row tableau and block calculations.

Corollary 1 (GHZ production instantiation)**.** *For the noiseless complete-extraction loop on* GHZ_n*, starting from $G_0 = 0$, Learned-row completes in expected polynomially many settings, whereas uniform Random satisfies*

$$\mathbb{E}T_R \geq 2(3/2)^n.$$

Thus Corollary 1 is a closed-form production instantiation of the structural separation in Theorem 1, rather than the definition of the regime itself.

A non-GHZ stepped-spectrum witness. The same scale-changing principle is not specific to GHZ algebra. Let $K_i = Z_{i-1} X_i Z_{i+1}$ be the generators of the cycle graph, let E_n be their even-parity subgroup, and add $H_n = Z^{\otimes n}$. The local sector E_n has an independent weight-four generating set $K_i K_{i+1} = ZYYZ$, so uniform Random itself reaches rank $n-1$ in $O(\log n)$ expected settings under complete extraction. The gap appears only in the final residual direction: once $\operatorname{rank} G = n-1$,

$Q_G = G^\perp/G$ has only three nonzero classes. A simple G-only quotient query reaches one with constant probability, whereas uniform Random must hit an exponentially sparse global coset. This construction uses a separate quotient-feedback rule to isolate the stepped-spectrum mechanism; Appendix C.3 proves:

Proposition 1 (A non-GHZ local-plus-global separation)**.** *For $n \geq 5$ and the full-rank target $S_n = \mathrm{span}(E_n, H_n)$ under noiseless complete extraction, consider the target-uninformed quotient-feedback rule that uses uniform Random while $\mathrm{rank}\, G < n-1$; if this stage reaches rank n it stops, and otherwise at rank $n-1$ it uniformly queries a representative of one of the three nonzero classes of $G^\perp/G$. Then*

$$\mathbb{E}T_{\mathrm{qfb}} \leq \left\lceil \frac{\log(n-1)}{\log(81/80)} \right\rceil + 84 = O(\log n), \qquad \mathbb{E}T_R \geq \tfrac{1}{4}(3/\lambda)^n, \quad \lambda = 2.931142\ldots < 3.$$

Proposition 1 uses an easy bounded-weight Ring-derived skeleton that Random already learns efficiently. The asymptotic separation appears after this local structure leaves a rare global residual class, extending the spectral-bottleneck phenomenon beyond GHZ. The proposition isolates quotient feedback on this construction; ordinary Ring production behavior is analyzed through local certificates and end-to-end experiments.

4.3 Accessible structure: when the gain is finite-budget

Reuse can improve learning even without an exponentially rare final class. If a complete generating basis has bounded weight, Random already completes in polynomially many settings, so a polynomial–exponential separation is mathematically unavailable. The relevant question is instead whether feedback strictly improves exposure and whether those gains accumulate along its realized trajectory. The same two one-query calculations explain the graph and check examples.

Local matching. For a graph corridor with generators $K_i = Z_{i-1}X_iZ_{i+1}$ and learned rows $G = \langle K_1, \ldots, K_r\rangle$, $r \geq 2$, the compiler returns two disjoint conflicting proposal pairs. Each pair is resolved by a fair random order. The two next-generator classes have exposure probabilities $1/6$ and $1/18$: the first needs one random boundary match after its proposal is retained; the second needs two. Their sum gives $g_A^W = 2/9$. Counting all Random representatives of these classes and their product gives

$$g_A^W - g_R^W = \frac{1}{18} + \frac{20}{243}(2/3)^r - \frac{(-1)^r}{486\,3^r} > 0.$$

This is an exact result for the corridor family, derived from its generated proposals (Proposition 4); $r = 1$ gives the positive surplus $1/36$ by direct enumeration.

Joint check capture. For the three explicit BB5-30 spans of ranks $1, 3, 5$ listed in Appendix C.5, express the independent weight-five X checks as products of the actual proposals. Random-order blocker counts and free-axis matching give expected-rank lower bounds $251/1296$, $49/144$, and $121/288$. Counting Random representatives gives the corresponding gain-ratio lower bounds $2.3758\ldots$, $4.9813\ldots$, and $4.9571\ldots$. These are finite calculations on the stated spans. The pack-cap experiments test the resulting joint benefit separately from single-pattern matching.

How one-query gains accumulate. Write $A_t^\pi = \mathrm{rank}(G_{t+1}^\pi) - \mathrm{rank}(G_t^\pi)$ for the rank actually admitted after query t. For a common initial span, summing these updates gives

$$\mathrm{IRD}_H(R) - \mathrm{IRD}_H(A) = \frac{1}{n}\sum_{t=0}^{H-1}(H-t)\big(\mathbb{E}A_t^A - \mathbb{E}A_t^R\big). \tag{6}$$

Here IRD sums normalized rank deficits after the first H queries. Equation 6 gives earlier gains greater weight because they reduce more subsequent deficits. The expectations follow each policy's own trajectory: a positive same-G surplus alone is insufficient. The identity holds directly for the implemented updates, including finite-sample rejection; the measured rollouts evaluate its trajectory-level terms (Appendix C.1).

Thus Theorem 1 gives the general scale-changing condition, with GHZ as a closed-form production instantiation; the Ring–global construction shows the same stepped-spectrum mechanism beyond GHZ algebra under a separate quotient-feedback rule. Ordinary Ring and BB5 provide local production certificates, and rollouts measure how those gains accumulate along full trajectories. All

proposals are computed from G alone; target relations enter only the analysis and offline audits. Finally, reuse needs physical structure worth preserving. Under the symmetry-preserving maximum-entropy completion law conditioned only on G, all product settings have the same gain distribution (Appendix D), consistent with the lack of stable gain on Random Clifford.

Learning takeaway. The theory organizes feedback value into three regimes: stepped spectra can change completion scale; accessible local/check structure can improve finite-budget progress; and isotropic uncertainty leaves no preferred G-only direction. The experiments test these predictions through full trajectories and interventions that hold the learned state fixed.

5 Experiments: Breadth, Mechanisms, and Noise

We sample stabilizer measurements with `stim` (Gidney, 2021). *Learned-row (Adaptive)* denotes Algorithm 1; targets are used only for evaluation. Our primary resource is the number of distinct product-Pauli settings. Unless stated otherwise, every policy uses 1024 discovery plus 128 fresh confirmation copies per setting, so fewer settings means fewer copies (57,600 at the 50-setting cap). Each fixed instance has 50 runs. Completed policies report mean settings $\pm$ SD; censored policies report completion count. Cohorts use the lower-is-better deficit $D_{50} = \sum_{t=1}^{50}(1 - r_t/n)$ with paired intervals. Full resource and access accounting is in Appendices E and H.

5.1 RQ1: Does one rule transfer across structures?

Table 2 reports the fixed-target breadth comparison.

Table 2: **End-to-end breadth across reusable stabilizer organizations.** Fifteen fixed instances span 12 constructions with 50 runs each. Entries are mean settings $\pm$ SD for 50/50 completion, otherwise completion count at the horizon; bold marks the fastest completed result. Learned-row is fastest on 11/15 and alone completes all 750 runs. Randomized-cohort intervals and controls are in Appendix F.9.

Target breadth: quotient-coordinate construction

Family	Hidden target	Random	RREF/free-column	**Learned-row (Adaptive)**
Stepped spectrum	GHZ-32	$0/50$	12.76 ± 2.80	$\mathbf{12.04 \pm 3.08}$
	$\mathrm{GHZ}_8^{\otimes 4}$	$10/50$	14.38 ± 4.17	$\mathbf{12.14 \pm 2.57}$
Local graph	Ring-32	21.52 ± 4.82	16.98 ± 4.06	$\mathbf{14.64 \pm 3.45}$
	Ring-64	24.76 ± 4.16	19.80 ± 3.74	$\mathbf{16.34 \pm 3.58}$
	Open-chain-64	24.22 ± 4.02	$20/50$	$\mathbf{16.86 \pm 3.03}$
	Range-2-chain-64	27.54 ± 4.99	$46/50$	$\mathbf{19.90 \pm 3.03}$
	Range-2-ring-64	27.38 ± 4.13	$12/50$	$\mathbf{20.64 \pm 3.17}$
	Irregular-chain-64 I	25.92 ± 4.68	$0/50$	$\mathbf{18.20 \pm 3.85}$
	Irregular-chain-64 II	24.80 ± 4.59	$1/50$	$\mathbf{17.82 \pm 2.46}$
	Irregular-chain-64 III	24.54 ± 4.47	$19/50$	$\mathbf{18.28 \pm 4.39}$
	Grid-8×8	31.80 ± 4.68	19.78 ± 3.43	$\mathbf{19.20 \pm 2.92}$
Code/check	BB5-30	29.82 ± 4.76	$\mathbf{14.82 \pm 4.39}$	17.24 ± 2.85
	BB5-48	$42/50$	$\mathbf{16.30 \pm 3.21}$	19.46 ± 3.18
	Toric-4×8	31.28 ± 3.56	$\mathbf{17.44 \pm 3.16}$	18.24 ± 3.24
	Local-HGP-64	$48/50$	$\mathbf{16.40 \pm 2.98}$	19.02 ± 3.82

RREF/free-column wins all four code/check rows but fails on many 64-qubit graph targets; Learned-row alone completes all 750 and never underperforms Random on a fixed structured row.

What the breadth establishes. One label-free rule covers stepped, local-graph, and check structure; the theory maps these to one scale-changing and two finite-budget mechanisms (Appendix F.1).

Figure 2 gives trajectories for six representative rows. Two preregistered-by-seed cohorts draw 12 new 64-qubit targets each: connected Erdős–Rényi graph states of requested mean degree 4, 6, or 8, with no fixed backbone, and sparse HGP/CSS states with uniformly placed weight-3/4 classical checks. Each target receives 20 runs per policy. Learned-row lowers D_{50} on all 24; target-balanced effects are -10.149 (95% CI $[-11.821, -8.437]$) and -4.682 ($[-5.280, -4.081]$), with completion $200/240$ versus $73/240$ and $240/240$ versus $231/240$. The degree-8 graph stratum is a visible boundary: Learned-row remains ahead but completes only $42/80$ within 50 settings. Its four targets

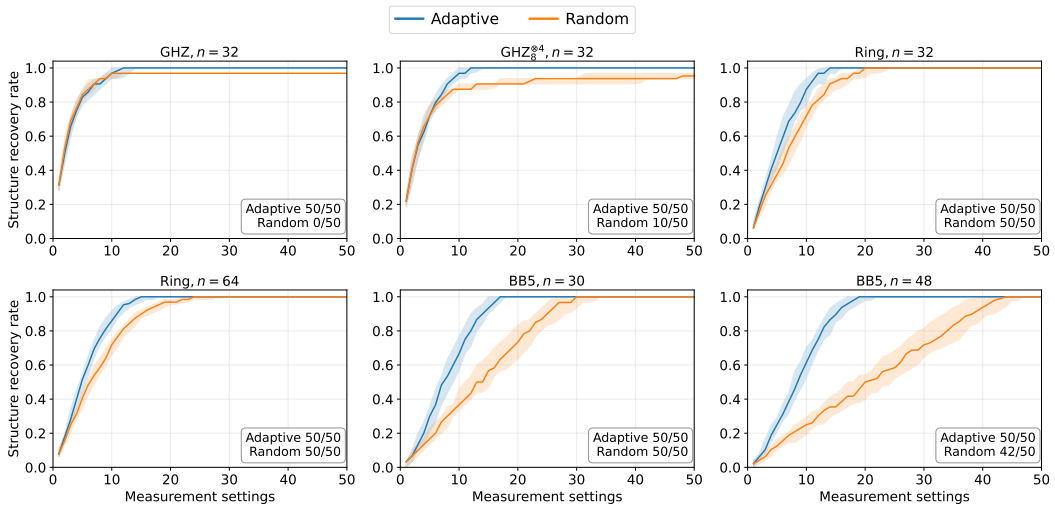


Figure 2: **Representative end-to-end trajectories under one interface.** Panels cover stepped-spectrum GHZ/GHZ-block, local Ring-32/64, and code-derived BB5-30/48 targets. Curves are median normalized true rank with IQR bands; insets report completion among 50 runs. Adaptive completes all six.

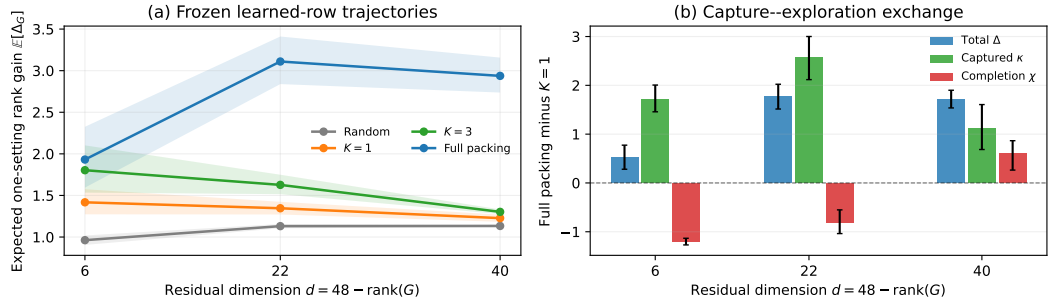


Figure 3: **Changing exploitation while holding learned information fixed.** BB5-48 frozen-G replay splits full-minus-$K = 1$ gain into capture κ and completion χ (10 origins/rank, 500 replays/origin).

are descriptive; population inference would require more target draws. Both cohorts are stress tests under the sampled graph/CSS distributions (Appendix F.2).

5.2 RQ2: What causes the gain?

Is the gain just a prior or generic feedback? These controls keep access and extraction fixed. Mixed Random's 10% mass on three global same-axis settings helps BB5-30 (22.66 versus Random's 29.82) but not Ring-32 (22.14 versus 21.52). Reuse-G points its identical verified input toward resolved relations and completes $0/50$ runs on both; the quotient policy completes them in 17.24 and 14.64 settings. Thus neither a fixed prior nor feedback alone explains the gain (Appendix F.9).

Coordinates, representatives, and order. The quotient fixes unresolved information while its physical realization controls product-measurement compatibility. randQ holds the learned span G and the abstract quotient Q_A fixed but applies a random invertible rebasing before packing; its degradation therefore isolates a failure of measurement equivalence despite algebraic equivalence. repR keeps the learned quotient coordinates fixed and perturbs only admissible physical lifts, and it tracks production closely. RREF/free-column supplies a separate coordinate compiler; WeightSort and conflict-degree ordering alter traversal of the production proposal geometry. Together these interventions locate the acquisition gain in the learned physical residual frame, co-exposure, and retained exploration (Appendices F.8 and D).

Table 3: **The extractor changes closed-loop recovery.** BB5-30 with 1% readout flips, 50 runs/policy, 50 settings; trajectories are in Appendix G.

Window W	Candidates / setting	Adaptive complete	Random complete
$[2, 5]$	174,406	8/50	0/50
$[6, 8]$	8,482,500	25/50	0/50
$[2, 8]$	8,656,906	48/50	0/50

Figure 3 removes trajectory shift by replaying acquisition from the same learned spans: full packing beats $K = 1$ at all three stages because captured structure repays displaced completion (Appendix F.9).

5.3 RQ3: Does verified feedback survive incomplete extraction?

Responsibility 1: extraction shapes future acquisition. Only verified relations enter G_t. Under independent 1% readout flips on BB5-30, both policies share fresh confirmation. Because unknown noisy-parity discovery is LPN-type hard (Blum et al., 2003), we use a family-agnostic bounded-support extractor with $O(n^{w_{\max}})$ candidates; the acquisition compiler remains polynomial. Table 3 compares $W = [2, 5], [6, 8], [2, 8]$ at 1024 copies.

Unseen terminal relations remain physically exposed but often lie outside the search window, so the plateau tracks extractor coverage.

Responsibility 2: learned structure narrows tail search. An optional late residual-verification routine bypasses ambient support enumeration by packing up to 16 random quotient combinations: low-weight recovery rises $8/50 \to 29/50$ while late scoring falls to 1.22 candidates per setting. Early scoring still dominates before useful verified structure accumulates (Appendix G.3).

Boundary, partial structure, and classical cost. On 20 paired Random-Clifford-32 targets, Learned-row and Random have no decisive ordering: D_{50} is 20.70 versus 22.26 (difference -1.56, 95% CI $[-3.35, 0.20]$), while completion is $10/20$ versus $14/20$. The interval includes zero and Random completes more runs within the horizon, consistent with the predicted isotropic boundary (Appendix F.3). The representation also applies to partial exact relation spaces: on a 32-qubit mixed target with a hidden rank-16 Ring relation space, both policies recover exactly that partial space in $50/50$ runs with zero false admissions, while Learned-row reduces mean recovery from 18.52 to 13.40 settings (Appendix F.4). Runtime and classical-scaling measurements are reported in Appendix E.

6 Discussion

Learning takeaway. Verified constraints form compact memory and a version space for the next query. The same relation state drives acquisition across stepped, local, and check structure, while Random Clifford realizes the isotropic boundary where that state supplies no preferred direction.

More broadly, state or circuit families may induce graphical, higher-order, or nonlinear relation languages beyond stabilizer $\mathrm{GF}(2)$ parity—for example, computational-basis variables across a Toffoli gate satisfy $t' = t \oplus (c_1 \wedge c_2)$. The same architecture would maintain a compact verified constraint state and compile it into executable measurements, linking quantum structure learning, constraint learning, and adaptive experimental design.

AI USE STATEMENT

Generative AI tools were used only to edit the manuscript for language, clarity, and readability. The authors reviewed all edits and take full responsibility for the final paper.

REPRODUCIBILITY STATEMENT

Appendices E–G document the targets, protocols, seeds, stopping rules, statistical analyses, controls, verification, and resource accounting. The anonymous artifact will provide the frozen targets, configurations, results, environment metadata, and scripts needed to regenerate the reported figures and tables.

REFERENCES

S. Aaronson and D. Gottesman. Identifying stabilizer states. Perimeter Institute (PIRSA:08080052), 2008.

Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004.

Andreas Bengtsson, Alex Opremcak, Mostafa Khezri, Daniel Sank, Alexandre Bourassa, Kevin J. Satzinger, Sabrina Hong, Catherine Erickson, Brian J. Lester, Kevin C. Miao, Alexander N. Korotkov, Julian Kelly, Zijun Chen, and Paul V. Klimov. Model-based optimization of superconducting qubit readout. *Physical Review Letters*, 132:100603, 2024. doi: 10.1103/PhysRevLett.132.100603.

C. Bertoni, J. Haferkamp, M. Hinsche, M. Ioannou, J. Eisert, and H. Pashayan. Shallow shadows: Expectation estimation using low-depth random clifford circuits. *Physical Review Letters*, 133: 020602, 2024. doi: 10.1103/PhysRevLett.133.020602.

Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003. doi: 10.1145/792538.792543.

N.-H. Chia, C.-Y. Lai, and H.-H. Lin. Efficient learning of t-doped stabilizer states with single-copy measurements. *Quantum*, 8:1250, 2024.

L. O. Conlon, T. Vogl, C. D. Marciniak, et al. Approaching optimal entangling collective measurements on quantum computing platforms. *Nature Physics*, 19:351–357, 2023. doi: 10.1038/s41567-022-01875-7.

M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu. Efficient quantum state tomography. *Nat. Commun.*, 1(1):149, 2010.

Ninnat Dangniam, Yun-Guang Han, and Huangjun Zhu. Optimal verification of stabilizer states. *Physical Review Research*, 2:043323, 2020. doi: 10.1103/PhysRevResearch.2.043323.

C. Ferrie. Self-guided quantum tomography. *Phys. Rev. Lett.*, 113(19):190404, 2014.

L. E. Fischer, D. Bultrini, T. Dao, I. Tavernelli, and F. Tacchino. Large-scale implementation of quantum subspace expansion with classical shadows. *Physical Review Letters*, 2026. doi: 10.1103/2w21-qs38. Accepted 24 August 2026.

Steven T. Flammia and Yi-Kai Liu. Direct fidelity estimation from few pauli measurements. *Physical Review Letters*, 106:230501, 2011. doi: 10.1103/PhysRevLett.106.230501.

C. Gidney. Stim: a fast stabilizer circuit simulator. *Quantum*, 5:497, 2021. doi: 10.22331/q-2021-07-06-497.

D. Gottesman. *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, 1997.

D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. *Phys. Rev. Lett.*, 105(15):150401, 2010.

J. Haah, A. W. Harrow, Z. Ji, X. Wu, and N. Yu. Sample-optimal tomography of quantum states. *IEEE Trans. Inf. Theory*, 63(9):5628–5641, 2017.

Charles Hadfield, Sergey Bravyi, Rudy Raymond, and Antonio Mezzacapo. Measurements of quantum hamiltonians with locally-biased classical shadows. *Communications in Mathematical Physics*, 391:951–967, 2022. doi: 10.1007/s00220-022-04343-8.

S. Hadfield. Adaptive pauli shadows for estimating quantities of interest. *Phys. Rev. A*, 105(2): 022422, 2022.

Zhibo Hou, Jun-Feng Tang, Jiangwei Shang, Huangjun Zhu, Jian Li, Yuan Yuan, Kang-Da Wu, Guo-Yong Xiang, Chuan-Feng Li, and Guang-Can Guo. Deterministic realization of collective measurements via photonic quantum walks. *Nature Communications*, 9:1414, 2018. doi: 10.1038/s41467-018-03849-x.

H.-Y. Hu, A. Gu, S. Majumder, H. Ren, Y. Zhang, D. S. Wang, Y.-Z. You, Z. Minev, S. F. Yelin, and A. Seif. Demonstration of robust and efficient quantum property learning with shallow shadows. *Nature Communications*, 16:2943, 2025. doi: 10.1038/s41467-025-57349-w.

H.-Y. Huang, R. Kueng, and J. Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16:1050–1057, 2020.

H.-Y. Huang, R. Kueng, and J. Preskill. Efficient estimation of pauli observables by derandomization. *Phys. Rev. Lett.*, 127(3):030503, 2021.

A. Kalev, A. Kyrillidis, and N. M. Linke. Validating and certifying stabilizer states. *Phys. Rev. A*, 99 (4):042337, 2019.

Keijo Korhonen, Hetta Vappula, Adam Glos, Marco Cattaneo, Zoltán Zimborás, Elsi-Mari Borrelli, Matteo A. C. Rossi, Guillermo García-Pérez, and Daniel Cavalcanti. Practical techniques for high-precision measurements on near-term quantum hardware and applications in molecular energy estimation. *npj Quantum Information*, 11:110, 2025. doi: 10.1038/s41534-025-01066-1.

H. Lange, M. Kebrič, M. Buser, U. Schollwöck, F. Grusdt, and A. Bohrdt. Adaptive quantum state tomography with active learning. *Quantum*, 7:1129, 2023.

D. H. Mahler, L. A. Rozema, A. Darabi, C. Ferrie, R. Blume-Kohout, and A. M. Steinberg. Adaptive quantum state tomography improves accuracy quadratically. *Phys. Rev. Lett.*, 111(18):183601, 2013.

A. Montanaro. Learning stabilizer states by bell sampling, 2017. arXiv:1707.04012.

Yingkai Ouyang and Marco Tomamichel. Learning quantum graph states with product measurements. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 2963–2968, 2022. doi: 10.1109/ISIT50566.2022.9834440.

Y. Quek, S. Fort, and H. K. Ng. Adaptive quantum state tomography with neural networks. *npj Quantum Inf.*, 7(1):105, 2021.

Robert Raussendorf and Hans J. Briegel. A one-way quantum computer. *Physical Review Letters*, 86 (22):5188–5191, 2001. doi: 10.1103/PhysRevLett.86.5188.

Vladyslav Verteletskyi, Tzu-Ching Yen, and Artur F. Izmaylov. Measurement optimization in the variational quantum eigensolver using a minimum clique cover, 2020. URL `https://arxiv.org/abs/1907.03358`.

M. Votto, M. Ljubotina, C. Lancien, J. I. Cirac, P. Zoller, M. Serbyn, L. Piroli, and B. Vermersch. Learning mixed quantum states in large-scale experiments. *Physical Review Letters*, 136:090801, 2026. doi: 10.1103/rbg2-f61m.

Ting Zhang, Jinzhao Sun, Xiao-Xu Fang, Xiao-Ming Zhang, Xiao Yuan, and He Lu. Experimental quantum state measurement with classical shadows. *Physical Review Letters*, 127:200501, 2021. doi: 10.1103/PhysRevLett.127.200501.

GUIDE TO THE SUPPLEMENTARY MATERIAL

The theoretical argument follows one chain: verified constraints define an exposure spectrum (A); the actual compiler reshapes it through matching and joint capture (B); this produces either scale-changing bottleneck gains or structure-growing exposure gains, with their embeddings (C); and symmetry and the cost of displaced exploration locate the boundary of useful bias (D). The production GHZ analysis closes the block-establishment and terminal bottleneck stages, giving a full-trajectory acquisition setting-complexity separation under exact extraction. The second regime gives strict conditional exposure gains, with production rollouts testing their accumulation; its local-matching and joint-capture routes can coexist. Protocols, controls, and noisy admission then test this same account (E–G), followed by related-work positioning (H). The numbered mechanism results are the compiler's matching/capture bound, strict boundary gain with its branch extension, and preservation of check-capture gains across shared interfaces. Residual identities, waiting-cost calculations, and the symmetry reference model support that argument; the admission bound in G supports the verified representation interface. The table gives the reading route.

Question / role	Result and location
What is learned and exposed?	Appendix A: representation, residual geometry, learned-row construction, and independent-rank spectrum.
How does the policy change exposure?	Appendix B: actual-proposal retention, matching compression, joint capture, and the cost of replacing exploration.
When does the scale change?	Appendix C.3: a family-independent spectral-bottleneck theorem, production GHZ closure, a Ring-derived non-GHZ quotient-feedback separation, and parallel-block composition.
How do structural gains accumulate and transfer?	Appendix C.1: exact finite-budget IRD lifting; Appendices C.4–C.5: boundary matching with branched-corridor transfer and check capture with shared-axis transfer.
What limits generic lookahead?	Appendix D: a maximum-entropy G-only information-gain objective is flat; structural capture is measured against displaced exploration.
Recovery and controls	Appendices E–F: protocol ledger, targets, recovery contrasts, and interventions.
Noisy learning and tail search	Appendix G: fresh-data validation, trajectory error bounds, and actual resource accounting.
Task and access boundary	Appendix H: reconstruction, prediction, promised state recovery, and stabilizer-type constraint discovery.

A FROM STABILIZER-TYPE CONSTRAINTS TO THE EXPOSURE SPECTRUM

A.1 NOTATION AND THE MAINTAINED REPRESENTATION

Modulo phase, write a Pauli as $(x, z) \in \mathbb{F}_2^{2n}$, with symplectic product $x \cdot z' + z \cdot x'$. Let S be the isotropic space of exact Pauli/stabilizer relations contained in the target state, with rank $r_S \le n$; the complete pure-stabilizer case has $r_S = n$ and $S = S^\perp$. Signs are stored separately. Throughout the exposure analysis, $G \subseteq S$ is the correctly learned span, $d = n - \dim G$, $\ell = r_S - \dim G$, and M_s is the span of the single-site axes of a complete product setting s. This is an interface condition, not an assumption about how the observations were collected or processed. Exposure is defined by the target relations compatible with the chosen axes. Extraction and validation determine which of these relations update G; their statistical guarantees are addressed in Appendix G.

Symbol	Meaning
S, G	True and learned relation spaces
$Q_G = G^\perp/G$, $L_G = S/G$	Feasible residual directions and true residual information
$U_s = (G + (S \cap M_s))/G$	New independent information exposed by setting s
$\Delta r = \dim U_s$, $g_\pi(G) = \mathbb{E}_\pi \Delta r$	Exposed rank and its policy expectation
C, $\rho_G(C)$, $\nu(C)$	Compatible core, retention bound, and target rank in its span

Residual geometry. Q_G is a nondegenerate symplectic space of dimension $2d$. Its Lagrangian subspaces correspond exactly to complete stabilizer extensions containing G, while the true residual relation space L_G is an ℓ-dimensional isotropic subspace. For every complete product setting,

$$E_G(s) = \pi_G(M_s \cap G^\perp)$$

is a d-dimensional Lagrangian subspace and $U_s = E_G(s) \cap L_G$. Consequently, $\Delta r = \dim(E_G(s) \cap L_G)$. To see this, isotropy of G gives the symplectic quotient; taking preimages establishes the completion correspondence. Put $a = \dim(M_s \cap G)$. The pairing map from G to M_s^* has kernel $G \cap M_s$, hence rank $\dim G - a$. Its annihilator in M_s is $M_s \cap G^\perp$, of dimension $d + a$. Quotienting removes a dimensions, leaving an isotropic space of dimension d. If a class belongs to both $E_G(s)$ and L_G, its representatives in M_s and S differ by an element of $G \subseteq S$; thus its measurable representative also lies in S. This gives the exposed-rank identity in Equation 2.

Exposure and admitted updates. Here $\Delta r = \Delta_G(s)$ denotes exposed rank, not an automatic claim about the implemented rank increment. If the extractor admits only correct relations measurable in the current setting, its new span is a subspace of U_s and its admitted rank is at most $\Delta_G(s)$. Complete extraction and acceptance give equality. These statements require no noise model: noise, finite samples, and candidate restrictions affect the extraction and admission steps. Using the same extractor across policies fixes this interface, but need not preserve an ordering of exposure gains, since the policies may expose different relations. Throughout the mechanism analysis, gain means exposed independent rank; statements about recovery times additionally specify extraction conditions.

Ambiguity remaining after learning G. For two complete settings inducing distinct slices E_1, E_2, the completion $L_G = E_1$ gives gains d and $\dim(E_1 \cap E_2) < d$, respectively; the completion $L_G = E_2$ reverses the comparison. Both are valid stabilizer completions of the same verified span, with signs extending the known signs. Thus a G-only rule cannot choose a setting attaining the target-specific maximum for both completions. This establishes ambiguity of query value. Policies with an explicit target-class law or additional likelihood state can resolve part of that ambiguity. Finite outcome records and rejected candidates may also carry statistical evidence beyond the stored positive constraints. The quotient is an exact algebraic reduction given G; a policy must still choose how to explore it.

A.2 Learned-row construction used by the acquisition policy

Let $A = \operatorname{supp} G$ and $B = A^c$. The feasible space decomposes as $G^\perp = (G^\perp \cap V_A) \oplus V_B$. The policy constructs proposals only on A; all unassigned axes, including B, are sampled uniformly.

Algorithm 2 expands that call into the pivot and update rule it actually executes. All results in this paper were computed with `stim 1.16.0` (release commit `e2fc1ec`), whose `from_stabilizers` is the loop in `src/stim/util_top/stabilizers_to_tableau.inl`. Three properties of that pinned version are used below, and all three are re-checked against the installed binary by the scripts archived with the artifact:

(S1) *The Z outputs are the supplied rows.* For $k < r$, Z_k equals the k-th independent supplied row. The supplied row order is therefore observable and no rebasing occurs.

(S2) *Pivots are first-unused-sites.* Row k consumes the first position $q \geq u$ in π-order on which it acts nontrivially, and positions are consumed in increasing order of u. The consumed set, and hence the free set, is a deterministic function of the supplied row order and of π alone.

(S3) *Free coordinates are bare.* A position that is never consumed is returned as the bare single-site pair X_j, Z_j on that position; no basis change is applied to the untouched coordinates.

Together (S1)–(S3) are the sense in which

$$\text{input row order} + \text{site order } \pi + \text{pivot/update rule} \implies \text{output logical pairs}$$

is a deterministic map rather than an implementation detail.

The compiler receives the maintained verified rows in their supplied order; in the implementation the public entry point concatenates the maintained local rows followed by the maintained global rows. It traverses that list once and keeps the first row that increases the $\mathrm{GF}(2)$ span, so row reduction is used only for independence and span tests and does not replace retained physical rows by RREF combinations. On the active support, sites are ordered by decreasing number of retained rows that touch them, with the physical index breaking ties. After restricting the retained rows to this ordered support, the implementation performs the deterministic underconstrained stabilizer-tableau completion specified in Algorithm 1. If $r = \operatorname{rank} G$ and $m = |A|$, it takes the tableau X- and Z-outputs indexed by $j = r, \ldots, m-1$ and embeds them back to the original physical qubits. These $2(m - r)$ representatives form the learned-row production frame for $(G^\perp \cap V_A)/G$. The

Algorithm 2 Deterministic symplectic elimination inside StabilizerComplete

Require: independent commuting rows $g_1, \ldots, g_r$ in the supplied order; site order π on A; $m = |A|$
Ensure: conjugate pairs (X_j, Z_j) for $j = r, \ldots, m-1$

$u \leftarrow 0$ ▷ u = first position not yet consumed
for $k = 1, \ldots, r$ **do**
  Reject if g_k carries X on some π_q with $q < u$ ▷ anticommutation guard
  $p \leftarrow \min\{\, q \geq u : \; g_k \text{ acts nontrivially on } \pi_q \,\}$ ▷ first unconsumed site it touches
  if no such p exists **then**
    Discard g_k ▷ redundant row; contributes no pivot
  end if
  If g_k has X or Y at π_p, apply H (or H_{YZ}) at π_p ▷ pivot axis $\to Z$
  for each $q \neq p$ with g_k nontrivial at π_q **do**
    Apply XCX, XCZ or XCY from π_p to π_q ▷ clear the rest of row k
  end for
  If $p \neq u$, exchange the data at positions p and u ▷ move pivot onto the diagonal
  If g_k has sign -1, apply X at π_u ▷ fix the sign
  $u \leftarrow u + 1$
end for
return the logical outputs of positions $u, \ldots, m-1$, embedded by π^{-1}

production call keeps this frame and these tableau lifts exactly: quotient-basis randomization and support-preserving coset-lift randomization define the randQ and repR interventions, respectively. Thus the site-incidence information carried by the admitted rows remains part of the physical proposal frame. For the pure-Z block spans used in the GHZ proof, the specified completion yields the block-supported logical pair stated in Lemma 1. The exact GHZ and corridor certificates below refer to this deterministic completion and its stated row/site order; the abstract quotient alone does not determine their physical conflict graphs.

Randomly permute the retained proposals. A proposal is packed iff each of its nonidentity letters agrees with every axis already assigned at that site. Fill unassigned axes independently from $\{X, Y, Z\}$. This construction returns a random-order maximal compatible set; exact maximum-cardinality packing is a separate combinatorial objective. The headline Learned-row configuration uses the deterministic learned-row basis and lifts, untruncated packing, and randomized proposal order; the randQ, repR, K, and WeightSort controls change these choices individually. If $G = 0$, or if the active-support quotient produces no proposal, the production acquisition loop draws a fully uniform Random setting.

The former RREF-coordinate compiler is retained as an explicit ablation. It forms commutation equations on A in fixed binary-column order, obtains a free-column nullspace basis, and filters it modulo G. RREF/free-column is a separate coordinate construction. The randQ, repR, and weight-order controls instead start from the learned-row production frame and alter one design axis at a time. All structural certificates used for the main mechanism claims are computed on that production frame.

Packing objective and computational scope. On a proposal conflict graph, maximum-cardinality packing is maximum independent set; fixed nonnegative additive weights give its weighted version. The unrestricted Pauli problem is NP-hard: for any simple graph H, assign two qubits to each edge $\{u, v\}$. Put XX in proposal q_u and ZZ in q_v on that edge's qubits, and identities in all other proposals. Add a private X qubit to each proposal. The proposals are linearly independent and globally commute, but have a product-measurement conflict exactly on the edges of H. Compatible subsets therefore correspond exactly to independent sets of H. This reduction establishes worst-case hardness for unrestricted proposal sets; the learned-row and RREF-coordinate frames encountered here may have additional structure. Related qubit-wise-commuting grouping problems also use graph optimization (Verteletskyi et al., 2020).

Our scan requires $O(m|A|)$ axis comparisons for m proposals on active support A, without solving a cardinality objective. Its design criterion is retention of potentially useful cores and continued exploration: Proposition 2 bounds the former. Cardinality and additive proposal weights optimize different objectives from unknown-target rank gain, which also includes joint products and random completion. The retention bound quantifies core survival under randomized packing and is used only for the exposure guarantees below.

A.3 Independent-rank exposure spectrum

Choose a complete flag $0 = F_0 < F_1 < \cdots < F_\ell = L_G$, $\dim F_i = i$, and define

$$p_{\pi,i} = \Pr_{\pi}\{U_s \cap (F_i \setminus F_{i-1}) \neq \varnothing \mid G\}, \qquad w_{\mathrm{eff},\pi,i} = -\log_3 p_{\pi,i}.$$

Zero probability corresponds to infinite effective weight. The increment $\dim(U_s \cap F_i) - \dim(U_s \cap F_{i-1})$ is zero or one and equals the indicator of the defining event. Telescoping and taking expectations gives

$$g_\pi(G) = \sum_i p_{\pi,i} = \sum_i 3^{-w_{\mathrm{eff},\pi,i}}, \tag{A.1}$$

where no independence of events is required. For a single nonzero class $c \in S/G$, we also use $w_{\mathrm{eff},\pi}(c) = -\log_3 \Pr(c \in U_s)$. These class probabilities cannot all be summed as rank: an exposed k-dimensional space contains $2^k - 1$ nonzero classes. Flag terms depend on the chosen decomposition, whereas their sum is invariant. Policy comparisons at fixed G use the same flag.

B How the Compiler Changes the Exposure Spectrum

The compiler must turn an algebraic opportunity into a physically compatible measurement. Three quantities determine its value: the probability of retaining a useful core, the independent relations that core can expose, and the exploration left after fixing its axes. The retention result applies to any proposal family produced from G without target access; later exact certificates state the particular G-derived frame they analyze.

B.1 Compatible cores: matching compression and joint capture

Let the vertices of a conflict graph be the compiler proposals; an edge means disagreement of nonidentity letters at some site. Let C be a compatible core. For each nonempty $U \subseteq C$, let b_U count outside proposals whose neighbors within C are exactly U. Define

$$\rho_G(C) = \prod_{U : b_U > 0} \frac{|U|}{|U| + b_U}. \tag{B.1}$$

Proposition 2 (From physical compatibility to independent exposure)**.** *Under Algorithm 1's untruncated random-order packing, every member of C is retained jointly with probability at least $\rho_G(C)$. Suppose $P \in S \setminus G$ and $P = g + \sum_{i \in C} q_i + t$, where $g \in G$, C is a compatible subset of the compiler proposals, and t is supported outside the active support. Then*

$$\Pr_A([P] \in U_s \mid G) \geq \rho_G(C) 3^{-\mathrm{wt}(t)}.$$

If the core's span contains $\nu(C)$ independent target classes, then $g_A(G) \geq \rho_G(C)\nu(C)$.

Proof. Assign independent uniform priorities to all proposals. Require every outside blocker to occur after at least one of its core neighbors. Under this event, consider the first core member that could be rejected. Its accepted blocker would itself have been rejected by an earlier accepted core neighbor, a contradiction. Conditional on core priorities (T_i), the sufficient event has probability $\prod_U (1 - \min_{i \in U} T_i)^{b_U}$. Its factors are all coordinatewise decreasing functions of independent priorities, so positive association bounds the expectation below by the product of expectations. For each group this expectation is $|U|/(|U| + b_U)$, obtained by integrating the minimum of $|U|$ uniforms. When C is retained, its sum is measurable. The required inactive letters are still independent uniform axes, giving the matching factor. The resulting representative differs from P by known information. For the rank bound, the entire compatible span is measurable whenever C is retained, so its target intersection contributes to exposed rank. □

Corollary 2 (Direct production-frame certificate)**.** *At any correct verified span G, apply Proposition 2 to the learned-row proposal family returned by Algorithm 1. Every compatible core, blocker count, residual-letter count, and captured-rank bound then holds for the production query itself; no RREF or target-family coordinate assumption is required.*

Corollary 2 lets the structural certificates below compute these quantities directly in learned-row coordinates. Physical representatives, in addition to quotient span, determine Pauli-letter conflicts. The production proofs therefore use learned-row coordinates directly, while RREF/free-column serves as the algorithmic ablation. For one candidate the retention bound is $1/(1 + \deg i)$. If b blockers all share k compatible core neighbors, it is $k/(k + b)$. Shared blockers can therefore make joint retention much less costly than treating each candidate as a separate trial. This connects proposal-level conflicts to information exposure under the implemented random ordering. Matching compression reduces the

remaining random-letter cost, while joint capture increases the independent information carried by a retained core. A finite cap K also requires retention before truncation; the displayed bound concerns the full production pack.

In any residual coordinate frame, write the target intersection as the kernel of a linear map Y. Restricting to C gives $\nu(C) = |C| - \operatorname{rank} Y_C$. Bounds for independent target witnesses may be summed even when their exposure events are dependent. Overlapping target spaces instead require the rank counting of Appendix A.3.

B.2 What packing must gain over random completion

For each realized pack, let P_K be the span of its quotient proposals. Its true captured rank is $\dim(P_K \cap L_G)$. Physical compatibility puts this subspace inside the full exposed space U_s; the remaining rank is $\dim U_s - \dim(P_K \cap L_G)$. Averaging gives κ_K and χ_K in the exact decomposition

$$\Gamma_K - \Gamma_0 = \kappa_K - (\Gamma_0 - \chi_K), \qquad \Gamma_K = \kappa_K + \chi_K, \tag{B.2}$$

where Random has an empty pack, so $\kappa_0 = 0$.

The algorithmic consequence is that neither pack size nor individual proposal weight is the objective: proposals are useful through the independent target information they jointly expose, after accounting for displaced exploration. Frozen-G replay measures exactly these two terms. The target is used only for offline scoring of this decomposition; proposal construction depends on G.

B.3 Why the mechanism can be local to an unknown target

A local information gain can apply inside different unknown targets. Fix the learned span and local information $K = G + W \subseteq S$. For any setting M,

$$U_M \cap (K/G) = \big(G + (K \cap M)\big)/G. \tag{B.3}$$

Indeed, if $x = g + t \in K$ with $g \in G$ and $t \in S \cap M$, then $t = x - g \in K \cap M$. The reverse inclusion is immediate. Take the quotient by G. For a G-only policy, Equation B.3 makes the local exposure law independent of how the unknown target is completed outside K. Changing the learned span is different: it can change physical proposals and their conflicts. The structural results below specify connections that preserve useful cores under that recompilation.

C Exposure Regimes and Transferable Structural Gains

The structural conclusions have two strengths. When feedback removes a necessary rare exposure scale, that bottleneck controls completion complexity; the GHZ analysis below closes both its preceding block-contraction phase and its terminal waiting cost for the production compiler. When Random already has a bounded exposure scale, the conclusion is a strict conditional exposure gain together with a finite-horizon lifting identity. This second regime has two routes—local matching and joint check capture—whose accumulation is tested by production rollouts. Their exact certificates below use the learned-row proposals, physical conflicts, and random-order packing of the production compiler itself.

Why the two regimes support different guarantees. Let $K_S(G, G')$ denote the one-round transition law on verified spans for a fixed target S, including extraction and recompilation, and let $T = \inf\{t : G_t = S\}$. A conditional exposure result compares actions at an encountered G and can transfer whenever the same local physical certificate appears inside a larger target. By contrast, an exact value or bound for $\mathbb{E}T$ must also control which certified states are reached and for how long: it depends on the target-specific kernel K_S over the entire reachable trajectory. In the bounded-scale regime, explicitly computing this kernel amounts to enumerating the same generator/extraction/packing evolution that a rollout samples, while obscuring the local condition that transfers to other graphs and codes. A scale-changing bottleneck is different: polynomial reachability plus one unavoidable rare decision can dominate T, so an full-trajectory acquisition setting separation follows under the exact extraction interface without resolving every intermediate state. Accordingly, the exposure geometry itself determines the conclusion: the GHZ bottleneck yields an acquisition completion-setting result under exact extraction, whereas bounded-scale Ring and check sectors yield conditional-exposure and transfer results, with an exact finite-budget criterion and production rollouts measuring its trajectory-level terms.

C.1 From conditional exposure to finite-budget end-to-end acceleration

The integrated rank deficit admits an exact lifting statement that does not require a closed form for completion time. For a policy π, write

$$R_t^\pi = \mathrm{rank}(G_t^\pi), \qquad A_t^\pi = R_{t+1}^\pi - R_t^\pi,$$

where A_t^π is the independent rank actually admitted after setting t. This definition includes the implemented extractor and verification rule.

Proposition 3 (Finite-horizon accumulation identity)**.** *For any two policies A and R with the same initial verified span and any horizon H,*

$$\mathrm{IRD}_H(R) - \mathrm{IRD}_H(A) = \frac{1}{n}\sum_{t=0}^{H-1}(H-t)\left(\mathbb{E}A_t^A - \mathbb{E}A_t^R\right), \quad \mathrm{IRD}_H(\pi) = \mathbb{E}_\pi \sum_{t=1}^{H}\left(1 - \frac{R_t^\pi}{n}\right). \tag{C.1}$$

Consequently a positive right-hand side is a strict end-to-end advantage in the reported finite-budget metric even when neither policy completes.

Proof. For either policy, $\sum_{t=1}^{H} R_t^\pi = HR_0 + \sum_{k=0}^{H-1}(H-k)A_k^\pi$. Subtract the policies, use their common R_0, divide by n, and reverse the sign because IRD counts residual rather than recovered rank. □

To connect Proposition 3, via Equation C.1, to the production mechanism, let

$$\widehat{g}_\pi(G) = \mathbb{E}[R_{t+1} - R_t \mid G_t = G, \pi]$$

denote admitted gain. Under complete extraction $\widehat{g}_\pi(G) = g_\pi(G)$; with a restricted extractor it is the exposed gain minus missed independent rank. Add and subtract Random's gain on the Adaptive-occupied state:

$$M_t = \mathbb{E}_{G_t^A}[\widehat{g}_A(G_t^A) - \widehat{g}_R(G_t^A)], \tag{C.2}$$

$$B_t = \mathbb{E}_{G_t^R}\widehat{g}_R(G_t^R) - \mathbb{E}_{G_t^A}\widehat{g}_R(G_t^A). \tag{C.3}$$

Here M_t is the same-state production margin isolated by frozen-G interventions. B_t is the trajectory-shift penalty: it is positive when Adaptive has consumed easier directions and therefore reaches a residual spectrum on which a hypothetical Random query gains less. The definitions give the exact decomposition

$$\mathrm{IRD}_H(R) - \mathrm{IRD}_H(A) = \frac{1}{n}\sum_{t=0}^{H-1}(H-t)(M_t - B_t). \tag{C.4}$$

Corollary 3 (Structural finite-budget acceleration certificate)**.** *Suppose production-core and displaced-exploration calculations give $M_t \geq m_t$, while the Random exposure spectra on the two occupied state distributions give $B_t \leq b_t$. Then*

$$\mathrm{IRD}_H(R) - \mathrm{IRD}_H(A) \geq \frac{1}{n}\sum_{t=0}^{H-1}(H-t)(m_t - b_t).$$

If the right-hand side is positive, production has a strict end-to-end finite-budget advantage over Random. If every prefix $\sum_{t=0}^{h-1}(m_t - b_t)$ is nonnegative, with a strict prefix somewhere, then production has at least as large expected recovered rank at every corresponding budget and a strict advantage for every strictly positive prefix. Under incomplete extraction the same statement holds for admitted gains, or for exposure gains after subtracting a bound on differential missed rank.

Proof. Insert the two bounds into Equation C.4. The prefix statement follows from $\mathbb{E}R_h^A - \mathbb{E}R_h^R = \sum_{t=0}^{h-1}(M_t - B_t)$. □

Corollary 3 can be made uniform without computing either transition kernel. Write $D_t^\pi = \dim(S/G_t^\pi)$.

Theorem 2 (Production growth dominates Random)**.** *Assume sound updates and a common initial deficit $D_0 = d_0$. Suppose constants $0 \le b < a \le 1$ satisfy*

$$\mathbb{E}[A_t^A \mid G_t^A = G] \ge a \dim(S/G) \quad \textit{for every Adaptive-reachable } G,$$

and

$$\mathbb{E}[A_t^R \mid G_t^R = G] \le b \dim(S/G) \quad \textit{for every Random-reachable } G.$$

Then, for every $t \ge 1$,

$$\mathbb{E}D_t^A \le d_0(1-a)^t < d_0(1-b)^t \le \mathbb{E}D_t^R, \tag{C.5}$$

and hence Adaptive has strictly larger expected recovered rank. Moreover,

$$\mathrm{IRD}_H(R) - \mathrm{IRD}_H(A) \ge \frac{d_0}{n} \sum_{t=1}^{H} \big[(1-b)^t - (1-a)^t\big] > 0. \tag{C.6}$$

This theorem provides an end-to-end finite-budget comparison of the two trajectories; asymptotic completion order is analyzed only in the bottleneck regime.

Proof. Because $D_{t+1}^\pi = D_t^\pi - A_t^\pi$, the two conditional assumptions give $\mathbb{E}[D_{t+1}^A \mid G_t^A] \le (1-a)D_t^A$ and $\mathbb{E}[D_{t+1}^R \mid G_t^R] \ge (1-b)D_t^R$. Taking expectations and iterating proves Equation C.5. Summing the expected-deficit difference over $t = 1, \dots, H$ proves Equation C.6. □

The constants in Theorem 2 are certified in the physical production frame. If at every Adaptive-reachable G a collection of independent learned-row witness cores has total admission-weighted lower bound

$$\sum_j \eta_j \rho_G(C_j) 3^{-r_j} \nu_j \ge a \dim(S/G),$$

then Proposition 2 supplies the first hypothesis. For Random, the coset weight mass $Z_S(1/3) - Z_G(1/3)$ upper-bounds exposed, hence admitted, rank; bounding it by $b \dim(S/G)$ on Random-reachable spans supplies the second. Thus local matching and joint capture can yield a trajectory-level theorem through motif density and coset mass, without enumerating individual transition probabilities. The exact Ring and BB5 calculations in Appendices C.4 and C.5 certify the constituent production margins; the rollouts test their persistence where no uniform family-wide (a, b) bound is asserted.

Equations 3 and 4 supply production lower bounds contributing to m_t, after displaced random completion is accounted for. The Random side is read from the same exposure spectrum in Equation A.1. The explicit B_t term accounts for the cost of trajectory shift when lifting frozen-state surplus to end-to-end finite-budget performance. Structural motif counts or phase-wise spectral bounds can control these aggregate terms without enumerating the full transition kernel; target-randomized rollouts test their realization when no uniform family bound is asserted.

C.2 Exposure scales: a bounded-weight reference

A complete generating set of weight at most w_* supplies a reference scale for Random without being given to the learner. For any $G \subseteq S$, choose d generators whose classes form a basis of S/G. Their direct exposure probabilities are at least $a = 3^{-w_*}$, and any simultaneously exposed subset is independent. Thus $g_R(G) \ge d3^{-w_*}$. With complete extraction this gives $\mathbb{E}[d_{t+1} \mid G_t] \le (1-a)d_t$ and hence $\Pr(T_R > t) \le \min(1, n(1-a)^t)$. Summing the tail, split at $t = \lceil a^{-1} \log n \rceil$, yields the reference recovery cost

$$\mathbb{E}T_R = O(3^{w_*}(1 + \log n)). \tag{C.7}$$

Equation C.7 gives the reference scale; Ring has $w_* = 3$. In BB5-30, the full weight-five pure-X and pure-Z collections have ranks 13 and 17, respectively, and span the pure target. This fact is instance-specific; the supplied logical generators need not themselves be a minimum-weight basis.

C.3 Bottleneck spectra: one or several scale changes

When easy relations leave a necessary, much rarer residual decision, changing its exposure scale can change recovery complexity. The decisive calculation is how learned constraints change that decision's probability; elapsed time then follows from geometric waiting costs.

A structural certificate rather than a hitting-time assumption. Let $E_n < S_n$ be a codimension-one subgroup and write the remaining coset as $\mathcal{C}_n = h_n + E_n$. For a verified span G, define the missing skeleton rank

$$d_E(G) = \dim E_n - \dim(E_n \cap G).$$

Call E_n (c_*, ρ_*, r_*)-*regenerative for the production frame* if, at every production-reachable correct G with $d_E(G) > 0$, the actual learned-row proposal family contains $m(G) \geq c_* d_E(G)$ independent skeleton classes with compatible cores $C_1, \ldots, C_{m(G)}$ such that $\rho_G(C_j) \geq \rho_*$ and each corresponding representative requires at most r_* random letters outside its core. This definition is checkable from the same physical proposals and blocker graph that Algorithm 1 uses; it does not posit a separate convergence time.

For the remaining coset define its product-measure enumerator

$$B_n := Z_{\mathcal{C}_n}(1/3) = \sum_{P \in \mathcal{C}_n} 3^{-\operatorname{wt}(P)}. \tag{C.8}$$

With B_n defined by Equation C.8, call the terminal coset (ρ_B, r_B)-accessible if, at $G = E_n$, the actual learned-row proposal family has a compatible core for one representative of $\mathcal{C}_n$, retained with probability at least ρ_B and requiring at most r_B random letters.

The extractor is (η_E, η_B)-*admissible on these certificates* if, conditional on a correct current span and on exposure, every certified skeleton witness is admitted as new independent rank with probability at least η_E, and a certified terminal-coset witness is admitted with probability at least η_B. This condition is imposed only on the witness family used in the proof; relations outside that family may be recovered or missed independently. A sound extractor additionally admits no relation outside the target span and assigns the correct sign.

Theorem 3 (Admission-robust structural spectral-bottleneck separation)**.** *This is the admission-robust extension of Theorem 1, which is the complete-extraction case $\eta_E = \eta_B = 1$; it is the form proved here. If E_n is (c_*, ρ_*, r_*)-regenerative, $\mathcal{C}_n$ is (ρ_B, r_B)-accessible, and a sound extractor is (η_E, η_B)-admissible on these certificates, then*

$$\mathbb{E}T_{\text{prod}} \leq \frac{1 + \log d_E(0)}{\eta_E c_* \rho_* 3^{-r_*}} + (\eta_B \rho_B 3^{-r_B})^{-1}, \qquad \mathbb{E}T_{\text{Random}} \geq B_n^{-1}. \tag{C.9}$$

Hence polynomial inverse core parameters and exponentially small B_n imply a full-trajectory acquisition setting-complexity separation whenever η_E^{-1} and η_B^{-1} are polynomial. Complete extraction is the special case $\eta_E = \eta_B = 1$.

Proof. Fix a production-reachable G with $d = d_E(G) > 0$. Proposition 2 exposes and admits each of the $m(G)$ independent skeleton classes with probability at least $\eta_E \rho_* 3^{-r_*}$. Although these events need not be independent, the exposed classes are linearly independent, so linearity of expectation gives

$$\mathbb{E}[d_E(G) - d_E(G') \mid G] \geq \eta_E c_* d \rho_* 3^{-r_*}.$$

Multiplicative drift therefore reaches $d_E = 0$ in expected time at most the first term of Equation C.9. If the terminal class was not learned earlier, Proposition 2 and terminal admissibility give hazard at least $\eta_B \rho_B 3^{-r_B}$ at $G = E_n$, proving the production upper bound.

Uniform Random exposes a fixed Pauli P with probability $3^{-\operatorname{wt}(P)}$. By a union bound, the probability that one setting exposes any member of the necessary coset $\mathcal{C}_n$ is at most B_n. Completion cannot precede this event, and independent Random settings therefore have expected waiting time at least B_n^{-1} even if its extractor is perfect. □

The quantities in Theorem 3 have direct structural meanings: B_n is the low-weight mass of a necessary residual coset; c_* counts how much missing skeleton rank is represented by production cores; ρ_* and ρ_B are determined by their physical blockers; and r_*, r_B count only axes left to random completion. No target-specific transition kernel or pre-specified hitting time appears.

Which generator compositions instantiate the theorem. The following structural examples instantiate the theorem under the stated certificate conditions.

- *Cat/GHZ blocks.* The even-Z subgroup is a regenerative skeleton, while its complementary full-support X/Y coset has $B_n = 2^{n-1}/3^n$. The learned-row block calculation below verifies the core conditions exactly. Several disjoint blocks produce several parallel jumps rather than one monolithic tail.

- *Local checks plus logical sectors.* A bounded-weight commuting check skeleton of rank $n-k$ may be followed by k high-distance logical stabilizer cosets. Whenever its production cores regenerate the checks and give polynomially accessible logical representatives, any logical coset with exponentially small enumerator yields the same separation. Distance alone is insufficient: the physical core and blocker conditions are the required certificate.
- *Hybrid and multistage targets.* Tensoring a certified bottleneck component with a separately learned polynomial-cost local graph or code component preserves the rare-coset lower bound. Nested skeletons $E_0 < E_1 < \cdots < S$ give several successive spectral jumps by applying the theorem stagewise; disjoint supports give the parallel waiting law in Equation C.11.

Thus the theorem is controlled by a discontinuity in the exposure spectrum and the compiler's ability to regenerate the easy skeleton. The familiar presentation $\langle Z_i Z_{i+1}, X^{\otimes n}\rangle$ is the cleanest exact instantiation.

Production closure for the GHZ bottleneck. Write S_Z for the even-parity pure-Z subgroup and $H = X^{\otimes n}$ for the remaining GHZ class. Before H is learned, every verified span reached under complete extraction has the form

$$G = \bigoplus_{j=1}^{B} \mathrm{EvenZ}(B_j)$$

for a partition of the qubits into blocks, with singleton blocks contributing zero rank. Indeed, a product setting exposes the even-Z space on its Z-measured sites; adding it merges precisely the old blocks that those sites touch.

Lemma 1 (Learned-row block quotient)**.** *Let $G = \bigoplus_j \mathrm{EvenZ}(B_j)$ be supplied to the learned-row compiler in Algorithm 1 as any independent even-Z row set, in any row order. Then for every block B_j the completion returns exactly one hyperbolic pair*

$$X^{B_j} \quad \textit{and} \quad Z_{r_j}, \qquad r_j \in B_j,$$

where r_j is the unique site of B_j that the pivot rule of Algorithm 2 never consumes. Different blocks give disjointly supported pairs. Under random proposal order, the relative order within each pair is an independent fair bit; singleton sites are filled uniformly.

Proof. Every row of G is a pure-Z string supported inside a single block, and $\mathrm{EvenZ}(B_j)$ has rank $|B_j| - 1$.

Step 1: pivots stay inside their block. By (S2), the pivot of a row g is the first position $q \geq u$ in π-order whose site π_q lies in $\operatorname{supp} g$. Since $\operatorname{supp} g \subseteq B_j$ for the block B_j carrying g, that pivot is a site of B_j. Positions are consumed one at a time, so distinct rows receive distinct pivots, and no row can consume a site of another block. The $|B_j| - 1$ independent rows carried by B_j therefore consume exactly $|B_j| - 1$ distinct sites of B_j, all inside it, and a row of B_j is never left without an unconsumed site of its own block to pivot on. Hence precisely one site $r_j \in B_j$ is never consumed, and this count is unaffected by the order in which the blocks are interleaved in the supplied row list. By (S3) the free coordinate at r_j is returned bare, so its Z output is Z_{r_j} and its X output is a single pure-X string.

Step 2: the conjugate is X^{B_j}. A tableau's outputs form a canonical symplectic basis, so the free X output X^S must commute with every supplied row and with every other free Z output, and anticommute with Z_{r_j}. Commutation with $\mathrm{EvenZ}(B)$ forces $S \cap B \in \{\emptyset, B\}$: if $a \in S \cap B$ and $b \in B \setminus S$ then X^S anticommutes with $Z_a Z_b \in \mathrm{EvenZ}(B)$. Commutation with $Z_{r_{j'}}$ forces $r_{j'} \notin S$, hence $S \cap B_{j'} = \emptyset$ for $j' \neq j$; anticommutation with Z_{r_j} forces $r_j \in S$, hence $S \supseteq B_j$. Therefore $S = B_j$: the conjugate is exactly X^{B_j}, with no support on any other block, and it is the unique such operator. Production keeps these outputs without quotient rebasing, so different block pairs remain disjointly supported.

Step 3: orientations. A uniform permutation of the proposal list exchanges X^{B_j} and Z_{r_j} within each pair with probability $1/2$, independently across the disjoint blocks, and leaves singleton sites to uniform filling. □

Call a block *Z-touched* if the setting assigns Z to at least one of its sites. Lemma 1 gives touch probability $1/2$ for every nonsingleton block and $1/3$ for every singleton, independently across

blocks. Complete extraction merges every touched collection. If $B_t = B \geq 2$, coupling all touch probabilities down to $1/3$ gives

$$\mathbb{E}[B_{t+1} - 1 \mid B_t = B,\ H \notin G_t] \leq \frac{2B}{3} - \left(\frac{2}{3}\right)^B \leq \frac{8}{9}(B-1). \tag{C.10}$$

Equation C.10 gives $O(\log n)$ expected pre-bottleneck contraction. If H is exposed early, the maintained span remains CSS. For this GHZ branch, the deterministic completion in Algorithm 1 yields d independent pure-Z target proposals among its $2d$ residual outputs. Each pure-Z proposal is retained whenever it precedes all of its blockers, an event of probability at least $1/(2d)$; their expected independent exposed rank is therefore at least $1/2$. Additive drift bounds the remaining pure-Z completion by $O(n)$.

These calculations also verify the parameters of Theorem 3, rather than merely a separate drift argument. Before H is learned, choose a star of $B-1$ independent even-Z relations joining a reference block to the others. A relation between two nonsingleton blocks has the compatible core formed by their two odd-Z proposals, retained with probability $1/4$; a singleton endpoint instead costs one random Z match. Uniformly, one may take $c_* = 1$, $\rho_* \geq 1/4$, and $r_* \leq 2$. If H arrives early, the preceding blocker count gives the weaker but still polynomial $\rho_* \geq 1/(2n)$ with $r_* = 0$. At the terminal skeleton, $\rho_B = 1/2$ and $r_B = 0$. Thus one set of parameters valid across both production branches is $c_* = 1$, $\rho_* = 1/(2n)$, and $r_* = 2$.

At $G = S_Z$, the production quotient pair is H and one odd-Z representative. The two conflict, so random order gives $p_A = 1/2$. Uniform Random exposes the coset $H + S_Z$ through its 2^{n-1} full-support X/Y representatives, whose matching events are disjoint, and hence $p_R = 2^{n-1}/3^n$ in every round. We obtain:

Proof of Corollary 1. Under noiseless complete extraction, Algorithm 1 recovers GHZ_n in expected $\mathrm{poly}(n)$ settings (the Z-only branch contracts in $O(\log n)$), whereas uniform Random requires

$$\mathbb{E}T_R \geq \frac{3^n}{2^{n-1}} = 2(3/2)^n.$$

The polynomial upper bound includes the early-H branch; the usual trajectory reaches S_Z first and then pays only the constant terminal wait. □

Corollary 4 (Admission-robust GHZ separation)**.** *For any sound extractor that admits the certified even-Z witnesses with probability at least η_E and the terminal X/Y-coset witness with probability at least η_B whenever exposed,*

$$\mathbb{E}T_{\mathrm{prod}} \leq 18n\eta_E^{-1}(1 + \log(n-1)) + 2\eta_B^{-1}, \qquad \mathbb{E}T_R \geq 2(3/2)^n.$$

Thus the exponential separation survives finite-sample false rejection whenever η_E^{-1} and η_B^{-1} are polynomial. The corollary only requires admission of the certified witnesses; other exposed relations may be missed.

Natural boundary of the completion-complexity statement. Corollary 4 carries the GHZ separation through finite-sample admission. Theorem 3 gives the general separation in terms of a necessary rare-coset enumerator and production-core regeneration. The proof of Corollary 1 verifies those certificates for the actual learned-row compiler on GHZ_n under noiseless complete extraction and correct span updates; the block formula below verifies the parallel composition. When the exposure spectrum has no growing bottleneck, as for the ordinary Ring and BB5 families, its natural invariant is instead conditional matching or joint capture at structurally certified states.

Several parallel bottlenecks. For disjoint blocks of sizes $b_1, \ldots, b_k$ whose even-Z subgroups are known, the learned-row construction separates by support. Complete extraction therefore gives a residual completion time satisfying

$$\Pr(\tau > t) = 1 - \prod_{j=1}^{k} \left[1 - (1-p_j)^t\right], \quad p_{R,j} = 2^{b_j - 1}/3^{b_j}, \quad p_{A,j} = 1/2. \tag{C.11}$$

Uniform completion and pair orientations separate across supports. Once a block is complete it requires no further residual acquisition, so unresolved blocks have independent geometric waiting times, yielding the displayed law. The formula separates two structural effects: fixed-size blocks change the number of parallel residual tails, whereas increasing block size changes their exposure scale.

A non-GHZ local-plus-global separation. The previous GHZ closure proves the rare-bottleneck mechanism for the actual production compiler. A second Ring-derived construction isolates the same scale change outside GHZ. Its local sector is already easy for uniform Random; feedback acts on the final rare residual class.

Proof of Proposition 1. For $n \geq 5$, let $K_i = Z_{i-1}X_iZ_{i+1}$ on a cycle and let

$$E_n = \left\{ \prod_{i=1}^{n} K_i^{a_i} : \sum_i a_i = 0 \pmod 2 \right\}, \qquad H_n = Z^{\otimes n}, \qquad S_n = \operatorname{span}(E_n, H_n).$$

Under noiseless complete extraction, consider the policy that uses uniform Random product-Pauli settings while $\operatorname{rank} G < n-1$. If this stage reaches rank n, recovery is complete; otherwise, at rank $n-1$ it chooses uniformly one of the three nonzero classes of $Q_G = G^\perp/G$, takes any physical Pauli representative of that class, and queries a product setting that matches all of its nonidentity letters. Then

$$\mathbb{E}T_{\mathrm{qfb}} \leq \left\lceil \frac{\log(n-1)}{\log(81/80)} \right\rceil + 84 = O(\log n), \qquad \mathbb{E}T_R \geq \frac{1}{4}(3/\lambda)^n, \quad \lambda = 2.931142\ldots < 3. \tag{C.12}$$

Write

$$L_i := K_iK_{i+1} = Z_{i-1}Y_iY_{i+1}Z_{i+2}$$

modulo phase. In the coefficient coordinates of the independent Ring generators, L_i corresponds to $e_i + e_{i+1}$. Hence $L_1, \ldots, L_{n-1}$ are independent and span the even-parity subgroup E_n, so $\dim E_n = n-1$. The operator H_n commutes with every even product of the K_i. In symplectic coordinates, every nonidentity $K_T = \prod_{i\in T} K_i$ has X-component $\mathbf{1}_T \neq 0$, whereas H_n has zero X-component and nonzero Z-component. Hence $H_n \notin \operatorname{span}\{K_i\}$ and in particular $H_n \notin E_n$; therefore S_n is abelian and full rank.

A uniform product-Pauli setting exposes a fixed L_i with probability $3^{-4} = 1/81$. Let τ_E be the first time that all $L_1, \ldots, L_{n-1}$ have been exposed. Complete extraction admits every such true relation, and a union bound gives

$$\Pr(\tau_E > t) \leq (n-1)\left(\frac{80}{81}\right)^t.$$

With

$$t_0 = \left\lceil \frac{\log(n-1)}{\log(81/80)} \right\rceil,$$

the tail-sum formula therefore yields

$$\mathbb{E}\tau_E \leq t_0 + \sum_{t \geq t_0} (n-1)\left(\frac{80}{81}\right)^t \leq t_0 + 81.$$

Let $\tau_{\geq n-1} := \inf\{t : \operatorname{rank} G_t \geq n-1\}$. Since admission of the independent $L_1, \ldots, L_{n-1}$ implies rank at least $n-1$, $\tau_{\geq n-1} \leq \tau_E$. Thus the Random first stage reaches rank at least $n-1$ in expected at most $t_0 + 81$ settings. If $\operatorname{rank} G_{\tau_{\geq n-1}} = n$, recovery is already complete; otherwise its rank is $n-1$ and the residual argument below applies. The local bounded-weight sector is deliberately handled by Random itself, so the proof isolates the final rare-residual advantage.

Now condition on any correct $G \subset S_n$ of rank $n-1$, regardless of which relations were learned first. Since G is isotropic,

$$\dim(G^\perp/G) = [2n - (n-1)] - (n-1) = 2.$$

Thus Q_G has exactly three nonzero classes, while the one-dimensional true residual space S_n/G is exactly one of them. Every Pauli representative can be exposed by a product setting that matches its nonidentity letters. Choosing a nonzero quotient class uniformly therefore exposes a representative in the true residual class with probability $1/3$; if a different query also exposes the missing true relation, completion only occurs sooner. The conditional expected terminal cost is consequently at most 3, proving the first bound in Equation C.12.

For the Random lower bound, any complete learner must expose at least one member of the necessary coset $H_n + E_n$. Write $K_T = \prod_{i\in T} K_i$. In H_nK_T, site i is the identity exactly when $i \notin T$ and

precisely one Ring neighbour lies in T. Hence the one-setting mass of that coset obeys

$$B_n := \sum_{\substack{T\subseteq[n]\\ |T| \text{ even}}} 3^{-\operatorname{wt}(H_n K_T)} \le 3^{-n}\operatorname{tr}(M^n), \qquad M = \begin{pmatrix} 1 & 3 & 0 & 0\\ 0 & 0 & 1 & 1\\ 3 & 1 & 0 & 0\\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

With states ordered as $00, 01, 10, 11$, $M_{(a,b),(b,c)} = 3^{\mathbf{1}\{b=0,\ a\oplus c=1\}}$ and is zero otherwise. Its characteristic polynomial is $x(x^3 - 2x^2 - 8)$, so $B_n \le 4(\lambda/3)^n$ for the largest root $\lambda = 2.931142\ldots < 3$. Uniform Random settings are independent, and the probability of exposing any member of the necessary coset in one round is at most B_n. Its expected waiting time is therefore at least $B_n^{-1} \ge \frac{1}{4}(3/\lambda)^n$, proving the second bound. □

Proposition 1 isolates the intended composition. The Ring-derived local sector supplies a bounded-weight skeleton that is easy even for Random; the global coset supplies the exponentially rare completion event; and a generic codimension-one quotient query removes that final bottleneck without using a Ring label, a generator list, or the identity of the missing relation. It therefore provides a second witness of rare-residual complexity separation, with ordinary Ring end-to-end production behavior evaluated empirically.

What controls the tail. The size and arrangement of the unresolved relations control the tail in addition to total qubit count. At the same total size, Random finishes 0/50 GHZ-32 runs versus 10/50 four-block runs; Adaptive finishes 50/50 in both (Figure 2). These experiments test how bottleneck size and parallel organization affect recovery at fixed total size.

C.4 Structure-growing route I: learned-row matching at local boundaries

The local certificate can be computed for the production tableau itself. Let $K_i = Z_{i-1}X_iZ_{i+1}$ on an open corridor and $G_r = \langle K_1, \ldots, K_r\rangle$, with these local rows supplied to the compiler in the displayed order. Write $a = [K_{r+1}]$, $b = [K_{r+2}]$, and $W = \operatorname{span}\{a, b\} \subseteq S/G_r$.

Lemma 2 (Learned-row corridor frame)**.** *For $r \ge 2$, the deterministic learned-row completion in Algorithm 1 returns the four physical proposals*

$$q_1 = L_r, \quad q_2 = Z_0, \quad q_3 = Z_rX_{r+1}, \quad q_4 = Z_{r+1},$$

where

$$L_r = \begin{cases} X_0Z_1, & r \text{ even},\\ (X_0X_2\cdots X_{r-1})Z_r, & r \text{ odd}. \end{cases}$$

Their product-measurement conflict graph consists exactly of the two disjoint edges $q_1 - q_2$ and $q_3 - q_4$.

Proof. Site order. K_i touches $\{i-1, i, i+1\}$, so the learned-row incidence is 1 at the endpoints 0 and $r+1$, 2 at 1 and r, and 3 at each interior site $2, \ldots, r-1$. Decreasing incidence with physical-index tie-breaking therefore gives

$$\pi = (2, 3, \ldots, r-1,\ 1,\ r,\ 0,\ r+1),$$

whose last two positions are exactly the two incidence-1 endpoints.

The endpoints are never consumed. Site 0 is touched only by K_1, whose support $\{0, 1, 2\}$ meets π first at position 0, i.e. at site 2; so K_1 pivots at site 2 and passes over site 0. Likewise site $r+1$ is touched only by K_r, whose support $\{r-1, r, r+1\}$ meets π first at site $r-1$ (at site 1 when $r = 2$), so K_r pivots there. Hence sites 0 and $r+1$ are never consumed. Since $\operatorname{rank} G_r = r$ while $m = r+2$, these are exactly the two free positions, so by (S3) the free Z outputs are the bare single-site Z_0 and Z_{r+1}.

The interior pivots chain. By (S2) the pivots are consumed in π-order, and row K_k has support $\{k-1, k, k+1\}$: after $K_1, \ldots, K_{k-1}$ have consumed the first $k-1$ positions of π, the first unconsumed position at which K_k acts is position $k-1$. Induction on k therefore gives the pivot sequence

$$K_1 \mapsto \text{site } 2,\ K_2 \mapsto \text{site } 3,\ \ldots,\ K_{r-1} \mapsto \text{site } r,\ K_r \mapsto \text{site } r-1,$$

with every remaining letter of K_k cleared by an XCZ or XCX controlled at its own pivot. Reading the accumulated elimination backwards along this chain propagates X over the even-indexed interior sites, so the two free coordinates acquire exactly the conjugates

$$Z_0 \longleftrightarrow L_r, \qquad Z_{r+1} \longleftrightarrow Z_r X_{r+1},$$

with the alternating pattern $L_r = X_0 Z_1$ for even r and $L_r = (X_0 X_2 \cdots X_{r-1}) Z_r$ for odd r. Each free Z is an endpoint letter and each conjugate carries the opposite axis there, which is why the X string alternates.

Production keeps these tableau outputs as its physical proposal frame, with no quotient rebasing or coset-lift randomization. Each pair anticommutes at its endpoint, all cross-pair symplectic products vanish, and together with G_r the four rows have the required quotient rank. The displayed Pauli letters then give the two physical conflict edges directly. □

Proposition 4 (Production matching gain at a learned corridor)**.** *For the actual learned-row compiler and every $r \geq 2$,*

$$g_A^W = \frac{2}{9}, \tag{C.13}$$

$$g_A^W - g_R^W = \frac{1}{18} + \frac{20}{243}\left(\frac{2}{3}\right)^r - \frac{(-1)^r}{486\,3^r} > 0. \tag{C.14}$$

For $r = 1$, the corresponding margin is $5/36 - 1/9 = 1/36 > 0$.

Proof. By Lemma 2, a uniform random proposal order keeps one endpoint from each conflict edge; each of the four resulting packs occurs with probability $1/4$. The class a is exposed when q_3 is retained and the one remaining boundary axis is Z, so $p_A(a) = 1/6$. The class b requires q_4 and two random boundary axes, so $p_A(b) = 1/18$. The class $a + b$ is incompatible with these production packs. The three exposure events require distinct axes at the separating boundary and therefore contribute independent rank $2/9$ in expectation.

For Random, the complete cosets are counted rather than one representative. The corridor recurrence gives

$$p_1 := p_R(a) = \frac{1}{12} - \frac{4}{81}(2/3)^r + \frac{(-1)^r}{324\,3^r},$$
$$p_2 := p_R(b) = \frac{1}{18} - \frac{4}{243}(2/3)^r - \frac{(-1)^r}{486\,3^r}.$$

Multiplication by K_{r+2} adds one exterior match, hence $p_R(a+b) = p_1/3$, and the same boundary exclusivity gives $g_R^W = 4p_1/3 + p_2$. Substitution yields Equation C.14. Direct elimination of the four $r = 1$ proposals gives the stated initial margin. □

Finite-sample admission. For $r \geq 2$, the continuation classes $a, b, a + b$ have weights three, three, and four. Under the reported 1% readout channel and independent $256 + 256$ screening/confirmation blocks, Proposition 7 gives $\eta_3 \geq 0.99832$ and $\eta_4 \geq 0.99778$. The production-minus-Random contributions of a and b are positive, while only the absent production class $a + b$ contributes negatively. Since $\eta_3 \geq \eta_4$, the admitted-rank surplus is at least $\eta_4(g_A^W - g_R^W) > 0$. Thus finite-sample false rejection does not reverse the production matching certificate when the $[2, 5]$ window enumerates these relations.

C.4.1 Transfer beyond an isolated corridor

The same calculation gives a production-frame condition rather than assuming that quotient bases are interchangeable. For a target subspace W, define the Random coset-mass upper bound

$$U_G(W) = Z_{G+W}(1/3) - Z_G(1/3).$$

Corollary 5 (Learned-row boundary transfer)**.** *At any correct G in a larger graph, suppose the* actual *learned-row proposal list contains compatible cores $C_1, \ldots, C_k$ for independent local continuation classes, with random-order retention bounds $\rho_G(C_j)$ and r_j unmatched physical axes. Then*

$$g_A^W(G) \geq \sum_{j=1}^{k} \rho_G(C_j) 3^{-r_j}.$$

If this sum exceeds $U_G(W)$, the production query has a strict local gain over Random. Extra branches enter only through the actual blocker groups defining $\rho_G(C_j)$, the unmatched-axis counts r_j, and the coset mass $U_G(W)$.

Corollary 5 explains both transfer and failure: bounded external degree with limited shared-neighbor overlap preserves a small completion cost, whereas dense shared neighborhoods can add blockers or make Random combinations cheap. No graph or corridor label is supplied to the learner; the condition is used only to analyze the proposals generated from G.

C.5 Structure-growing route II: production joint capture in organized checks

Joint capture admits an equally direct learned-row certificate. Let $Q_{\mathrm{LR}}(G)$ be the physical proposal list actually returned by the production tableau. Choose independent target witnesses $P_1, \dots, P_m$ and decompositions

$$P_j = g_j + \sum_{q \in C_j} q + t_j, \qquad g_j \in G, \quad C_j \subseteq Q_{\mathrm{LR}}(G),$$

where each C_j is physically compatible and t_j lies outside the active support. Define

$$B_{\mathrm{LR}}(G, W) = \sum_{j=1}^{m} \rho_G(C_j) 3^{-\operatorname{wt}(t_j)}.$$

Theorem 4 (Production sector certificate)**.** *For the learned-row query,*

$$g_A^W(G) \geq B_{\mathrm{LR}}(G, W), \qquad g_R^W(G) \leq U_G(W).$$

Consequently $B_{\mathrm{LR}}(G, W) > U_G(W)$ certifies a strict same-G production gain. The certificate includes joint capture: a witness core may contain several proposals even when none is a target relation by itself.

Proof. Proposition 2 exposes witness j with probability at least $\rho_G(C_j) 3^{-\operatorname{wt}(t_j)}$. Independence of their target classes makes the sum a lower bound on expected exposed rank; the events need not be independent. For Random, union-bound each target class over all its representatives in $G + W$. Summing $3^{-\operatorname{wt}(P)}$ over $(G + W) \setminus G$ gives $U_G(W)$. □

Exact BB5 production instances. Apply the archived learned-row tableau and its actual blocker graph to successive correct BB5-30 spans generated by the following stored relations:

```
ZYIXIZXIIIXIIIIYIZIXIIZIIYIZXY
IIIZYIZXIIIZIIIIIZIXXIZZYIIIII
IIIIIIIIZIIZIIIZZZIIIIIIIZIIII
IZIIZIIZIIZIIZIIIIIIIIIIIIIIII
XIIIIIXIXIIXIIIIIXIIXIXXXXIXIX
```

The target sectors use independent weight-five pure-X checks. Every number below is an exact rational calculation on $Q_{\mathrm{LR}}(G)$, not a trajectory average.

$\dim G$	certified / available	B_{LR}	$U_G(W)$	ratio
1	13/13	251/1296	1864836089174/22876792454961	2.3758 …
3	12/13	49/144	1562710571072/22876792454961	4.9813 …
5	12/12	121/288	5816698716032/68630377364883	4.9571 …

The calculation enumerates the finite weight-five check space, expresses each witness in the production proposal frame, retains compatible independent cores, and applies the grouped-blocker bound in Equation B.1. It is reproduced by `learned_row_production_certificates.py`. All certified production witnesses have weight five. The implemented $256 + 256$ verifier has $\eta_5 \geq 0.99710$ under the reported 1% channel, so multiplying the production lower bounds by η_5 leaves admitted-gain ratios at least 2.369, 4.967, and 4.943 against the displayed Random exposure upper bounds. This conditional statement separates statistical admission from window recall: weight five is enumerated by $[2, 5]$, whereas missed higher-weight relations can still change which later spans are reached.

Theorem 5 (Transfer of production capture cores)**.** *Embed one or more certified regions into a larger target. If recompiling the* actual learned-row tableau *preserves the embedded witness cores, does not increase their grouped blocker counts or unmatched-axis costs, and preserves their independence modulo the enlarged G, then their production lower bounds are preserved and add across independent target sectors. Comparing the sum to the enlarged coset mass $U_G(W)$ certifies transfer of the strict gain.*

Proof. Each hypothesis preserves the corresponding factor in $\rho_G(C_j)3^{-\operatorname{wt}(t_j)}$; independent embedded witnesses add in expected rank. Theorem 4 then applies in the enlarged learned-row frame. The explicit coset mass accounts for any new Random representatives created by the interface. □

Theorem 5 requires preservation of the physical production cores and their blockers. Frozen-G learned-row replay checks the same quantities on states reached by the algorithm, while the end-to-end BB5 and randomized sparse-check cohorts test their accumulation.

D The Symmetry Boundary and Why Generic Lookahead Is Not Free

The positive results use physical organization inherited from verified relations. This section separates that structural bias from a generic *maximize information gain* prescription. The distinction is important: G defines the exact feasible completion set, but not a probability law over that set.

D.1 Maximum-entropy information gain conditioned only on G

Consider the symmetry-preserving maximum-entropy model in which, conditional on G, the unknown residual target L_G is uniform over the Lagrangian subspaces of Q_G. A complete product setting s induces the Lagrangian slice $E_G(s)$ defined in Appendix A.

Proposition 5 (A G-only maximum-entropy lookahead is flat)**.** *Under the uniform completion law, the full distribution of*

$$\Delta_G(s) = \dim\big(L_G \cap E_G(s)\big)$$

is identical for every complete product setting. Consequently, exact one-step expected rank gain and the corresponding noiseless information-gain score cannot rank settings. Uniform tie breaking is exactly uniform Random.

Proof. The symplectic group of Q_G acts transitively on Lagrangian subspaces and preserves the uniform law of L_G. For any two settings s, s', some symplectic map sends $E_G(s)$ to $E_G(s')$; applying it simultaneously to the random completion preserves its law and the intersection dimension. The entire rank-gain distribution is therefore setting independent. In the noiseless complete-extraction model, the posterior reduction produced by a setting is determined by the exposed intersection and its signs, whose symmetric sign law is likewise setting independent. □

This is the strongest posterior-free information-gain baseline induced by the feasible set alone, and it collapses analytically across all settings. An information-gain, Bayesian, particle-filter, or neural policy can become directional only by adding a nonuniform target prior, a likelihood state retaining statistics beyond verified relations, or training data from a target distribution. Such policies are valid extensions with additional predictive state beyond the verified-G interface studied here.

Cost of exact generic Bayesian lookahead. Modulo phase, the unrestricted Pauli universe contains 4^n strings, the product-setting space has size 3^n, and one complete setting exposes 2^n parities (with $\binom{n}{w}$ at weight w). More directly, a $2d$-dimensional residual symplectic space has $\prod_{j=1}^{d}(2^j+1)$ Lagrangian completions. Thus an exact arbitrary posterior over candidates, settings, or residual stabilizer completions is exponential in the worst case. Tractable directional approximations therefore add a modeling choice—for example a bounded candidate window, a factorized likelihood, a finite particle family, a parametric target-family prior, or training data—and therefore changes either the information, approximation, or computational budget.

The screening and confirmation statistics do not provide such a lookahead for free. They score candidates measurable in settings that were already drawn; assigning probabilities to unseen settings still requires a candidate universe and a likelihood or structural prior. A bounded weight-$w_{\max}$ universe is polynomial, $O(n^{w_{\max}})$, but incomplete: the noisy tail diagnostics in Appendix G show

exposed residual relations outside the active window. Independent candidate probabilities also discard the commutation, span, and product dependencies that define a stabilizer space. Finite-particle or factorized policies are legitimate approximations, but do not inherit a worst-case guarantee of representing the unrestricted posterior.

Moreover, confirmation remains fresh precisely so that adaptively proposed candidates can be validated independently. One may retain a separate soft posterior from screening data and optimize it, but that creates an extractor- and noise-model-specific policy state beyond G. The verified span is not the only conceivable polynomial state; it is the exact, sound, polynomial-size state available without selecting an additional target model. Our prior-informed control makes such extra structure explicit: Mixed Random assigns substantial mass to the three global same-axis settings. It transfers to BB5 but not Ring (Appendix F.7.1), illustrating both the potential and the family dependence of a nonuniform prior.

Proposition 5 does *not* say that every G-only policy is equivalent. Algorithm 1 deliberately breaks the abstract symmetry using the physical supports and Pauli letters of the relations that were actually discovered. The graph and check results identify when this inductive bias predicts useful future exposure.

Conflict-Degree-Ordered Greedy Packing (CDGP) provides an explicit stronger G-only comparator. On the same quotient proposals, it repeatedly selects a uniformly tie-broken minimum-degree vertex of the residual physical conflict graph and deletes its closed neighborhood; all free axes are then completed as usual. It therefore uses no target prior while optimizing a genuine observable surrogate. The resulting set is maximal and has size at least $m/(\Delta + 1)$ for m proposals and maximum conflict degree Δ, but degree alone does not control target capture $\nu_G(C)$ or the exploration it displaces. Seven paired 50-run suites confirm the distinction across GHZ-32, Ring-32/64, BB5-30/48, Toric, and local HGP: both policies complete all $350/350$ runs, and every target-level paired interval includes zero. Giving each target equal weight, the CDGP-minus-random-order effects are -0.14 settings (95% CI $[-0.60, 0.31]$) and $+0.042$ IRD ($[-0.098, 0.178]$). Table 13 reports every target.

D.2 Simple G-only optimization controls

The posterior-free alternatives in the experiments optimize observable surrogates. Reuse-G maximizes direct agreement with verified directions; WeightSort prioritizes the lightest quotient representatives; finite pack caps control commitment; and full greedy packing produces a maximal compatible set in one scan. Reuse-G and WeightSort both stall, while the randomized learned-row compiler completes the structured targets. Thus neither repeated agreement, individual weight, nor pack cardinality is a substitute for the unknown target-rank objective.

For each realized pack, Equation B.2 gives the exact target-scored decomposition

$$\Gamma_K - \Gamma_0 = \underbrace{\kappa_K}_{\text{jointly captured structure}} - \underbrace{(\Gamma_0 - \chi_K)}_{\text{displaced completion}} .$$

The target is used only offline to evaluate these terms. On production-reachable BB5-48 spans, full packing exceeds $K = 1$ at residual dimensions $40, 22, 6$ by $1.71, 1.76, 0.51$ expected ranks, respectively; the capture term repays the lost completion at every stage. This directly evaluates the objective that maximum-cardinality packing cannot observe.

The resulting boundary is precise. Quotient construction always removes resolved redundancy. A directional acquisition advantage additionally requires either matching compression or joint capture tied to physical organization. Without a predictive law or such organization, generic lookahead is flat; with an externally supplied prior, the comparison becomes a different learning problem.

E Experiment Protocols, Statistical Units, and Targets

E.1 Protocol at a glance

All measurements use fresh copies, local $X/Y/Z$ basis choices, and classical readout. We use `stim` for stabilizer sampling. Target generators and membership queries are used for simulation and offline scoring only. The policy sees measurement outcomes, accepted relations, and the declared readout error rate. It receives no family label or generator list. Table 4 summarizes budgets and statistical units.

Phase A splits discovery and validation into independently sampled pools. For noiseless extraction, subtract the first outcome from every other outcome over $\mathbb{F}_2$ and compute the full right nullspace

Table 4: **Central protocol ledger.** A setting uses the listed total copies, including validation. Stops are upper limits; rank-n recovery stops a run early unless the row specifies a fixed horizon.

Experiment	Repetitions / inference unit	Copies per setting	Stop and comparison
Structured recovery / coordinate controls	50 acquisition runs per target and policy; run-level summaries	$1024 + 128 = 1152$	At most 50 settings; separate evolving G per policy
Random-target graph/check cohorts	Two declared families $\times$ 12 independently drawn targets $\times$ 20 runs per policy; target is the outer unit	$1024 + 128 = 1152$	Fixed seeds chosen before acquisition; at most 50 settings
Random-Clifford boundary	20 independently drawn targets paired across policies; target is the outer unit	$1024 + 128 = 1152$	At most 50 settings; no structured-family claim
Partial-stabilizer control	50 matched run IDs per policy; run-level summaries	$1024 + 128 = 1152$	Fixed 50 settings; rank-16 completion is scored offline, not supplied to the policy
Frozen BB5-48, multiple origins	10 learned-row origins $\times$ 3 stages; 500 replays per policy	Exact conditional exposure replay	Same G across Random, $K = 1$, $K = 3$, and full packing; origins are the inference unit
Noisy window comparison	50 runs per window and policy, $p = 0.01$	$768 + 256 = 1024$	At most 50; same acquisition rule, different searchable windows
Low-weight + Phase B	50 run IDs matched to low-weight baseline	A: $768+256$; B: 1024 validation	Separate A/B caps 50/15; report actual consumption, not equal-horizon completion

of this difference matrix. Every returned support is mapped to a Pauli using the setting; no weight filtering or subset-minimality pruning is applied to this basis. The separate 128-row batch confirms candidates with the noise-aware verifier at $p = 0$, hence acceptance threshold 0.75. Its confirmed sign is stored with each rank-increasing relation. Finite discovery can produce accidental extra constraints, which motivates independent validation. Exposure theory describes the true measurable relations; the implemented update depends on which candidates are recovered and accepted.

Noisy screening uses 256 rows from the 768-row discovery pool; the separate 256-row pool is reserved for confirmation. Phase B fixes its packed candidates before collecting a new 1024-row validation pool. No reported main protocol adds candidate-dependent verification shots. Model-training configuration fields in archived files are inactive when the extractor is GF(2)-based.

For acquisition run ID j, the base pseudorandom seed is $20260428 + 1000j$. The setting generator uses this base seed. At iteration t, discovery, confirmation, and screening use seeds base $+100000t+1$, $+100000t + 2$, and $+100000t + 3$, respectively; Phase B uses base $+100000t + 50000$ for its fresh batch. These schedules reproduce sampling conditional on a fixed target.

E.2 Metrics, stopping, and uncertainty

True rank is the rank of accepted relations that belong to the target, whereas internal rank counts all accepted independent relations. Plot curves are median true rank divided by n, with interquartile bands; completed trajectories stay at rank n thereafter. Full-recovery counts and terminal rank are reported alongside settings. Mean completion settings are used when all runs finish; otherwise a success-conditioned mean is explicitly labeled. Actual settings consumed under a stopping rule are not an uncensored mean completion time.

The integrated rank deficit is $D_{50} = \sum_{t=1}^{50}(1 - r_t/n)$, with zero contribution after recovery. For structured fixed targets, resample runs independently within each policy. Confidence intervals are targetwise rather than simultaneous familywise intervals. The breadth conclusion is based on the consistent direction across all 15 reported structured targets, not on selecting individually significant comparisons.

For each random-target cohort, the target is the outer statistical unit. We first average within each of 12 independently generated targets and then bootstrap targets and acquisition runs hierarchically (20,000 percentile resamples, seed 20260925). Each average claim is restricted to its declared connected Erdős–Rényi graph or random-support sparse HGP/CSS distribution.

Table 11 was recomputed from the archived raw results with 10,000 independent bootstrap resamples, NumPy generator seed 20260921, and percentile 95% intervals. The matched Phase B comparison instead uses run IDs, 20,000 paired resamples, and its recorded actual budgets. The partial-stabilizer control likewise uses 20,000 paired percentile-bootstrap resamples (seed 20260924) over its 50 matched run IDs. The Random-Clifford boundary uses 20,000 target-paired percentile-bootstrap resamples (seed 20260925) over its 20 independently drawn targets. Shared run IDs do not imply identical settings or identical evolving G. Frozen-state contrasts hold the actual representation fixed; repeated states from one trajectory should not be treated as independent experimental targets. Stage summaries in Appendix F.9 are descriptive conditional statistics, separate from the recovery confidence intervals.

E.3 Data and implementation map

The archived result metadata, rather than current script defaults, specifies the protocol. Paths below are relative to the experiment directory.

Result	Source
Learned-row recovery	`rawg_generalization_20seeds/results` with the 50-run compact export in `compact_first50.json`
Broad random graph/CSS cohorts	`broad_random_families_20260925`: generators, 960 atomic trajectories, hierarchical summaries, and a frozen result archive
Random-Clifford boundary	`rawg_generalization_20seeds/results/random_clifford32`: 20 paired targets and trajectories per policy
RREF/free-column compiler	`results_*_50seeds`: 50-run same-protocol Random, $K = 1$, and RREF trajectories for the principal targets
G-only CDGP baseline	`cdgp_baseline_20260924/results`: 50 paired full/CDGP runs on each of seven bottleneck, graph, and code/check targets
Partial stabilizer structure in a mixed state	`partial_relation_20260924/results`: 50 matched Adaptive/Random trajectories, fixed horizon, plus paired analysis summary
$K = 1$, $K = 3$, and compiler controls	`learned_row_completion_20260923/results`; atomic files for every target, variant, and run ID
Multiple-origin replay	`learned_row_completion_20260923/frozen_shards`: ten origin files, three ranks, 500 replays
Noisy windows and packed tail	`learned_row_completion_20260923/noisy_results`: 50 atomic run files per condition
Implementation	`learned_row_completion_20260923/core/ICLR`: archived policy, learner, extractors, and verifier

The original 4096+2048-copy centralizer-only and candidate-budget cohorts are not pooled with the 1152-copy noiseless or 1024-copy noisy main experiments. The source inventory and per-file metadata are retained in `EXPERIMENT_EVIDENCE_INDEX.json`. The deterministic reanalysis is `analysis/revision_statistics.py` in the accompanying analysis material. The anonymous artifact mirrors these relative paths, removes internal machine paths, records file hashes and environment metadata, and includes frozen JSON inputs sufficient to regenerate every main-text figure and table without rerunning the simulations.

E.4 What is counted as a resource

Quantum resources are settings and total copies. Candidate evaluations count classical parity scoring, separately from quotient construction, packing, and external confirmation. Existing runtime logs time the full simulator loop, including sampling and target-aware diagnostics; these logs document feasibility-scale wall time. Pure classical compiler costs are reported separately through analytic scaling and candidate counts. Phase B reduces total candidate scoring from 431,306,038 to 421,539,864 in the 50-run cohort. The end-to-end reduction is modest because early Phase A dominates, while the additional tail settings score only $562/459 \simeq 1.22$ preselected candidates each instead of enumerating 174,406 low-weight supports.

Observed 64-qubit wall time and analytic classical scaling. The archived one-thread CPU logs provide a practical scale check, while keeping their scope explicit. Across the nine structured 64-qubit

targets, target-wise median full-trajectory times range from 1.71 to 2.72 seconds for Adaptive and 1.42 to 2.69 seconds for Random; median time per consumed setting ranges from 0.102 to 0.146 seconds and from 0.053 to 0.083 seconds, respectively. These are 30–50 Adaptive and 50 Random trajectories per target run across the same two CPU servers; they include outcome simulation and diagnostic scoring, so they demonstrate feasibility at 64 qubits rather than isolating deployment latency.

The policy state and its binary linear algebra require $O(n^2)$ storage; naive tableau elimination is $O(n^3)$ time, and the random-greedy scan is $O(m|A|)$ for $m = O(n)$ proposals on active support A. No peak-RSS trace was recorded. Noisy extraction is a separate cost: a Phase-A window W scores $\sum_{w\in W}\binom{n}{w}$ measurable supports per setting, whereas Phase B scores only its preselected packed proposals. Thus the logs and asymptotics separate the polynomial acquisition compiler from the standard bounded-weight parity search discussed in Appendix G.

E.5 Experimental states and stabilizer generators

The generator descriptions below are reserved for target construction, interpretation, and offline analysis. Acquisition itself receives measurement outcomes and its evolving verified span. They make the structural regimes easy to inspect and clarify why a fixed global-axis prior aligns differently across targets. In every case the listed independent commuting generators fix the pure stabilizer state used in the experiments.

Table 5: **Generator structure of the experimental targets.** These descriptions support interpretation and reproducibility; acquisition accesses the target only through measurement outcomes.

Target / instance	Independent stabilizer generators	Structural feature relevant to acquisition
GHZ-n	$(n-1)$ weight-2 ZZ relations and one weight-n global $X^{\otimes n}$ direction	Local Z sector plus a global rare sector
$\mathrm{GHZ}_8^{\otimes 4}$	Four disjoint copies of seven weight-2 ZZ relations plus one weight-8 $X^{\otimes 8}$ direction	Four independent global-bottleneck blocks
Ring-n	n weight-3 generators $Z_{i-1}X_iZ_{i+1}$	Local but mixed-Pauli ZXZ structure
BB5-30	13 weight-5 X checks, 13 weight-5 Z checks, 4 additional logical-Z stabilizers	CSS, strongly same-axis compatible
BB5-48	22 weight-5 X checks, 22 weight-5 Z checks, 4 logical-Z stabilizers of weights $8, 8, 12, 12$	CSS, strongly same-axis compatible

GHZ states. The n-qubit GHZ stabilizer group is

$$S_{\mathrm{GHZ},n} = \langle Z_1Z_2,\ Z_2Z_3,\ \ldots,\ Z_{n-1}Z_n,\ X_1X_2\cdots X_n\rangle,$$

with rank n. The first $n-1$ generators are local weight-2 Z-type relations, while $X^{\otimes n}$ supplies the independent global non-Z direction.

$\mathrm{GHZ}_8^{\otimes 4}$. The 32-qubit block-product state is the tensor product of four independent 8-qubit GHZ states. Each block is generated by seven nearest-neighbor ZZ relations and one weight-8 $X^{\otimes 8}$ relation. The four blocks therefore supply $4 \times 8 = 32$ independent commuting generators in total.

Ring graph states. The n-qubit ring graph state is defined on the cycle C_n and generated by

$$K_i = Z_{i-1}X_iZ_{i+1}, \qquad i = 1, \ldots, n,$$

with indices modulo n. Equivalently, the implementation uses the shifted convention $Z_iX_{i+1}Z_{i+2}$. All n generators have weight 3 and form a full-rank basis. Unlike the BB5 instances below, each canonical generator mixes Pauli axes locally.

BB5-30. BB5-30 is derived from a $[[30, 4, 5]]$ CSS code. The code stabilizer contributes 13 independent weight-5 X-type checks and 13 independent weight-5 Z-type checks. We adjoin four independent logical-Z operators whose literal implementation weights are $5, 5, 8, 8$, fixing all four logical degrees of freedom and thereby obtaining a unique pure 30-qubit stabilizer state. The final state is therefore specified by

$$13\ X\text{-checks}\ +\ 13\ Z\text{-checks}\ +\ 4\ \text{logical-}Z\ =\ 30$$

independent commuting generators, all with eigenvalue $+1$. Every generator is purely X-type or purely Z-type (with identities off its support), which makes the instance especially compatible with globally aligned X or Z measurement settings. The exact Pauli strings used in the experiments are fixed in the accompanying implementation.

BB5-48. BB5-48 is a separate code-derived instance constructed analogously from a $[[48, 4, 7]]$ CSS code; it is not treated as a second size point of a common BB5 scaling family. Its code stabilizer contains 22 independent weight-5 X-type checks and 22 independent weight-5 Z-type checks; four logical-Z operators of weights $8, 8, 12, 12$ are then adjoined to obtain 48 independent commuting generators and hence a unique pure stabilizer state. It supplies an additional recovery instance, multiple-origin frozen-state tests, and the qubit-label control of Appendix F.8.

For reproducibility, the BB5-48 checks are generated algebraically with $\ell = 8$ and $m = 3$. Let Q_8 and Q_3 denote cyclic-shift matrices and define

$$\begin{aligned} A_1 &= I_{24}, & A_2 &= Q_8 \otimes I_3, \\ A_3 &= I_{24}, & A_4 &= I_8 \otimes Q_3, \\ A_5 &= Q_8^3 \otimes Q_3^2. \end{aligned}$$

The CSS check matrices are

$$H_X = [A_1 + A_2 \mid A_3 + A_4 + A_5], \qquad H_Z = [A_3^T + A_4^T + A_5^T \mid A_1^T + A_2^T],$$

with arithmetic over $\mathbb{F}_2$. Independent-row extraction gives rank 22 for each check family. The four added logical-Z directions are chosen from $\ker(H_X)/\operatorname{rowspan}(H_Z)$. This yields the generator counts and weights reported in Table 5.

Relation to the fixed same-axis preference control (Mixed Random). Mixed Random allocates probability $0.1/3$ to each global axis, in addition to its uniform branch. Thus a specified global-axis setting has probability $0.9 \cdot 3^{-n} + 0.1/3$. This fixed policy uses neither target generators nor measurement outcomes. Its inductive bias happens to align with CSS structure more than with mixed-axis graph-state relations. It tests how one strong fixed prior transfers across target organizations.

Additional breadth targets and instance variation. The remaining graph-state targets are generated from their graphs by $K_v = X_v \prod_{u \in N(v)} Z_u$. Open-chain-64 uses the path backbone; range-two-chain-64 adds every distance-two edge; range-two-ring-64 makes this construction periodic; and Grid-8×8 uses the open square lattice. The three irregular-chain targets are genuinely different target instances from a declared distribution: each retains the 64-site open-chain backbone and includes every possible distance-two edge independently with probability $1/2$, using seeds 2026092201, 2026092202, and 2026092203. Repeated acquisition runs condition on each frozen draw. They therefore provide a limited within-family transfer check, while the full table tests breadth across named constructions; neither is presented as population-level generalization.

The broader random graph cohort contains 12 connected 64-vertex Erdős–Rényi graphs with requested mean degrees 4, 6, and 8 (four targets per stratum). There is no chain, ring, lattice, or other fixed backbone. Seeds 2026092500–2026092511 and all ensemble parameters were fixed before either policy was run; each target receives 20 Learned-row and 20 Random trajectories.

The broader random HGP/CSS cohort independently samples full-row-rank $H_1 \in \mathbb{F}_2^{4\times 8}$ and $H_2 \in \mathbb{F}_2^{4\times 6}$ with uniform row supports, not cyclic local intervals. Row-weight pairs cycle through $(3, 3), (3, 4), (4, 3), (4, 4)$. The standard hypergraph-product construction has 64 physical qubits, 24 independent X checks, 32 independent Z checks, and eight logical degrees of freedom. A deterministic lowest-weight-first logical-Z completion fixes a unique state without consulting acquisition outcomes. Seeds 2026092600–2026092611 were fixed before either policy was run; all 12 check-matrix pairs are distinct and receive 20 runs per policy.

Toric-4×8 is the fixed rectangular toric-code state described by the archived star/plaquette generators and logical-state fixers. Local-HGP-64 is a fixed hypergraph-product construction from full-row-rank 4×8 and 4×6 binary matrices whose rows contain overlapping local triples. It is included as an organized local-check target; no distance or asymptotic HGP claim is made. Exact generators, seeds, hashes, and validation ranks are stored with the archived target JSON files.

F EXPERIMENTAL CONTROLS AND ADDITIONAL RESULTS

The controls separate information access, the use of learned relations, quotient coordinates, and exploration. Frozen-state experiments condition on the same learned representation; independent rollouts retain their own evolving representations.

F.1 CLAIM–CONDITION–EVIDENCE LEDGER

Table 6 records each experiment's role without interrupting the main paper's learning-loop narrative. Recovery tests use the same target-uninformed acquisition rule. Frozen-state tests hold learned information fixed; full rollouts retain their own histories.

Table 6: **Mechanisms, controlled conditions, and evidence.** Read each row horizontally as claim → controlled condition → supporting result. Numerical arrows compare Random with Adaptive, except for the labeled Phase-B extension.

Mechanism / claim	Controlled condition	Evidence
Direct stabilizer-type constraint discovery	Fixed product-Pauli access; unknown Pauli-relation list	Batch elimination discovers constraints without a dense state or exhaustive support queries; reuse begins before full recovery.
Scale-changing bottleneck	GHZ-32 versus $\mathrm{GHZ}_8^{\otimes 4}$: same n, different global-relation scales	Production block contraction plus terminal exposure gives Corollary 1; Random completes 0/50 versus 10/50 and Adaptive 50/50 on both.
Structure-growing local matching	Ring-32/64: same local generator type, doubled size	Exact learned-row production certificate (Prop. 4); completion: $21.52 \to 14.64$, $24.76 \to 16.34$.
Structure-growing joint capture	BB5-30/48: exact learned-row blockers plus frozen production G	BB5 production certificate and frozen-G capture (Fig. 3); recovery: $29.82 \to 17.24$, 42/50→50/50.
Useful reuse requires organization and exploration	Vary fixed prior, feedback direction, coordinates/lifts, proposal order, and pack cap	Maximum-entropy G-only information gain is exactly flat; across seven targets, conflict-degree packing has no stable gain over random order; Mixed Random helps BB5 but not Ring; Reuse-G and weight-first policies stall.
Transfer without family labels	Fifteen fixed targets plus 12 connected random graphs and 12 random-support sparse HGP/CSS draws	Adaptive completes all 750 fixed-target runs and lowers target-balanced D_{50} on 12/12 targets in each randomized cohort; both hierarchical intervals exclude zero.
Partial stabilizer structure without a stabilizer-complete state	Ring-16 tensored with 16 mixed non-stabilizer spectators; fixed 50-setting horizon	Both policies recover the exact rank-16 stabilizer relation space in 50/50 runs; Adaptive reduces mean completion from 18.52 to 13.40 settings with no false admissions.
Extractor–policy coupling and targeted tail search	BB5-30, 1% readout flips; window interventions and packed Phase B	Windows change trajectories; Phase B raises low-weight recovery 8/50→29/50 under its stopping rule.

F.2 TARGET-RANDOMIZED TRANSFER IN GRAPH AND CHECK FAMILIES

The 50 repetitions in the headline table quantify acquisition randomness conditional on a target; they are not 50 target draws. We therefore generated two broader 64-qubit cohorts, with 12 independent targets per family and 20 Learned-row and 20 Random trajectories per target under the unchanged 1152-copy, 50-setting protocol. Seeds and ensemble parameters were fixed before either policy was run.

Each graph is sampled from $G(64, p)$ conditional on connectivity, with $p \in \{4/63, 6/63, 8/63\}$ and no chain, ring, or lattice backbone. Learned-row lowers mean D_{50} on all 12 targets. The hierarchical interval resamples both targets and runs. Degree eight exposes a finite-budget boundary: the rank trajectory remains substantially better than Random, but only 42/80 Learned-row runs finish. No false admission or rejection occurs.

Table 7: **Connected Erdős–Rényi graph-state stress test.** Each stratum contains four independently drawn 64-qubit targets and 20 runs per policy and target. ΔD_{50} is Learned-row minus Random; completion is Learned-row / Random at 50 settings.

Requested mean degree	Targets	ΔD_{50}	Complete	Targets improved
4	4	-6.802	$80/80$ / $71/80$	$4/4$
6	4	-13.358	$78/80$ / $2/80$	$4/4$
8	4	-10.288	$42/80$ / $0/80$	$4/4$
All	12	-10.149 $[-11.821, -8.437]$	$200/240$ / $73/240$	$12/12$

The second cohort draws independent sparse HGP/CSS check layouts. Each target uses full-row-rank $H_1 \in \mathbb{F}_2^{4\times 8}$ and $H_2 \in \mathbb{F}_2^{4\times 6}$ whose row supports are uniform subsets rather than cyclic intervals. Row-weight pairs cycle through $(3,3)$, $(3,4)$, $(4,3)$, and $(4,4)$. The construction supplies 24 independent X checks, 32 independent Z checks, and eight logical degrees fixed by a deterministic logical-Z completion.

Table 8: **Random-support sparse HGP/CSS stress test.** Each stratum contains three independent 64-qubit targets and 20 runs per policy and target. ΔD_{50} and completion follow Table 7.

Row weights (H_1, H_2)	Targets	ΔD_{50}	Complete	Targets improved
$(3,3)$	3	-3.467	$60/60$ / $58/60$	$3/3$
$(3,4)$	3	-4.504	$60/60$ / $57/60$	$3/3$
$(4,3)$	3	-4.721	$60/60$ / $58/60$	$3/3$
$(4,4)$	3	-6.034	$60/60$ / $58/60$	$3/3$
All	12	-4.682 $[-5.280, -4.081]$	$240/240$ / $231/240$	$12/12$

Table 8 shows lower D_{50} on all 12 check targets and every row-weight stratum, with no false admission or rejection. Together the two cohorts remove the fixed geometric templates used by the named examples and provide separate target-level stress tests for graph matching and organized check capture across sparse organized regimes. Dense random codes and arbitrary stabilizer states form separate regimes.

F.3 Unstructured Random-Clifford boundary

The method is designed to inherit stabilizer-type structure whose physical organization persists in verified rows. To test the opposite boundary, we use 20 independently drawn 32-qubit stabilizer states generated by random Clifford circuits. Each target is paired across Learned-row and Random under the same 1152-copy, 50-setting protocol. The target is the statistical unit.

Table 9: **Random-Clifford boundary control.** D_{50} is lower better; settings are reported only for runs completing by the 50-setting horizon.

Policy	Mean D_{50}	Complete	Success-conditioned settings
Random	22.26 ± 3.18	$14/20$	42.57
Learned-row (Adaptive)	20.70 ± 4.07	$10/20$	40.70

Table 9 gives a paired Learned-row-minus-Random D_{50} difference of -1.56 with a 20,000-resample target-paired bootstrap interval $[-3.35, 0.20]$; 14 targets favor Learned-row in D_{50} and six favor Random, while Random completes more within the horizon. The two metrics therefore give mixed ordering across this cohort. Unlike the connected-graph and sparse-check cohorts, a random Clifford circuit supplies no persistent local or check organization for the compiler to inherit. Under this isotropic orientation, random physical exploration is already matched to the target distribution, whereas packing can fix axes without a reliable capture surplus. We use this cohort as the empirical symmetry-boundary control, separate from the structured-target breadth suite.

F.4 Partial stabilizer relations in a mixed non-stabilizer state

The main recovery suite uses pure stabilizer states so that full rank gives an unambiguous endpoint. To test whether the *entire state* must be stabilizer-complete, while keeping the learned structure explicitly within the method's stabilizer-type target class, we use the 32-qubit product state

$$\rho_{\mathrm{partial}} = |\mathrm{Ring}_{16}\rangle\langle\mathrm{Ring}_{16}| \otimes \rho_M^{\otimes 16}, \qquad \rho_M = \tfrac{1}{2}(I + 0.4X + 0.4Y + 0.4Z). \tag{F.1}$$

The spectator is physical and mixed because its Bloch-vector norm is $\sqrt{0.48} < 1$, and it lies outside the single-qubit stabilizer octahedron because $|0.4| + |0.4| + |0.4| = 1.2 > 1$. Every nonidentity spectator Pauli has expectation magnitude $0.4 < 1$; hence the exact deterministic Pauli-relation space of (F.1) is precisely the rank-16 Ring stabilizer space padded by identities. The global state is therefore mixed, non-stabilizer, and not stabilizer-complete.

The policy receives only measurement outcomes and its verified span; the family, generators, and exact rank 16 are withheld. Each policy runs for a fixed 50-setting horizon using the same product-Pauli sampler, GF(2) extractor, independent $1024 + 128$ discovery and confirmation batches, and learned-row compiler as the main noiseless experiments. Rank 16 is used only offline to score full exact-relation recovery; the run continues afterward, providing an audit for spurious spectator relations.

Table 10: **Partial stabilizer-relation recovery on a globally mixed, non-stabilizer target.** Means are over 50 runs; parenthesized values are sample standard deviations. Partial IRD normalizes rank by the true exact-relation rank 16 over the fixed 50-setting horizon.

Policy	Complete	Settings to rank 16	Partial IRD	False admissions
Random	50/50	18.52 (3.98)	6.97 (1.57)	0
Learned-row (Adaptive)	50/50	13.40 (3.95)	5.49 (1.68)	0

Table 10 gives a paired Adaptive-minus-Random completion-setting difference of -5.12 with a 95% paired-bootstrap interval $[-6.70, -3.52]$, a 27.6% reduction (95% interval $[19.8\%, 34.7\%]$). The partial-IRD difference is -1.48 with interval $[-2.08, -0.86]$. All 100 internal spans finish at rank 16 and match the true relation rank, with zero false admissions or rejections. Thus the policy discovers and reuses partial exact stabilizer structure even when the full state is not a stabilizer state. Its learned object is precisely the rank-16 relation space; the remaining non-stabilizer degrees of freedom lie outside that representation.

F.5 Alternative quotient-coordinate compiler

The following 50-run experiment uses the same product-Pauli interface, verified-G input, GF(2) extractor, $1024 + 128$ copies per setting, and 50-setting horizon as the learned-row results. It changes the core coordinate construction: a deterministic RREF/free-column frame replaces the learned-row tableau completion, after which the same random-greedy physical realization is used. It is therefore a direct same-information compiler baseline, and also the frame in which several exact structural certificates in Appendix C are most transparent. The comparison highlights complementary structure

Table 11: **Integrated rank deficit for the archived RREF-frame diagnostic (lower is better).** Intervals are percentile 95% bootstrap intervals over 50 runs per structured target and policy.

Target	Random	$K = 1$	RREF	Random$-K{=}1$	$K{=}1-$RREF
GHZ-32	3.55	2.35	2.56	1.21 $[1.04, 1.37]$	-0.22 $[-0.40, -0.03]$
Ring-32	6.99	6.75	5.95	0.24 $[-0.24, 0.72]$	0.80 $[0.28, 1.32]$
BB5-30	13.35	13.23	6.70	0.12 $[-0.93, 1.16]$	6.53 $[5.65, 7.42]$

sensitivity. RREF is stronger on BB5-30 (14.82 ± 4.39 versus 17.24 ± 2.85 settings) and BB5-48 (16.30 ± 3.21 versus 19.46 ± 3.18), while learned-row is stronger on Ring-32 (14.64 ± 3.45 versus 16.98 ± 4.06) and changes range-two-ring completion from $12/50$ to $50/50$ (learned-row mean 20.64 ± 3.17 settings). Thus the old frame is both a serious baseline and a useful analytic coordinate system; the learned-row production choice is supported by transfer across the broader structured suite.

F.6 Frozen-Representation Snapshot Protocol

The frozen-G intervention uses representations encountered along ten actual BB5-48 learned-row trajectories rather than synthetic subspaces. For each trajectory we retain the first snapshots at ranks 8, 26, and 42 (residual dimensions $d = 40, 22, 6$), hold that verified span fixed, and replay Random, $K = 1$, $K = 3$, and full greedy packing 500 times each. Thus every within-snapshot contrast sees exactly the same unresolved geometry and differs only in how that geometry is compiled into one product-Pauli setting. The resulting capture–completion decomposition is reported in Table 15 and Figure 3; it is a conditional mechanism test, while Table 14 separately measures end-to-end learning.

F.7 Diagnostic Hierarchy of Information Use

The controls distinguish a fixed acquisition prior (Mixed Random), feedback toward already learned relations (Reuse-G), and interventions on residual coordinates and traversal. Independent rollouts use the same access model, not an identical evolving G; frozen-representation interventions hold G fixed.

F.7.1 Fixed same-axis preference: Mixed Random

At each round, Mixed Random samples uniform product-Pauli settings with probability 0.9 and a uniformly chosen global X, Y, or Z setting with probability 0.1. It uses neither outcomes nor G_t and has no support-weight parameter. For a complete setting $s \in \{X, Y, Z\}^n$,

$$q_{\mathrm{Mixed}}(s) = 0.9\, 3^{-n} + \frac{0.1}{3}\, \mathbf{1}\{s \in \{X^n, Y^n, Z^n\}\}. \tag{F.2}$$

Under Equation F.2, each specified global same-axis setting receives probability $0.9\, 3^{-n} + 0.1/3$, rather than Random's 3^{-n}. The three together receive $0.1 + 0.9\, 3^{1-n}$. Thus a constant mixture component is concentrated on three of the 3^n legal settings, while all others retain 90% of their Random probability. A chosen setting exposes every compatible relation, irrespective of its weight; the prior acts on complete settings, not a selected relation-weight class.

The preference aligns with pure-axis check structure while leaving target checks, support locations, and family labels hidden. It therefore provides a strong fixed structural prior. Table 12 shows that this prior helps on BB5-30 and slightly hurts Ring-32, demonstrating its structure dependence.

F.7.2 Strong G preference: feedback without commutation residualization

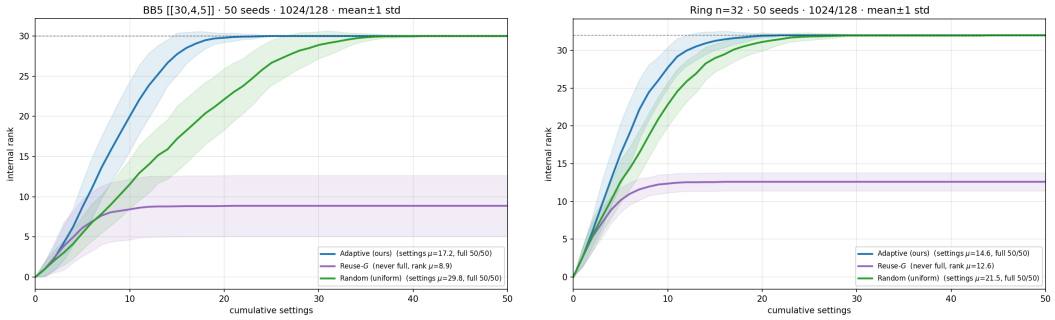
Reuse-G tests a different use of exactly the information available to the adaptive learner. At each round its only state is the current maintained span G_t, but it places a strong preference on the verified directions themselves: it takes the independent generator basis $B(G_t)$ as its proposal pool, removes duplicate entries while preserving the maintained ordering, and randomizes the proposal order. It greedily merges every generator whose non-identity Pauli letters are locally compatible with the current partial product-Pauli setting, then completes unassigned qubits uniformly from $\{X, Y, Z\}$.

No centralizer or residual coordinate space is constructed. The two uses of verified feedback can therefore be summarized as

$$\begin{aligned} \text{Reuse-}G: &\quad B(G_t) \longrightarrow \text{pack verified directions,} \\ \text{Our acquisition}: &\quad G_t \longrightarrow G_t^{\perp} \longrightarrow G_t^{\perp}/G_t \longrightarrow \text{pack unresolved directions.} \end{aligned}$$

The distinction is the role assigned to the same learned representation. Reuse-G is adaptive and G_t-only, but it treats large overlap with G_t as a measurement preference. Our policy instead applies the only universal stabilizer mechanism available from G_t—commutation—to constrain where new structure can remain, with the quotient serving as a nonredundant residual coordinate system. Both use the same single-copy product-Pauli interface, the same per-setting sample budget, and no target-family label. The subsequent quotient interventions go further still: they keep this commutation space fixed and vary how its basis and exploration–exploitation policy are constructed.

Figure 4 shows the resulting trajectories on BB5-30 and Ring-32 over 50 independent runs with a 50-setting budget. Reuse-G rapidly enters a low-rank plateau and never reaches full structural rank in any run, ending at mean ranks $8.86/30$ on BB5-30 and $12.60/32$ on Ring-32. In contrast, the commutation-residual acquisition policy reaches full rank in all 50 runs. The plateau makes the mechanism concrete: adaptive feedback becomes useful when the learned constraints are transformed into a search geometry for unresolved structure.



(a) BB5-30. (b) Ring-32.

Figure 4: Direct reuse of verified feedback. Reuse-G uses the current verified generator basis as its proposal pool and applies the same sequential compatibility rule before random completion. The method saturates at low rank on both systems, whereas acquisition built from the commutation-induced residual geometry continues to full structural recovery. Curves are means over 50 runs and shaded bands are ± 1 SD. Reuse-G reaches full rank in $0/50$ runs on both systems, ending at mean ranks $8.86/30$ on BB5-30 (IQR $[5.2, 11.8]$) and $12.60/32$ on Ring-32 (IQR $[12.0, 13.0]$); the learned-row quotient policy exceeds it by 21.1 and 19.4 terminal ranks respectively.

F.8 Quotient Representation and Ordering Interventions

The quotient interventions use no target generators or family label. Unlike Reuse-G, every variant constructs the commutation-induced residual space. Frozen-state comparisons use the same verified span G_t and quotient; independent learning rollouts start from $G_0 = 0$ and may subsequently learn different spans. The interventions vary coordinates, representatives, ordering, or exploration strength while keeping the physical compatibility rule and 1,152-copy setting cost (1,024 discovery plus 128 independent verification outcomes) fixed. Each rollout control uses 50 runs and a 50-setting limit.

F.8.1 Intervention definitions

Algebraic equivalence versus product-measurement equivalence. The quotient Q_A is an abstract vector space, so any $R \in \mathrm{GL}(q, \mathbb{F}_2)$ gives an equally valid algebraic basis. A product setting, however, depends on the literal Pauli supports and letters of the chosen representatives. Rebasing can therefore change conflicts and co-exposure while leaving G and Q_A unchanged. The interventions below separate this physical realization from the information content of the learned span.

Write C_A for the commuting subspace on the active support A and $Q_A = C_A/G$. Let $\widetilde{u}_1, \ldots, \widetilde{u}_q \in C_A$ be the learned-row tableau coordinates constructed from the admitted physical rows of G as described in Section 3. Their cosets form a hyperbolic coordinate system for Q_A. All interventions below begin from these same coordinates.

randQ. Sample a uniformly random invertible matrix $R \in \mathrm{GL}(q, \mathbb{F}_2)$, define new quotient coordinates $[u_j] = \sum_k R_{jk}[\widetilde{u}_k]$, with the corresponding linear-combination lifts $u_j = \sum_k R_{jk}\widetilde{u}_k$. The abstract quotient space and learned information are unchanged; only the physical coordinate frame is rebased. The same randomized basis is then used for compatibility packing, so randQ directly tests whether algebraically equivalent quotient coordinates remain equivalent under product-measurement geometry.

repR. Keep the quotient coordinates $[\widetilde{u}_j]$ fixed and randomize each physical lift with admissible combinations from G on its core support, without increasing its weight. Each abstract quotient coordinate is therefore preserved.

WeightSort. Keep the learned-row coordinates $[\widetilde{u}_1], \ldots, [\widetilde{u}_q]$ and their lifts exactly as in the default. The proposal *set* is therefore identical to the default's; only the visiting order changes: candidates are sorted by increasing Pauli weight instead of being shuffled, so the lightest representatives are packed first. This is an ordering-only intervention, which is what makes it the cleanest control for exploration: proposal content, proposal weights, and the conflict graph are all held fixed.

Conflict-Degree-Ordered Greedy Packing (CDGP). Keep the same learned-row proposals and build their physical conflict graph from G. Repeatedly select a uniformly tie-broken minimum-degree vertex in the residual graph, retain it, and delete its closed neighborhood. Unassigned physical axes

are still filled uniformly. CDGP is therefore an ordering-only control after the main learned-row coordinate construction, not an independent quotient compiler. It tests whether a more explicit use of the induced conflict graph improves physical realization, while deliberately reducing exploration across conflict resolutions whenever proposal degrees differ.

Pack caps. $K = 1$ retains one randomly encountered proposal and removes joint packing; $K = 3$ permits limited co-exposure; the main compiler has no fixed cap. These interventions preserve the coordinate frame and random ordering.

Centralizer-only. Use C_A itself rather than quotienting out G. This preserves every commuting direction but reintroduces resolved redundancy.

Qubit relabeling. Apply one fixed permutation (seed 2027) to all BB5-48 Pauli columns before learning. This changes only coordinate names, not the physical relation pattern; Figure 5 tests whether deterministic tableau tie-breaking creates the result.

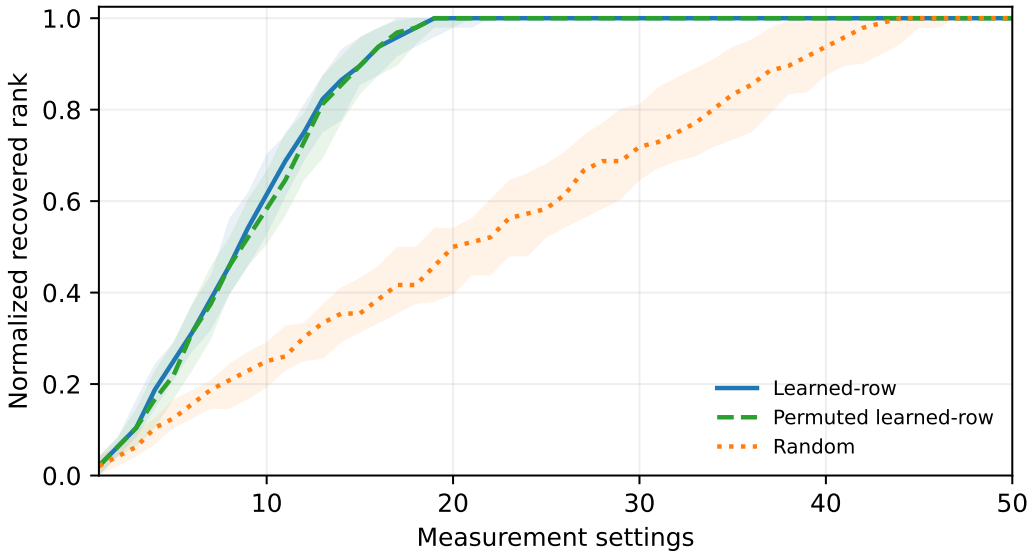


Figure 5: **Qubit-label permutation control on BB5-48.** Original and fixed-relabeled (seed 2027) learned-row trajectories are overlaid against the permutation-invariant Random reference. Mean completion changes only from 19.46 to 19.70 settings over 50 runs.

F.8.2 Full recovery trajectories

Figure 6 shows the full trajectories behind the aggregate intervention results reported in the main text.

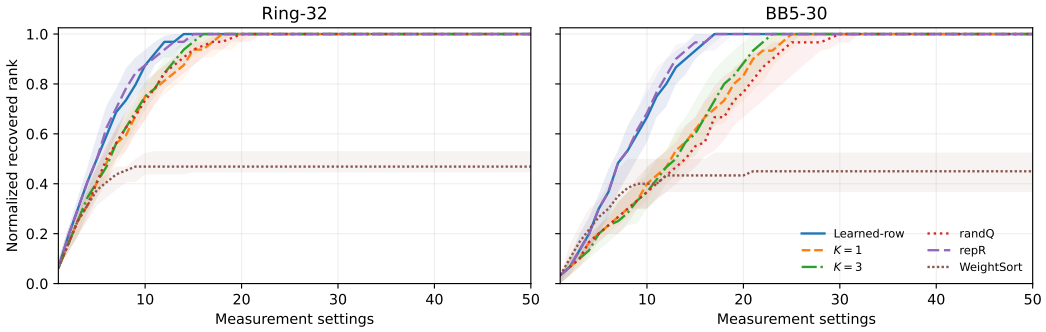


Figure 6: **Direct learned-row ablations over 50 runs.** Limiting packing to $K = 1$ or 3 slows both targets; randQ degrades the physical organization; repR closely tracks the main compiler; deterministic WeightSort stalls. Curves show median normalized rank and interquartile range.

The curves show the same aggregate pattern across the reported run distributions. On both targets repR is statistically indistinguishable from the main compiler in paired IRD, whereas randQ and both finite pack caps are worse. WeightSort reaches only mean terminal ranks $15.96/32$ and $13.12/30$: low individual weight and large compatible subsets cannot replace traversal across residual hypotheses.

F.9 Additional Experimental Diagnostics

The main text presents the principal mechanism comparisons. This section reports the fuller controls used to test alternative explanations, state dependence, traversal, and noisy robustness.

Table 12: **Complete information-use and compiler controls under identical local quantum access.** Entries are mean settings $\pm$ SD when all runs complete; "rank" is terminal-rank mean $\pm$ SD when no run completes. MaxEnt-IG is an analytic identity rather than a duplicate rollout.

Controlled policy	Information source / use	Ring-32	BB5-30
Random	No learned structure; iid local Pauli	21.52 ± 4.82	29.82 ± 4.76
MaxEnt-IG	Uniform completion posterior given G	exactly Random in law	exactly Random in law
Mixed Random	Fixed same-axis preference	22.14 ± 4.83	22.66 ± 8.74
Reuse-G	Pack verified, already resolved directions	rank 12.60 ± 1.21	rank 8.86 ± 3.82
RREF/free-column	Direct quotient-coordinate compiler	16.98 ± 4.06	14.82 ± 4.39
Learned-row	Production compiler; random-order full packing	14.64 ± 3.45	17.24 ± 2.85
$K = 3$	Learned-row with pack cap three	16.66 ± 3.05	23.48 ± 3.89
$K = 1$	Learned-row with one retained proposal	18.84 ± 3.99	25.70 ± 3.90
randQ	Randomized quotient coordinates	20.08 ± 3.74	28.44 ± 5.38
repR	Learned coordinates, randomized representatives	14.64 ± 3.32	17.12 ± 3.03
WeightSort	Increasing representative weight	rank 15.96 ± 2.39	rank 13.12 ± 3.07
Centralizer-only	Compatible directions without quotient removal	18.92 ± 5.89	16.80 ± 4.31
CDGP	Residual minimum-degree conflict ordering	14.72 ± 3.27	17.24 ± 4.11

The MaxEnt-IG row is an exact identity, not a duplicate rollout: Proposition 5 proves that its score ties on every product setting, so uniform tie breaking reproduces Random. A directional lookahead requires additional prior or likelihood information.

CDGP is a directional G-only policy, so it complements rather than contradicts the MaxEnt identity: directionality comes from the physical coordinates of discovered rows. Table 13 broadens the comparison beyond the two shared anchors.

Table 13: **Conflict-degree ordering versus the production random scan.** All entries use 50 paired seeds; both policies complete every run. Settings are mean $\pm$ SD. Differences and 95% paired-bootstrap intervals are CDGP minus random order.

Target	Random order	CDGP	Δ settings [95% CI]	Δ IRD [95% CI]
GHZ-32	12.04 ± 3.08	12.56 ± 3.23	$+0.52\,[-0.50, 1.58]$	$+0.017\,[-0.113, 0.149]$
Ring-32	14.64 ± 3.45	14.72 ± 3.27	$+0.08\,[-1.16, 1.32]$	$-0.054\,[-0.389, 0.295]$
Ring-64	16.34 ± 3.58	15.36 ± 2.67	$-0.98\,[-2.36, 0.38]$	$-0.158\,[-0.395, 0.083]$
BB5-30	17.24 ± 2.85	17.24 ± 4.11	$0.00\,[-1.30, 1.36]$	$-0.263\,[-0.728, 0.195]$
BB5-48	19.46 ± 3.18	19.70 ± 3.22	$+0.24\,[-0.98, 1.48]$	$+0.420\,[-0.135, 0.967]$
Toric-4 $\times$ 8	18.24 ± 3.24	17.90 ± 2.65	$-0.34\,[-1.36, 0.66]$	$+0.183\,[-0.130, 0.521]$
Local-HGP-64	19.02 ± 3.82	18.52 ± 2.60	$-0.50\,[-1.66, 0.66]$	$+0.150\,[-0.192, 0.486]$
Target-balanced	–	–	$-0.14\,[-0.60, 0.31]$	$+0.042\,[-0.098, 0.178]$

No target exhibits a stable setting or IRD improvement. Conflict-degree optimization therefore supplies a strong same-information control but no evidence against the simpler random traversal.

The Reuse-G trajectories reach full rank in $0/50$ runs on both systems, whereas the learned-row quotient policy completes all 50 runs. This control tests the role of *residualization*: feedback is not

useful merely because measurements are correlated with known structure; the known structure must be transformed into a search geometry for what remains unresolved.

Table 14: **End-to-end learned-row packing ablation.** All entries use 50 matched run IDs and all runs finish; values are mean settings to full rank.

Target	Full packing	$K = 3$	$K = 1$
GHZ-32	12.04	12.02	12.48
$\mathrm{GHZ}_8^{\otimes 4}$	12.14	13.20	14.64
Ring-32	14.64	16.66	18.84
Ring-64	16.34	18.98	21.06
BB5-30	17.24	23.48	25.70
BB5-48	19.46	31.26	36.60

Along the main learned-row trajectories, the mean proposal conflict ratio and expected full-greedy pack size are $(0.245, 6.92)$ for Ring-32 and $(0.164, 10.73)$ for BB5-30, evaluated over 682 and 491 nonterminal frozen representations, respectively. The larger BB5 packability agrees with its larger end-to-end separation between full packing and $K = 1$.

Table 15: **Frozen-G capture–exploration decomposition on BB5-48.** Ten learned-row trajectory origins at each rank and 500 replays per origin. Intervals are 95% bootstrap intervals over trajectory origins.

Residual d	Full$-K{=}1$ gain	Captured κ	Completion χ	Full mean pack
40	$1.710\ [1.537, 1.899]$	$+1.108$	$+0.602$	30.2
22	$1.764\ [1.515, 2.022]$	$+2.576$	-0.812	20.3
6	$0.514\ [0.281, 0.774]$	$+1.718$	-1.204	5.9

The identity $\Delta = \kappa + \chi$ is visible directly. Full packing can reduce random-completion gain at later stages, yet its captured rank more than repays that loss at all three diagnosed residual dimensions. At the early $d = 40$ state both terms increase, showing that structured cores can also make the remaining random fill more useful. These are state-conditional mechanism measurements; Table 14 supplies the separate end-to-end comparison.

Table 16: **Quotient-coordinate and realization controls.** Fifty runs per cell. RREF is the alternative coordinate compiler; all other rows intervene on learned-row proposals. Settings are mean $\pm$ standard deviation when all runs finish; otherwise terminal mean rank is reported.

Variant	Ring-32	BB5-30
Full learned-row	14.64 ± 3.45	17.24 ± 2.85
RREF/free-column	16.98 ± 4.06	14.82 ± 4.39
$K = 3$	16.66 ± 3.05	23.48 ± 3.89
$K = 1$	18.84 ± 3.99	25.70 ± 3.90
randQ	20.08 ± 3.74	28.44 ± 5.38
repR	14.64 ± 3.32	17.12 ± 3.03
WeightSort	rank $15.96/32$	rank $13.12/30$
Centralizer-only	18.92 ± 5.89	16.80 ± 4.31

Table 16 summarizes the controls, and Figure 6 shows the corresponding trajectories. Relative to the full compiler, paired IRD increases by 1.65 for $K = 1$ and 1.62 for randQ on Ring-32, and by 4.66 and 5.43 on BB5-30; all four bootstrap intervals exclude zero. repR differs by only -0.03 and -0.04, with both intervals crossing zero. WeightSort keeps the same learned-row proposal set but removes randomized traversal and stalls in every run. Centralizer-only is not consistently beneficial across targets. Together the controls attribute the common-rule gain to organized residual coordinates, co-exposure, and retained exploration rather than to a unique coset lift or individual weight.

Noisy paired statistics. Figure 7 diagnoses physical exposure versus extractor access. For the residual Phase-B rescue reported in Appendix G.3, 20,000 paired bootstrap resamples give a terminal-rank mean effect $+0.80$ (95% CI $[+0.54, +1.08]$) and a settings-consumed mean effect $+8.06$ (95%

CI $[+6.20, +9.92]$). Full-rank completion changes from $8/50$ to $29/50$; the McNemar discordant pairs are $b = 21, c = 0$ with exact $p = 9.54 \times 10^{-7}$. The reduction from 174,406 ambient candidates per late setting to 1.22 preselected residual candidates per setting is a Phase-B search comparison; end-to-end runtime is reported separately.

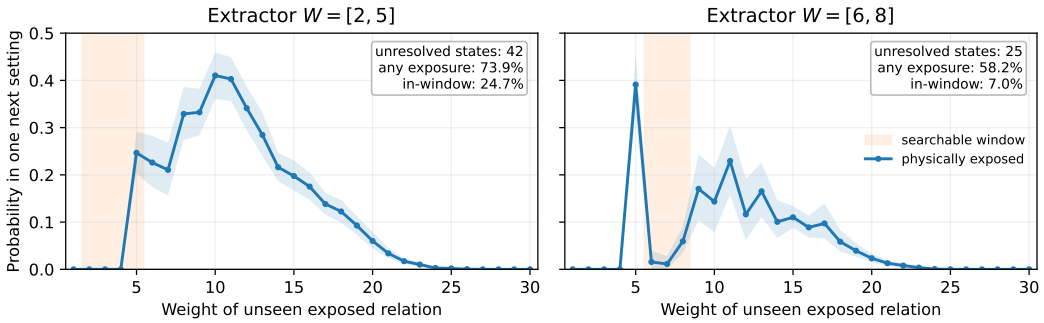


Figure 7: **Terminal-snapshot diagnosis separates physical exposure from extractor accessibility.** For every unresolved run after 50 Phase-A settings, we freeze its verified G and sample the exact next-setting exposure: 500 settings for each of 42 $[2, 5]$ states and 200 for each of 25 $[6, 8]$ states. Shading is a 95% bootstrap interval over run-level snapshots. The policy often exposes unseen structure, but most exposed relations lie outside the active searchable window.

G Noisy Extraction, Validation, and Phase B

The strategy analysis takes a correct span $G \subseteq S$ as its input and quantifies physical exposure. This section specifies the observation and admission interface that supplies that span. Validation controls the risk of invalid updates, while candidate restrictions and missed acceptances determine how much exposed information becomes available for subsequent acquisition. Adaptive and Random use the same extraction and validation rules within each comparison.

G.1 Candidate generation and independent confirmation

Unrestricted recovery of parity constraints in noise is computationally hard under standard LPN-type assumptions (Blum et al., 2003). Bounded-support enumeration is therefore a conventional tractable extractor: for a fixed window W, one product setting supplies $\sum_{w \in W} \binom{n}{w}$ measurable support candidates, polynomial in n for fixed $\max W$. This extractor cost is shared by Adaptive and Random; it is not the cost of quotient acquisition. Phase B later uses verified residual structure to avoid repeating the ambient support search in the tail. Before a nontrivial verified span exists, no target-informed residual list is available from which to select Phase-B candidates. The early learner must therefore pay for a generic searchable class; bounded low support is the standard target-uninformed choice rather than an avoidable cost of our policy.

At each setting, the pure stabilizer outcomes are independently flipped at each measured bit with probability $p = 0.01$. A measurable true weight-w relation has signed parity mean $\pm\mu_w$, $\mu_w = (1 - 2p)^w$; a measurable non-stabilizer Pauli has mean zero. This controlled readout channel tests extractor–acquisition coupling at a hardware-relevant error scale (Appendix H.3). Gate and preparation errors, correlated readout, and crosstalk are outside this channel.

Phase A enumerates supports in $[w_{\min}, w_{\max}]$. For each support, the setting supplies its Pauli letters. One shared 256-row screening block is selected from the 768 discovery outcomes using randomness independent of their values. Supports passing screening are presented to the separate 256-row confirmation pool. All candidates in a pool may share the same rows; candidate tests are not assumed independent. No candidate is generated from the confirmation outcomes.

For parity $Y_j \in \{-1, 1\}$ and $\widehat{\mu} = N^{-1} \sum_j Y_j$, the common noise-aware verifier uses $\alpha = 10^{-4}$, acceptance fraction 0.75, and rejection fraction 0.50:

$$\tau_a(w, N) = \min\{1, \max[\sqrt{2 \log(2/\alpha)/N}, 0.75\mu_w]\}, \tag{G.1}$$

$$\tau_r(w, N) = \min\{\tau_a(w, N), 0.50\mu_w\}. \tag{G.2}$$

Accept when $|\widehat{\mu}| \geq \tau_a$, reject when $|\widehat{\mu}| \leq \tau_r$, and otherwise return inconclusive. Only acceptance can update G; the sign is the sign of the confirmation mean. Screening uses only the acceptance threshold. The learner discards duplicate and rank-dependent accepted candidates. Target-membership checks are logged for evaluation, not used to make admission decisions.

Window	Supports / setting	Adaptive full recovery	Random full recovery
$[2, 5]$	174,406	8/50	0/50
$[6, 8]$	8,482,500	25/50	0/50
$[2, 8]$	8,656,906	48/50	0/50

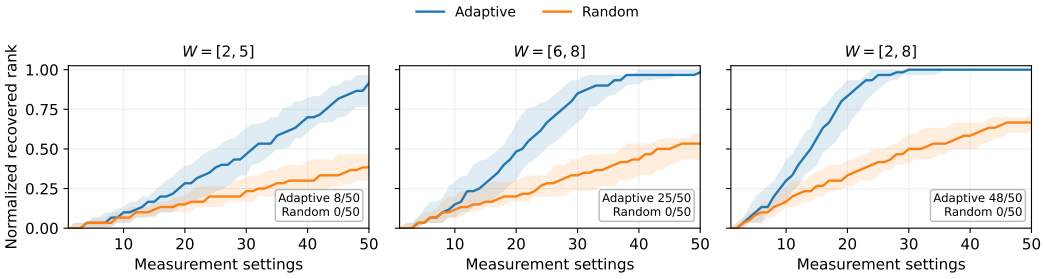


Figure 8: **Noisy recovery trajectories across extractor windows.** BB5-30 with independent 1% readout flips, 50 runs per policy. Curves show median normalized recovered rank with IQR bands; insets report completion at the 50-setting horizon.

Figure 8 shows diagnostic windows on BB5-30, not a rule for choosing weights on an unknown target. We freeze every unresolved state at the 50-setting horizon and use the target only afterward to evaluate exact exposure. The mean probability that one further learned-row setting exposes any unseen relation is 73.9% for $[2, 5]$ and 58.2% for $[6, 8]$, while the probability of exposing one inside the respective searchable class is only 24.7% and 7.0%. These target-aware checks identify an accessibility bottleneck at the diagnosed terminal states; they do not assume that each trajectory has already converged to a permanent plateau.

Target-uninformed operating rule versus diagnostic windows. The deployed low-weight route fixes $W = [2, 5]$ from a classical candidate budget, not from the hidden BB5 generators. Its Phase-B switch uses only observable learner state: residual rank gap at most three and three consecutive rounds without a verified rank increase, with a final attempt at the Phase-A cap. The $[6, 8]$ and $[2, 8]$ runs are target-aware diagnostic interventions used to establish that extraction accessibility changes the trajectory; they are not proposed as automatic window choices. A generic alternative is a budgeted expansion schedule that increments $w_{\max}$ after validated stalls, but its rapidly increasing classical cost and stopping calibration are not evaluated here. We therefore report the fixed low-weight rule and the residual tail rule rather than claim automatic window selection.

G.2 A TRAJECTORY-LEVEL ADMISSION BOUND

Proposition 6 (Validation with adaptive queries and shared test pools)**.** *Condition on the past, the new setting, and the candidate list, before its independent validation outcomes are observed. Suppose every candidate is measurable in that setting, N_t fresh iid outcomes are used, and no additional outcome-dependent tests are generated from that validation pool. Then the probability of at least one non-stabilizer admission over at most T rounds is bounded by*

$$\mathbb{E} \sum_{t=1}^{T} \sum_{h \in \mathcal{H}_t} 2 \exp\{-N_t \tau_a(w_h, N_t)^2/2\}. \tag{G.3}$$

A valid bound for incorrect signs of true relations is obtained by replacing each summand with $\exp\{-N_t(\mu_{w_h} + \tau_a(w_h, N_t))^2/2\}$.

Proof. Conditional on the pre-validation information, a non-stabilizer candidate has iid bounded parity variables with mean zero. Two-sided Hoeffding bounds its acceptance probability by the

summand in Equation G.3. A true relation accepted with the wrong sign requires a deviation of at least $\mu_w + \tau_a$ in the opposite direction; one-sided Hoeffding gives the second bound. Union bound over the fixed candidate list, then over rounds, and average the conditional bounds. Shared outcomes create dependence between tests but do not invalidate the union bound. Stopping early only removes possible tests. The candidate list may depend arbitrarily on previous rounds and an independent discovery pool. Adaptive choices across rounds therefore remain covered. □

This bound applies to the stated independent readout model and fixed-size validation pools; it does not license optional extra-shot stopping or generating new candidates from their own validation data. For the broad Phase A window, $w \le 8$, $N = 256$, so $\tau_a \ge 0.75(0.98)^8 = 0.63807\ldots$. Bounding the confirmation list by all 8,656,906 possible supports at every one of 50 settings yields a non-stabilizer-admission probability below 2.02×10^{-14} per trajectory. This conservative bound does not multiply two screening and confirmation probabilities, and already accounts for all support comparisons. The corresponding wrong-sign bound is below 2.60×10^{-115}. These small numbers depend on the model and sample sizes; they neither bound false rejection nor guarantee complete recovery. A post-hoc audit of 5,607 admitted relations across the 200 new learned-row noisy trajectories found no non-stabilizer admission. Target-membership logs provide this empirical check but are never used by the learner.

Proposition 7 (Admission efficiency of the implemented verifier)**.** *Suppose a true weight-w relation is included in the candidate list and has mean magnitude $\mu_w > \tau_a(w, N)$. On one fresh block of N outcomes it is accepted with the correct sign with probability at least*

$$\eta_w(N) = 1 - \exp\{-N(\mu_w - \tau_a(w, N))^2/2\}.$$

If screening and confirmation use independent blocks, the corresponding two-stage admission probability is at least $\eta_w(N_{\text{screen}})\eta_w(N_{\text{verify}})$.

Proof. After orienting by the true sign, failure to cross the positive acceptance threshold implies a lower-tail deviation of at least $\mu_w - \tau_a(w, N)$. One-sided Hoeffding gives the first bound. Independence of the two data blocks gives the product bound. □

For the reported $p = 1\%$ channel and $N_{\text{screen}} = N_{\text{verify}} = 256$, the worst case over $w \le 8$ is $w = 8$ and gives $\eta_w(256)^2 \ge 0.99389$. This is a conditional admission guarantee for a true relation that the chosen window actually enumerates; it does not claim that the window contains every exposed relation. For direct Phase-B verification, $N = 1024$ and $w \le 30$ give $\eta_w(1024) \ge 0.99992$. Consequently the η_E, η_B factors in Theorem 3 are statistical quantities supplied by the extractor interface, rather than an assumption that extraction is complete.

Starting from an empty correct span, the absence of any non-stabilizer or wrong-sign admission ensures that all subsequent stored relations remain correct. The two trajectory-level bounds therefore control the probability of leaving the correct-representation interface, including through adaptive feedback. A true relation that is not proposed or not accepted instead leaves information unresolved: it can alter the learning trajectory without making the retained constraints invalid. Correctness and extraction completeness thus have different roles.

How imperfect extraction affects the strategy theory. On the event that every admitted row has correct membership and sign, every conditional exposure, matching, and joint-capture statement applies at each realized G_t, even if many true relations were missed. False rejection or a restricted window can change which correct span is reached and how long the trajectory remains there; it can invalidate a completion-time prediction, but not the conditional policy calculation at that span. A false admission or sign error is qualitatively different because $G_t \subseteq S$ may fail and the quotient no longer represents the true residual problem. Proposition 6 controls the probability of this event under the stated channel. More realistic correlated channels require a replacement validation guarantee before the same strategy-level reasoning can be invoked.

G.3 Packed residual verification in Phase B

The Phase-A/Phase-B terminology is local to this noisy tail protocol. Phase A is Algorithm 1 with the bounded-support noisy extractor; Phase B is the optional direct residual-verification tail described here. The tail-enabled low-weight learner switches from Phase A when the remaining rank gap is at most three and three rounds have stalled. It also attempts the tail after reaching the Phase A limit

without recovery. Phase A has at most 50 rounds and Phase B at most 15; recovery ends the run immediately.

Before collecting a tail batch, form up to 16 nonzero random linear combinations of an active quotient basis. On this packed active-support path the quotient representatives are used directly, with no within-coset randomization. If the active quotient is empty, the verifier uses its full-space quotient fallback to force one nonzero residual anchor before completion. Greedily pack compatible proposals and then fill free axes with the verifier's known-generator-avoidance completion: it bans avoidable matching letters and uses short randomized rejection fallback, while leaving already forced axes unchanged. The packed candidates are fixed before sampling. A fresh 1024-outcome pool then verifies these candidates directly. There is no ambient support enumeration, no discovery stage, and no use of this batch to generate new candidates. Accepted independent relations update G. The packed span's target intersection describes the information physically exposed; Phase B verifies only its preselected candidates, so its admitted rank can be smaller than that intersection rank.

With at most 16 candidates per round, $w \leq 30$, and $N = 1024$, the same theorem gives a Phase B non-stabilizer-admission bound below 2.92×10^{-35} over 15 rounds and a wrong-sign bound below 5.68×10^{-201}. These bounds are evaluated for the actual Phase B sample size and possible weights, rather than borrowed from the low-weight Phase A test.

Neither the acquisition policy nor targeted verification requires a global pure-state or stabilizer-state promise. They operate on whichever Pauli-parity relations pass validation; if none do, $G = 0$ and the acquisition law is exactly uniform Random. Exact stabilizer targets are used here to define rank-n completion, score ground-truth recovery, and obtain the closed algebraic theory. A mixed or non-stabilizer state may still supply useful exact or strong Pauli-type relations, producing a partial stabilizer-type constraint representation rather than a full-state certificate. Softened relations require calibrated approximate-constraint semantics and a different stopping rule. Spatially correlated readout also changes the weight-only signal model μ_w, while temporal correlation invalidates the iid concentration premise. These channels require robust validation, not a change to the principle of using validated relations to propose a small Phase-B candidate list.

Resource / outcome, 50 runs	Low-weight only	Low-weight + Phase B
Full recovery	8/50	29/50
Mean terminal true rank	26.08	26.88
Settings: Phase A / Phase B	2473 / 0	2417 / 459
Mean settings consumed	49.46	57.52
Total candidate evaluations	431,306,038	421,539,864
Tail candidates / tail setting	174,406	$562/459 \simeq 1.22$

Thirty-six tail-enabled runs use more than 50 settings, with maximum 65; 37 use more settings than their matched baseline run. These are different counts. Paired run-level bootstrap gives a mean terminal-rank difference 0.80 with 95% interval $[0.54, 1.08]$ and an actual-settings difference $+8.06$ with interval $[+6.20, +9.92]$. These compare each protocol under its own stopping rule, not equal-horizon success probabilities. The broad-window learner has higher completion; the tail method improves the low-weight route while testing only a small residual list in its additional settings.

H Measurement Access and Learning Objectives

H.1 A fixed observation interface

Our acquisition layer uses independent fresh copies with local Pauli basis rotations and readout. It adds no entangling measurement operation, although the preparation of the unknown target may of course require entanglement. This same measurement interface supports Pauli compressed-sensing tomography, many-body reconstruction from local data, and product-Pauli shadow estimators (Gross et al., 2010; Cramer et al., 2010; Huang et al., 2020; Hadfield et al., 2022; Huang et al., 2021; Hadfield, 2022). Those methods show that the hardware primitive is broadly useful; our first distinction is the learned object—an unknown constraint system rather than a state estimate or estimators for supplied observables.

Stabilizer learning and verification span several distinct single-copy and multi-copy measurement primitives. Bell sampling uses pairwise entangling measurements on two copies (Montanaro, 2017). The single-copy protocol of Chia et al. (2024) permits a full randomly chosen commuting stabilizer-basis measurement on each copy; such a basis generally requires an entangling Clifford circuit

before readout. Product-measurement graph-state learning instead keeps local readout but assumes a graph-state family (Ouyang & Tomamichel, 2022). Pauli verification and direct fidelity estimation can also keep local readout, but their tested target or observable distribution is supplied in advance (Flammia & Liu, 2011; Kalev et al., 2019; Dangniam et al., 2020). We supply none of a graph, code family, generator list, or target state to acquisition.

Here *target-uninformed* means that the only instance-specific information available to the policy is the measurement record summarized by its verified relations. Pauli commutation, closure, and physical support provide the stabilizer-specific inductive bias. Generator identities, support patterns, graphs, code constructions, family labels, and target distributions remain hidden from acquisition. At $G = 0$ the policy is exactly uniform Random; the reported suites then measure how verified structure changes subsequent queries.

Same-access distinction: supplied coordinates and learned coordinates. Adaptive Pauli allocation, derandomization, and biased-shadow methods change local measurement probabilities using a supplied observable collection or predictive objective (Huang et al., 2021; Hadfield et al., 2022; Hadfield, 2022). Product-measurement graph-state learning organizes inference inside a promised graph family (Ouyang & Tomamichel, 2022). These approaches establish the value of product-Pauli acquisition when the task already supplies the coordinates that organize allocation or search. Our coordinate system is generated online: the learner discovers and verifies relations from bitstrings, G_t defines the feasible quotient Q_{G_t}, and the admitted physical rows determine the learned-row frame from which the next product setting is compiled. The same unknown relation system is therefore both the learning target and the state that organizes future queries. We call this *representation-generating feedback*.

From stabilizer algebra to an acquisition policy. Stabilizer commutation and quotienting provide the exact algebraic feasible space. Product-Pauli acquisition adds a physical geometry to that space. Two bases related by $\mathrm{GL}(q, \mathbb{F}_2)$ encode the same quotient, yet their literal Pauli supports and letters can induce different conflict graphs and different co-exposure in one local setting. The learned-row compiler fixes this map from residual algebra to a hardware action by retaining a physical frame inherited from verified rows and packing that frame under local compatibility. The randQ experiment holds the learned span and abstract quotient fixed while changing only this coordinate frame; repR separately changes the physical lift of fixed quotient coordinates. These interventions make the algebra-to-measurement layer empirically identifiable rather than a change of notation. Table 17 summarizes the neighboring learning objectives and access assumptions.

H.2 Compatibility requirements for suite-wide numerical baselines

A suite-wide numerical baseline must match the learning task, quantum access, and supplied information of the end-to-end protocol. Under our protocol it takes fresh-copy product-Pauli bitstrings as quantum input, uses online verified relations as its instance-specific state, and outputs previously unknown relations. Bell measurements alter the quantum primitive, while graph-state promises alter the supplied information.

Route	Extra input or operation	Mismatch with suite-wide protocol
Generic / compressed tomography	State variable plus a low-rank, locality, or ansatz model for tractability	A generic density matrix has 4^n entries; a structured solver changes the supplied model and must still be followed by relation search.
Classical shadows	Observable list or predictive objective	Estimating a Pauli is efficient once named; discovering an unknown generating set still needs a candidate generator. Exhaustive Pauli post-processing has 4^n candidates.
Ouyang–Tomamichel	Promise of a d-regular graph state in the graph-state measurement frame	It is a valid product-measurement comparator on that promised graph subset, but is undefined as the same algorithm on code/check and unrestricted structured targets.
Chia–Lai–Lin	Random stabilizer-basis measurement on each copy and a stabilizer/low-magic promise	A general stabilizer-basis measurement uses an entangling Clifford basis change, rather than independent single-qubit Pauli axes.
Bell sampling	Two coherent copies and transversal Bell measurements	Pairwise entangling measurements across copies are outside the fresh-copy product interface.

Table 17: **Learning objectives separated by quantum-side access and prior information.** "Product Pauli" means independent single-qubit basis rotations and readout on each fresh copy.

Route	Quantum measurement	Information supplied	Output / objective
Pauli / structured tomography (Gross et al., 2010; Cramer et al., 2010)	Product-Pauli-compatible local data	Low-rank or many-body reconstruction model	State estimate
Pauli shadows (Huang et al., 2020; Hadfield et al., 2022; Huang et al., 2021; Hadfield, 2022)	Product Pauli	Observable list, Hamiltonian, or estimator objective for optimized variants	Property estimates
Stabilizer verification (Flammia & Liu, 2011; Kalev et al., 2019; Dangniam et al., 2020)	Local Pauli tests	Target state, stabilizers, or target-dependent sampling law	Accept/reject or fidelity
Graph-state learning (Ouyang & Tomamichel, 2022)	Product measurements	Graph-state family promise	Unknown graph state
This work	Product Pauli	No instance/family coordinates; G_t is learned and verified online	Unknown stabilizer-type constraints and adaptive acquisition
General stabilizer-basis learning (Chia et al., 2024)	Entangling basis change within each copy	Stabilizer / low-magic promise	State recovery
Bell sampling (Montanaro, 2017)	Pairwise Bell measurements across two copies	Stabilizer-state promise	State recovery
Collective identification (Aaronson & Gottesman, 2008)	Joint measurement across multiple copies	Stabilizer-state promise	State recovery

With an empty observable/candidate list, local-Pauli shadow acquisition uses the same random local-basis law as the Random baseline here; the downstream estimator and post-processing differ. Turning such predictions into unknown-generator discovery still requires a candidate-generation mechanism.

Operational numbers on the overlapping graph subset. The closest overlap is the degree-two Ring graph-state family. Learned-row uses the same uninformed policy as on every other target and completes Ring-32 and Ring-64 in 14.64 ± 3.45 and 16.34 ± 3.58 product-basis settings, respectively, under the common 1152-copy-per-setting finite-sample protocol. That batch size is held fixed solely as a common finite-sample realization of each queried setting. The native Ouyang–Tomamichel guarantee instead identifies a promised d-regular graph state from $O(d \log(1/\epsilon) + d^2 \log n)$ copies; for Ring, $d = 2$ (Ouyang & Tomamichel, 2022). Bell sampling learns an arbitrary promised pure stabilizer state with $O(n)$ copies and exponentially small failure probability, but consumes coherent pairs in transversal Bell measurements (Montanaro, 2017). We report these as family- or access-informed native resource references on different resource axes. Their asymptotic copy bounds and our empirical setting counts use different supplied information or quantum operations. Stabilizer verification is still less comparable—its target relations are inputs, so unknown-relation discovery cost is undefined rather than zero.

Consequently, the Random–Adaptive comparison deliberately gives both policies the same relation extractor and representation: it isolates the second-layer acquisition contribution. The first-layer representation claim is supported instead by the task mismatch above—tomography first builds a state object, whereas shadow prediction first needs candidates—and by demonstrating direct joint discovery and maintenance of unknown stabilizer-type constraints. Giving an adjacent method our extractor would make it another acquisition policy inside this pipeline; giving it its native promise or measurement primitive answers a different learning problem. Family-informed rules remain useful upper references, and Mixed Random explicitly tests one such fixed prior.

Why the product interface is the deployment constraint. Product Pauli readout adds only one layer of independently chosen single-qubit rotations before the platform's standard readout, and is already used in large-register randomized-measurement experiments (Votto et al., 2026; Fischer et al., 2026). Shallow-shadow protocols show a continuum in which entangling circuits before

readout can improve some property-estimation tasks (Bertoni et al., 2024; Hu et al., 2025); they use a strictly stronger measurement primitive than the depth-zero product setting studied here. Bell or more general collective learners additionally require coherent simultaneous copies and inter-copy entangling operations. Collective advantages have been demonstrated for few-copy, small-system experiments (Hou et al., 2018; Conlon et al., 2023), but these operations are not currently a routine scalable characterization interface for tens of qubits. The access distinction is therefore operational rather than terminological: stronger-resource results provide complementary reference points for the hardware primitive fixed in this paper.

Information-matched strategy comparisons. The controls fix the measurement/extraction protocol and provide no target generators or family label. They vary distinct uses of the available information:

Design axis	Compared rules	Question isolated
Fixed preference	Random / Mixed Random	Can a fixed same-axis bias substitute for learned organization?
Feedback direction	Reuse-G / Adaptive	Should verified relations be repeated or used to construct unresolved directions?
Coordinates and lifts	Canonical / randQ / repR	Which physical realization of the same abstract span is useful?
Proposal selection/order	Random order / WeightSort / MinWeight-Q	Can weight preference replace randomized exploration?
Pack capacity	$K = 1$, $K = 3$, full packing	When does joint capture outweigh displaced completion?

Information matching is defined at the level of permitted input. Full rollouts naturally develop different G_t, while frozen-G tests hold the actual learned span fixed. Mixed Random ignores feedback and supplies a fixed axis bias with no target-specific generator information. These comparisons connect the observed gains to residualization, physical organization, and retained exploration.

Predictive objectives and the verified-state interface. The same G admits completions with opposite query preferences (Appendix A). Thus an information-gain or lookahead objective needs a specified predictive law over these completions in addition to the feasible quotient Q_G. This law could be an explicit prior conditioned on G, or a likelihood model retaining outcome statistics beyond G. Our policy instead operates directly on verified constraints, without maintaining a posterior or estimating the value of hypothetical future updates.

Verification controls admission of hard relations, while rejected and unverified statistics can still carry likelihood information. A noise-aware likelihood can use such statistics without admitting them as exact constraints. That yields a richer policy state under the same quantum measurement interface. Exhaustive storage over all Pauli strings is exponential, but is not necessary for every approximate predictive model. The experimental question here is which structural use of verified relations improves acquisition under the common constraint-learning pipeline.

Likewise, exact cardinality packing optimizes pack size, while the discovery objective is newly exposed target rank. Appendix A.2 separates its unrestricted worst-case hardness from the randomized compiler's core-retention guarantee. The controls assess information capture and exploration rather than closeness to a maximum-cardinality pack.

H.3 Stabilizer-type constraint discovery versus reconstruction and prediction

Generic tomography reconstructs a state description; structured tomography can exploit low rank, locality, or an ansatz to make that description much smaller (Haah et al., 2017; Gross et al., 2010; Cramer et al., 2010). The Pauli-measurement literature is especially relevant here: compressed sensing reconstructs low-rank states from Pauli data, while adaptive tomography changes measurements to reduce reconstruction error (Gross et al., 2010; Lange et al., 2023; Mahler et al., 2013; Ferrie, 2014; Quek et al., 2021). These are state-level objectives.

Classical shadows instead organize randomized product-Pauli measurements into estimators for observable properties (Huang et al., 2020). Local bias, derandomization, and adaptive Pauli allocation improve the estimation of a given Hamiltonian or observable collection (Hadfield et al., 2022; Huang et al., 2021; Hadfield, 2022); the product-measurement implementation has also been demonstrated

experimentally (Zhang et al., 2021). Observables need not all be chosen before collection, but evaluating an unknown constraint still requires generating its Pauli candidate. The first contribution concerns how those unknown candidates are discovered jointly and turned into a reusable algebraic state before any acquisition advantage is considered. The three routes place that discovery problem in different steps:

Route	Maintained object	Unknown-structure discovery
Generic tomography	State estimate $\widehat{\rho}$	Reconstruct a state, then identify relations it satisfies.
Shadow prediction	Snapshots / observable estimator	Generate Pauli candidates, estimate their expectations, and collect independent relations.
Stabilizer-type constraint learning	Verified Pauli-relation span G	Discover relations jointly from outcome batches and reuse their algebraic closure immediately.

A learning object before a state-family promise. For a general density operator ρ, a deterministic Pauli relation has the form $P\rho = (-1)^c\rho$, equivalently $\mathrm{Tr}(\rho P) = (-1)^c$. It can therefore be discovered as a parity constraint in a matching product setting without first assuming that ρ is a stabilizer state. Consistent exact Pauli relations commute: anticommuting Paulis have no common eigenvector. At rank r, their signed span restricts the support of ρ to a common eigenspace of dimension 2^{n-r} and is stored in $O(nr)$ bits. Such constraints can be informative without completely specifying the target; a generic state need not have any nontrivial exact Pauli constraint. This exact relation space is the object learned here. Pure stabilizer states instantiate the complete case $r = n$, where relation recovery also identifies the state; partial spaces remain meaningful outputs without that identification claim. When no relation is validated, the maintained span is empty and the policy reduces exactly to uniform Random. The guarantees concern recovery and exposure of this stabilizer-type structure under the stated observation and validation models.

Our learning object is thus a growing system of signed relations, reusable before state identification. A product setting exposes many parities at once; in the noiseless protocol, binary elimination finds a basis of candidate relations in time polynomial in batch size and n, without enumerating all parity supports or all 4^n Pauli strings. Finite-sample verification then checks the candidates (Appendix E). The learned span has $O(n^2)$ storage, its commutation constraints define a residual hypothesis geometry, and that geometry guides future settings. These properties make the verified relation span a useful learning state; the underlying stabilizer encoding itself remains standard.

Estimating a supplied Pauli and searching for an unknown generating set are distinct computational tasks. Conversely, a shadow record can be combined with a structure-discovery algorithm; our distinction is the discovery layer and reusable learning state. Shadow records can also carry stabilizer information when paired with an appropriate candidate-generation procedure. The Random baseline also uses our relation extractor and representation, so Random–Adaptive recovery contrasts isolate acquisition effects rather than measure the representation advantage over tomography or shadows.

The relevant costs should be separated. Explicit generic density matrices have 4^n entries. An unrestricted Pauli candidate list is exponential. Uniform local Pauli measurements match a specified weight-w string with probability 3^{-w}. Structured tomography, supplied-observable prediction, and classical simulation of a *known* stabilizer state can all remain efficient under their respective structural assumptions. Our experiments concern discovering an unknown relation system and using it to improve acquisition under the fixed interface. Rare-setting costs and the LPN-type hardness of unrestricted noisy parity discovery remain present alongside the compact representation. The standard tractable response is a structured candidate class: bounded-support enumeration is polynomial in n for fixed maximum weight. Our noisy protocol uses that common extractor, reports its cost separately from acquisition, and then uses Phase B to turn the learned representation into a much smaller targeted tail search. The representation contribution is the direct, reusable route from measurement relations to unknown structure. Unrestricted noisy parity learning remains outside the present computational scope.

Hardware relevance of the simulation benchmark. The observation interface uses local basis rotations and bitstring readout, as in large-scale local randomized-measurement experiments (Votto et al., 2026; Fischer et al., 2026): the former learns states up to 96 qubits, and the latter implements informationally complete measurements up to 80 qubits with more than 3×10^4 basis randomizations per circuit. Our 30–64-qubit targets therefore lie within demonstrated register scale, although our

learner itself is evaluated in simulation. The independent 1% readout-flip channel used for BB5-30 is also representative in magnitude. Bengtsson et al. (2024) report 1.5% error per qubit for simultaneous 17-qubit superconducting readout, while Korhonen et al. (2025) describe 1–3$\%$ as typical for single-qubit measurements on IBM hardware and use random local $X/Y/Z$ measurements experimentally. Algorithm-design work on this interface also commonly separates hardware simplicity from platform deployment: for example, the product-Pauli derandomization of Huang et al. (2021) establishes its measurement-design gains by theory and numerical experiments.

Simulated outcome sampling lets us compare acquisition and extraction under a controlled observation channel. The purpose of the 1% experiment is to test whether statistically verified relations remain a reliable policy state and whether extractor bias feeds back into acquisition. It is not a surrogate for a device-level noise study. Preparation error can remove exact stabilizer relations; imperfect basis rotations, temporal drift, correlated assignment errors, and crosstalk can make the observation channel basis-, qubit-, and time-dependent. Those effects require calibrated likelihoods and verification models. We therefore use *hardware-scale controlled benchmark* to mean that the measurement primitive, register size, and error magnitude are experimentally grounded. Full hardware execution and broader physical error channels remain future evaluation axes.

H.4 The classical adaptive-learning interpretation

The distinction is between *structure in the target* and *structure specified to the learner*. Graph connections, check supports, and bottleneck decompositions enter only the analysis. Acquisition acts on relations recovered from data and builds its physical coordinates and compatible packs from that learned state. These operations provide the inductive bias that turns verified structure into future queries. The structural results identify circumstances in which that bias improves exposure.

The representation is both an inference result and a policy state. Quotient construction removes resolved directions, but does not determine a unique best query: physical representatives have different compatibility patterns, and the unknown completion matters. Randomized packing combines exploitation of this learned organization with exploration through proposal order and unassigned axes. The exposure spectrum measures the independent information made accessible by this action distribution.

The maximum-entropy law over completions consistent with G provides a clean neutral reference: every product setting has the same one-step gain distribution (Proposition 5). Hence a generic information-gain label does not define a stronger G-only action. Directional lookahead requires extra prior, likelihood, or training information. Our compiler instead uses the physical organization of verified rows as an inductive bias. The graph and check certificates state when that bias improves exposure, and the frozen-G decomposition measures whether captured structure repays displaced random completion. The resulting boundary is structural: valid constraints enable compression, while inherited physical organization enables an acquisition advantage.

Under noise, the extractor determines which constraints can enter the policy state. Its bias therefore changes later data collection, not just the estimation of a fixed dataset. Phase B closes the loop in the other direction: the partial representation defines a smaller, physically targeted candidate search. This coupling between stabilizer-type representation, acquisition, and extraction is the learning principle studied here. The paper makes no algorithmic or empirical claim for other relation languages; extending the same closed-loop idea would require a different composition law, verifier, and query compiler.